CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
UNIDADE ARAXÁ

NAIM KHALIL AYACHE

AVALIAÇÃO DAS CONDIÇÕES DE ESTABILIDADE DE TALUDES DE MINA POR MEIO DE ÁRVORES DE DECISÃO.

ARAXÁ-MG
2021

NAIM KHALIL AYACHE

AVALIAÇÃO DAS CONDIÇÕES DE ESTABILIDADE DE TALUDES DE MINA POR MEIO DE ÁRVORES DE DECISÃO.

Trabalho de Conclusão de Curso apresentado ao Centro Federal de Educação Tecnológica de Minas Gerais - Unidade Araxá, como requisito parcial para obtenção do grau de Bacharel em Engenharia de Minas.

Orientador: Prof. Dr. Allan Erlikhman Medeiros Santos

ARAXÁ-MG

2021

NAIM KHALIL AYACHE

AVALIAÇÃO DAS CONDIÇÕES DE ESTABILIDADE DE TALUDES DE MINA POR MEIO DE ÁRVORES DE DECISÃO.

Trabalho de Conclusão de Curso apresentado ao Centro Federal de Educação Tecnológica de Minas Gerais - Unidade Araxá, como requisito parcial para obtenção do grau de Bacharel em Engenharia de Minas.

Data de Defesa: Araxá, 10 de setembro de 2021.

Orientador: Prof.º Me. Allan Erlikhman Medeiros Santos
Centro Federal de Educação Tecnológica de Minas Gerais - Unidade Araxá

Prof.° Dr. Hildor José Seer
Centro Federal de Educação Tecnológica de Minas Gerais - Unidade Araxá

Dr. Luciano Nunes Capponi
Mosaic Fertilizantes Ltda.

DEDICATÓRIA

DEDICO ESTE TRABALHO

Aos meus pais e irmãos, familiares, amigos e ao meu orientador que me aguentaram por todo o tempo que levei fazendo este trabalho.

AGRADECIMENTOS

Queria agradecer a todos que me ajudaram nesse percurso desejando muito juízo a todo mundo, agradeço aos meus familiares e amigos por me ajudarem e que foram agraciados com o juízo da minha presença. Queria agradecer também ao meu orientador por ter participado do desenvolvimento deste trabalho, você foi recompensado com a áurea do meu juízo. Além disso, queria agradecer a você leitor, que teve a curiosidade de ler este trabalho e passar pelos meus agradecimentos, eu lhe concedo o dom do juízo, não precisa me agradecer...

Este parágrafo é para os leitores que estão desenvolvendo os seus próprios trabalhos de pesquisa, eu recomendo que você esqueça isso, a vida tem prazeres muito maiores que ficar escrevendo TCC e artigos, saia de casa, toque a grama e sinta o cheiro da relva molhada, aprimore o seu bom senso e juízo e deixe as questões frívolas e a lenta marcha do tempo que nos leva por um caminho sem volta de lado. Aprecie a vultuosa e sorumbática companhia dos seus amigos e familiares, aprecie a agremiação calorosa e beligerante da solidão e contemple o juízo cróceo que eu te concedo.

Recomendo uma leitura reforçada dos Tópicos 2 e 3 para que você tenha total entendimento dos conceitos abordados neste trabalho. As ideias tratadas aqui podem ser obscuras caso você não tenha familiaridade com o assunto. Boa Leitura.

EPÍGRAFE

"Se tivesse de escolher entre a alegria e a tristeza, não trocaria as tristezas do meu coração, pelas as alegrias do mundo inteiro."

Khalil Gibran

RESUMO

A prática de ângulos de talude mais íngremes em lavras a céu aberto com vistas a fatores econômicos e produtivos impacta o aumento das probabilidades de rupturas, de médio e macro escala. As rupturas em taludes afetam diversos setores da mineração, tais como operação, segurança, ambiental e econômico. Consequentemente, a avaliação contínua da estabilidade de taludes é um componente vital do projeto e da operação a céu aberto. A presente pesquisa tem como objetivo o desenvolvimento de uma ferramenta para avaliação da estabilidade de taludes rochosos de mineração, com base em um banco de dados geotécnico mundial, utilizando a árvore de decisão, técnica de aprendizado de máquina amplamente utilizada. As variáveis que compõem o banco de dados são tipo de rocha, precipitação, resistência da rocha intacta, RQD, alteração, regime tectônico, fluxo de água subterrânea, propriedades das descontinuidades, altura e ângulo dos taludes, método de desmonte, histórico de instabilidade. Além destas variáveis o banco de dados apresenta a informação de estabilidade dos taludes, em três níveis de estabilidade: os taludes estáveis, instáveis em macro escala e com instabilidade em bancadas (média escala). Diferentes modelos são avaliados na presente pesquisa, o modelo geral (com todas as variáveis); um modelo matemático (com variáveis selecionadas a partir de sua importância), utilizando o *Random Forest*, para escolha das variáveis com vistas ao *target* estabilidade do talude); dois modelos de especialistas utilizando variáveis aplicadas em sistemas de classificação. A validação do modelo foi feita por meio da amostra de teste, usando *bootstrap* e matrizes de confusão por partição visando a reprodutibilidade dos resultados. Um estudo dos erros utilizando a Análise da Componente Principal (PCA) permitiu a identificação de amostras inconsistentes com as demais, com isso os modelos foram refeitos e comparados com os anteriores. Desta forma foi encontrada a melhor modelagem no modelo baseado nas variáveis selecionadas pelo *Random Forest* com o banco de dados sem as amostras problemáticas. Um ponto importante do presente trabalho é que este não substitui as análises clássicas de estabilidade de taludes, pelo contrário, contribui para engenheiros e geólogos com uma ferramenta para monitoramento das condições de estabilidade dos taludes em uma mineração. Sabe-se que a análise de estabilidade de taludes deve ser feita durante toda a vida útil da mina e, portanto, acredita-se que a ferramenta aqui proposta pode otimizar a seleção dos taludes mais susceptíveis a instabilidade.

PALAVRAS-CHAVE: Condições de estabilidade de taludes de mina. Árvores de decisão. *Random Forest*. Parâmetros geomecânicos

ABSTRACT

The practice of steeper slope angles in open pit mining for economic and production factors impact the increase in the likelihood of medium and macro-scale failures. Slope failures affect several mining sectors, such as operational, safety, environmental and economic. Consequently, continuous evaluation of slope stability is a vital component of open-pit design and operation. The present research aims to develop a tool for mining rock slope stability assessment based on a worldwide geotechnical database using a decision tree, a widely used machine learning technique. The variables that make up the database are rock type, precipitation, intact rock strength, RQD, alteration, tectonic regime, groundwater flow, discontinuity properties, slope height and angle, blasting method, and instability history. In addition to these variables, the database presents the stability information of the slopes, in three levels of stability: the stable slopes, unstable at macroscale, and with bench instability (medium scale). Different models are evaluated in this research, the general model (with all variables); a mathematical model (with variables selected based on their importance), using Random Forest, to choose the variables with a view to target slope stability); two expert models using variables applied in classification systems. The model validation was done through the test sample, using bootstrap and partition confusion matrices aiming at the reproducibility of the results. A study of the errors using Principal Component Analysis (PCA) allowed the identification of samples inconsistent with the others, so the models were remade and compared with the previous ones. This way the best modeling was found based on the variables selected by Random Forest with the database without the problematic samples. An important point of this work is that it does not replace the classic analyses of slope stability, on the contrary, it contributes to engineers and geologists with a tool for monitoring the stability conditions of slopes in a mining operation. It is known that the analysis of slope stability should be done throughout the life of the mine and, therefore, it is believed that the tool proposed here can optimize the selection of slopes most susceptible to instability.

KEY WORDS: Mine slope stability conditions. Geotechnical database. Decision trees. Random Forest. Geomechanical parameters.

ÍNDICE DE ILUSTRAÇÔES

ÍNDICE DE TABELAS

ÍNDICE DE SIGLAS

MG	Modelo Geral
MM	Modelo Matemático
MQ	Modelo Q-slope
MS	Modelo Santos (2020)
RF	*Random Forest*
OOB	*Out-of-Bag*
MSII	*Mine Slope Instability Index*
DLDA	*Diagonal Linear Discriminant Analysis*
PCA	*Principal Component Analysis*

SUMÁRIO

1. INTRODUÇÃO

Os taludes em mineração são estruturas utilizadas na maior parte de tempo do empreendimento, inseridos nas fases de desenvolvimento, explotação e fechamento de mina. Estas estruturas auxiliares, os taludes, permitem as condições mínimas de operacionalidade como por exemplo o transporte do material, desmonte de rochas e acesso de equipamentos e principalmente criar acessos para alcançar os diferentes níveis da reserva mineral maximizando a sua recuperação. A correta determinação do ângulo do talude é de grande importância em qualquer operação mineraria a céu aberto, já que este tem influência direta em todas as dinâmicas produtivas do planejamento de mina. Desde a perfuração e desmonte dos maciços, seguindo para o tamanho dos equipamentos utilizados nas operações, o transporte desse material desmontado e a estabilidade final dos taludes, são determinantes para a escolha da geometria do talude.

Além disso, ao utilizar taludes com maior ângulo as mineradoras conseguem um melhor aproveitamento do espaço, possibilitam uma recuperação maior do material ao mesmo tempo evitando a retirada de substâncias sem valor econômico, otimizando a relação estéril-minério. Porém há um fator determinante que limita este ângulo: inclinações mais íngremes, resultam em um aumento do risco de rupturas. A ocorrência de ruptura impacta diretamente as atividades mineradoras, como interrupção de vias de acesso, diluição no minério, custos adicionais com a reprogramação das operações, realocação de equipamentos para reabilitação da área impactada, além disso destaca-se possíveis perdas de vidas, danos em estruturas e de equipamentos.

O objetivo geral da presente pesquisa é o desenvolvimento de um modelo capaz de interpretar as informações retiradas de taludes e gerar como resposta uma estimativa confiável acerca das condições de estabilidade do maciço rochoso. A utilização da técnica de árvores de decisão permite que diferentes usuários, público alvo em geral, aplique o modelo de maneira rápida e precisa, mesmo este não possuindo os conhecimentos específicos de aprendizado de máquinas.

As árvores de decisão se tornaram um dos métodos mais usados para a criação de algoritmos de inferência e tem grande aplicabilidade em diversas áreas como, por exemplo, diagnóstico médico e risco de crédito e no caso da mineração, a criação de métodos de validação de estabilidade de taludes a exemplo desta pesquisa. A escolha deste método para este trabalho foi feita pelos seguintes motivos: consegue manipular de forma eficiente os

diferentes “sub-espaços” criados possibilitando uma boa resposta; é facilmente usado em diversos bancos de dados; gera resultados com altíssima confiabilidade; pela grande inteligibilidade dos resultados que produz.

Para isso, será utilizado o banco de dados proposto por Zare Naghadehi *et al.* (2013). No trabalho citado os autores apresentam o desenvolvimento de um novo Índice de Instabilidade de Taludes de Mina (MSII) que tem como objetivo a determinação das condições de estabilidade de taludes em mineração em operações a céu aberto, utilizando redes neurais artificiais e o sistema RES proposto por Hudson (1992).

No banco de dados de Zare Naghadehi *et al.* (2013), dezoito parâmetros relacionados a estabilidade de taludes, são empregados para a definição do índice desenvolvido na pesquisa. Esses parâmetros levam em conta desde as condições naturais do talude como por exemplo o tipo de rocha, RQD, condições hidráulicas no maciço e propriedades das descontinuidades; condições climáticas como precipitação e clima da região e por fim as características de construção do talude como método de desmonte e geometria aplicada.

Por isso, é de suma importância uma constante determinação da integridade dos taludes no decorrer da vida útil de uma mina para evitar riscos na operação. Pensando neste problema, esse trabalho tem como objetivo testar um novo modelo para interpretação e análise de taludes a partir do uso do método de Árvore de Decisões juntamente com o uso de validadores de análise discriminantes que serão melhor explicados no decorrer do texto.

2. REVISÃO BIBLIOGRÁFICA

2.1.Conceitos iniciais

Para a mecânica das rochas o maciço rochoso é um conjunto de blocos do material rocha, por vezes denominado, indevidamente, de rocha intacta, definidos pela interseção de conjuntos (denominados famílias) de superfícies subparalelas, regularmente planares, de origem geológica qualquer, por exemplo: juntas, fraturas, falhas, estratificações, clivagens, que são coletivamente designadas por descontinuidades. Portanto de maneira resumida o maciço rochoso é um conjunto de blocos de rocha definidos pela interseção de famílias de descontinuidades.

Os maciços rochosos são objetos de estudo das Engenharias Geotécnica, de Minas e Civil. Eles consistem em unidades geológicas que compõem a superfície do nosso planeta e podem ser formados por conjuntos de rochas variadas. Para caracterizar o maciço rochoso coleta-se parâmetros provenientes da rocha intacta e das famílias de descontinuidades que o compõe. Para determinação de parâmetros de resistência do maciço rochoso, sistemas de classificação são aplicados.

Talude é qualquer superfície com uma inclinação em um maciço constituído por solo e/ou rocha. Estas estruturas podem ser de origem natural, sendo comumente denominados de encostas, ou construído pelo homem com a finalidade de estabilização de uma estrutura em rocha ou solo, como, por exemplo, os aterros e cortes. Associados a estes taludes, existem parâmetros que determinam as condições físicas e estruturais do maciço. Estes indicadores são denominados de parâmetros geomecânicos.

2.2. Parâmetros geomecânicos

Na realização de qualquer projeto de obra geotécnica leva-se em conta diversos parâmetros geomecânicos das formações rochosas que são necessários para análise de estabilidade. Estes parâmetros são definidos com base em campanhas de levantamento de dados e amostras em campo que são posteriormente submetidas a ensaios com técnicas de caracterização *in situ* e laboratoriais.

De acordo com os resultados dessas campanhas, cada área recebe sua classificação geotécnica estabelecendo as propriedades para cada zona. Este é um exercício com certo grau de subjetividade, porém seu resultado é de extrema importância para os próximos estágios de qualquer empreendimento que queira trabalhar com esses tipos de problema (Pinheiro, *et.al* 2016).

Por causa dessa importância foram desenvolvidos diversos sistemas de classificação geomecânica como *Rock Mass Rating* (RMR) de Bieniawski (1989) e Q-*slope* de Barton *et al*. (1974), o SMR proposto por Romana (1985) e o RMi proposto por Palmström (1995). Os sistemas de classificação são aplicados para determinação de parâmetros de resistência do maciço rochoso que são *inseridos*em análises de estabilidade.

Segundo Avila (2012), os sistemas de classificação de maciços rochosos além de criarem os indicativos para a operação do maciço, muitas vezes, possibilitam também o desenvolvimento de uma base de dados para se estimar as propriedades mecânicas, como a deformabilidade e a resistência dos maciços rochosos.

A maioria dos sistemas de classificação multi-parâmetro (Barton *et al*. , 1974) foram desenvolvidos a partir de bancos de dados de obras em engenharia na qual todos os componentes do maciço rochoso foram incluídos.

2.2.1. PARÂMETROS NATURAIS DE ROCHA

A rocha é constituída por uma junção quase que totalmente compacta de grãos cristalinos podendo apresentar em sua composição componentes de matéria amorfa, além de apresentar descontinuidades ou vazios, microfissuras, existentes entre esses grãos. As propriedades desse material são influenciadas pela composição química desses grãos e sua mineralogia, granulometria e disposição espacial das partículas e por fim pela forma, quantidade e distribuição das descontinuidades ou vazios.

Essas características do material determinam importantes parâmetros que alteram o comportamento da rocha intacta em função das modificações do ambiente no entorno do maciço.

A resistência mecânica é a propriedade de um sólido de contrapor uma tensão aplicada nele impedindo o surgimento de uma ruptura, essa tensão pode ser de natureza estática ou dinâmica. Quando a força aplicada na rocha ou maciço alcança um certo limite intrínseco ao material no qual o solido inicia o processo de ruptura segundo os mecanismos de cisalhamento, tração ou compressão é chamado de tensão de ruptura. Esse parâmetro é muito usado para determinar as capacidades estruturais do maciço rochoso além de influenciar outros parâmetros geotécnicos.

Este parâmetro pode ser obtido com testes ensaios de laboratório de acordo com tensões aplicadas à corpos de prova para obtenção de dados como tensão cisalhante, normal e ângulo de atrito interno (De Toledo *et al.*1993).

Condições naturais do ambiente e as modificações causadas pelas atividades humanas de engenharia alteram as condições de estabilidade de taludes que caso não sejam bem aplicadas podem resultar em graves acidentes como as rupturas. Por isso, é de grande importância determinar as condições de resistência dessas estruturas que são muito usadas em empreendimentos de mina a céu aberto (Tao *et al*. 2018).

Muitas definições acerca da alteração da rocha têm aparecido na literatura a exemplo de Fookes, (1988), embora todos reconheçam a importância da interação da hidrosfera e atmosfera em maciços rochosos, além é claro do fator de tempo considerado geralmente numa escala geológica. A capacidade de um material rochoso de sofrer intempéries ou degradar é dependente principalmente de um afastamento do ambiente de sua formação.

Os dois processos dominantes que alteram a rocha incluem intemperismo físico, que resulta na desagregação de rochas sem mudança mineralógica e intemperismo químico, resultando na decomposição dos minerais constituintes estáveis ou produtos minerais secundários metaestáveis (Fookes, 1988).

De acordo com Fookes (1988), a alteração do material do maciço rochoso no tempo geológico tem uma influência direta e importante na durabilidade em estruturas de engenharia como taludes e quando estas estruturas estão diretamente expostas a estas intempéries, a alteração do maciço se torna mais evidente em períodos de tempo mais curtos. Sendo assim, é muito importante avaliar o grau de alteração para a determinação da estabilidade do talude.

Rock-quality designation (RQD) (Deere, 1964) é uma medida aproximada do grau de descontinuidades ou fraturas em um maciço rochoso, medida como uma porcentagem do núcleo de perfuração em comprimentos de 10 cm ou mais. A rocha de alta qualidade tem um RQD de mais de 75% e a rocha de baixa qualidade de menos de 50%.

A definição mais amplamente usada foi desenvolvida por Deere (1964). A porcentagem de recuperação do testemunho de sondagem que incorpora apenas partes do testemunho com mais de 100 mm de comprimento medido ao longo da linha central do núcleo são usadas para a determinação deste parâmetro. O RQD é um elemento básico que é muito usado em sistemas de classificação de maciço rochoso como o RMR de Bieniawski (1989) e o *Q-system* (Barton N**.** & Bar N., 1974).

2.2.2. PARÂMETROS DE DESCONTINUIDADES

As propriedades empregadas para caracterizar descontinuidades estão descritas abaixo:

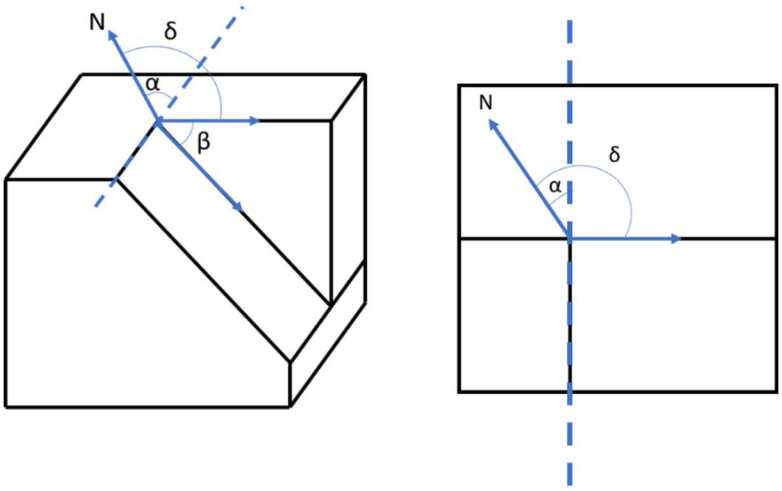
Orientação das descontinuidades: atitude da camada de descontinuidade, em geral dado pelo mergulho da camada de descontinuidade e direção de mergulho. Na análise cinemática a orientação das famílias de descontinuidades em conjunto com a orientação da face do talude permite a determinação de mecanismos de ruptura. Na Figura 1 estão representadas as principais características que definem a orientação das descontinuidades como, α (direção da camada), δ (direção de mergulho) medidos a partir do Norte e β (ângulo do mergulho).

Número de famílias de descontinuidades: este é o número de conjuntos principais de descontinuidade no maciço rochoso. A forma do bloco do maciço rochoso é afetada por esta propriedade sendo que as descontinuidades são agregadas nessas famílias de acordo com a semelhança na direção e mergulho dessas descontinuidades.

Persistência da descontinuidade: a persistência reflete o comprimento das descontinuidades que influenciam fortemente o tamanho dos blocos que podem ser formados.

Espaçamento das descontinuidades: distancia perpendicular entre duas descontinuidades de mesma família. De acordo com Zare Naghadehi *et al.* (2013), o espaçamento das descontinuidades afeta o tamanho dos blocos no maciço e seu comportamento geral. Por exemplo, várias descontinuidades espaçadas próximas tendem a aumentar a instabilidade do maciço com a redução da coesão, enquanto que descontinuidades muito espaçadas favorecem o travamento destes blocos formados pelas próprias descontinuidades.

Figura 1 - Representação da orientação da descontinuidade no maciço rochoso em vistas de perfil e planta respectivamente.



Abertura de descontinuidade: A abertura é determinada pela distância perpendicular entres as superfícies das paredes de uma descontinuidade aberta. Esta abertura possibilita o aumento principalmente da infiltração de água no maciço que podem alterar o maciço rochoso além é claro de influenciar a saturação da rocha local.

Rugosidade das descontinuidades: Forma e tipo de superfície do plano de descontinuidade afetando fortemente na resistência ao cisalhamento das descontinuidades e a estabilidade das escavações em maciços. Quanto maior esta rugosidade, maior deverá ser a força (peso) do bloco para tirar este do repouso, ou seja, maior a estabilidade do maciço.

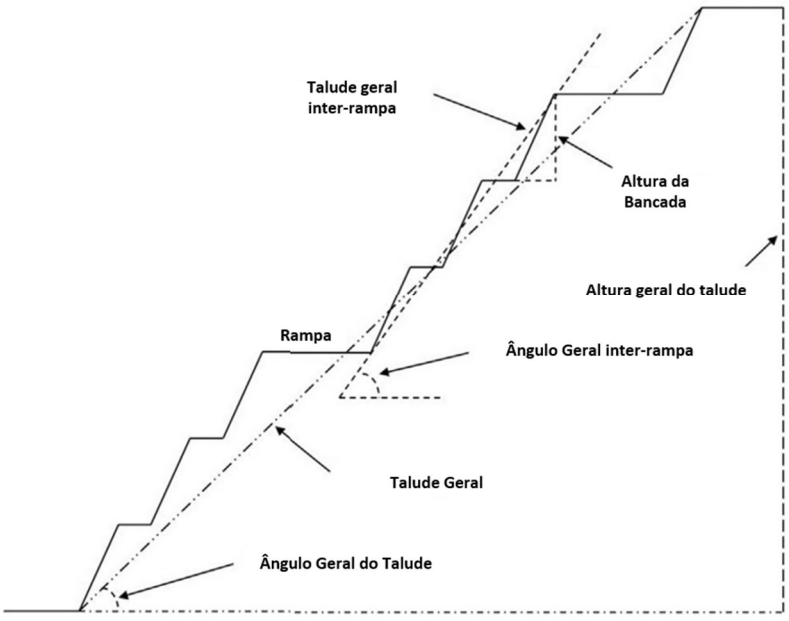
Preenchimento das descontinuidades: Os preenchimentos podem ter um significativo impacto na resistência das descontinuidades. Para uma correta determinação do efeito deste preenchimento para a estabilidade dos blocos é importante que seja feito uma avaliação técnica do material constituinte, caso tenha boas condições de resistência, este preenchimento irá influenciar positivamente na estabilidade dos blocos do maciço rochoso (Zare Naghadehi *et al*., 2013).

2.3. Geometria dos taludes

Uma geometria típica de um talude de mina a céu aberto está mostrada na Figura 2. Para representar esta geometria, dois parâmetros podem ser usados:

- Ângulo geral do talude: O ângulo do talude desempenha um importante papel em relação a sua estabilidade e na relação estéril-minério. Com o aumento do ângulo de um talude há um aumento do potencial de rupturas, tornando os blocos removíveis mais propensos a falha. Em contrapartida uma redução do ângulo de talude aumenta a relação estéril-minério.
- Altura geral do talude: Blocos de rocha em taludes mais altos têm mais energia potencial do que rochas em taludes mais baixos, sendo assim eles estão mais propensos a apresentarem instabilidade (Zare Naghadehi *et al*., 2013).

Figura 2 - Representação geral de um talude de mina.



Fonte - Adaptado de Zare Naghadehi *et al*. (2013).

2.4. Demais variáveis do banco de Zare Naghadehi *et al*. (2013).

Por fim, outros parâmetros usados para determinar a estabilidade de taludes estão associados com sua construção, histórico de instabilidade, regime tectônico e a presença de água no maciço rochoso:

- Método de desmonte: danos às faces da rocha causados por repetidas explosões realizadas em operações de mina a céu aberto pode causar um aumento da instabilidade do talude, sendo importante contabilizar o efeito deste parâmetro.
- Instabilidades passadas: a presença de instabilidades anteriores demonstra a existência de combinações críticas de fatores que levaram a instabilidades e falhas no maciço rochoso podendo ser usadas como evidenciadores de instabilidades futuras (Zare Naghadehi *et al*. 2013).
- Regime tectônico: Massas rochosas são submetidas a tensões *in situ* do peso dos estratos de sobreposição e de tensões tectônicas. O regime tectônico influencia essas tensões principalmente quando estes taludes estão localizados em regiões próximas de encontros de placas tectônicas. Isso é demonstrado pelo *World Stress Map* (2008), que mostra que a orientação da tensão horizontal máxima depende de sua localização nestas placas (Zare Naghadehi *et al*. 2013).
- Água subterrânea: a água subterrânea em um talude de rocha diminui a sua estabilidade, pois a água atua reduzindo a resistência ao cisalhamento ao reduzir o estresse atuante na rocha, além disso a água também serve como um fator a mais para o aumento do desgaste do maciço rochoso.

2.5. Geometria dos taludes e mecanismos de ruptura

Os conhecimentos sobre o comportamento dos mecanismos de ruptura de taludes de rocha aumentaram consideravelmente durante a última década em resposta ao desenvolvimento de soluções mais econômicas na exploração de grandes minas a céu aberto. Estas demandas aumentaram consideravelmente os esforços em cima do uso de taludes mais altos e íngremes exigindo considerações da geologia estrutural desde a microescala até a escala tectônica regional (Stead & Wolter, 2015).

As características geológicas de uma região, como dobras, falhas e descontinuidades, são de grande importância para determinar o comportamento do talude e contribuem para a estabilização ou desestabilização de taludes rochosos.

Os mecanismos de ruptura ocorrem de acordo com a relação entre a orientação da face do talude e das famílias de descontinuidades. Os principais mecanismos são classificados como: planar, cunha, tombamentos e circular.

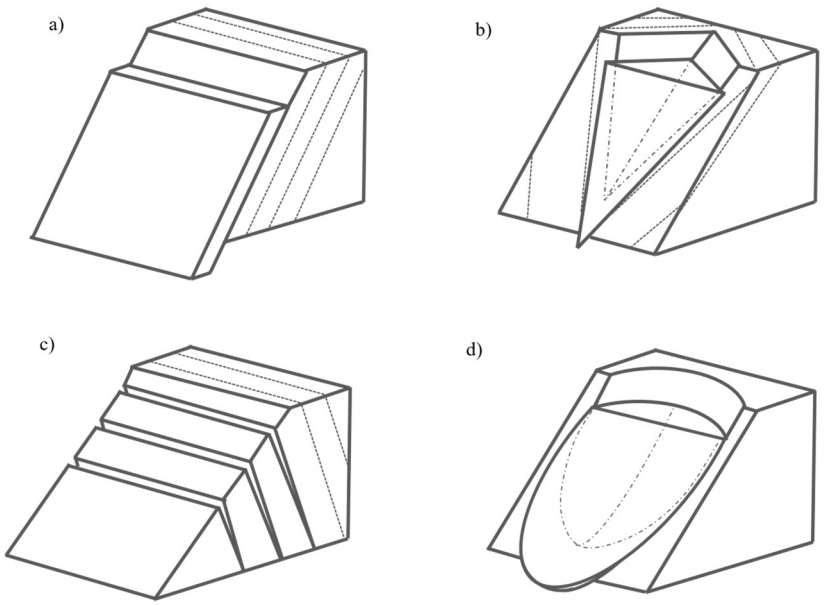
Na ruptura planar, a família de descontinuidade mergulha para a mesma direção do talude, caso o maciço não tenha resistência para sustentar a camada este plano escorrega causando o deslizamento.

Na ruptura em cunha, a intersecção de duas famílias forma uma estrutura em formato de cunha que, caso esteja na mesma direção do talude, pode escorregar causando um dano a estrutura original do maciço.

Já no caso do da ruptura por tombamento, as descontinuidades estão em um mergulho oposto ao do talude. Esta configuração pode fazer com que estes blocos formados pelas descontinuidades tombem formando escadas nos taludes, prejudicando a estrutura do maciço.

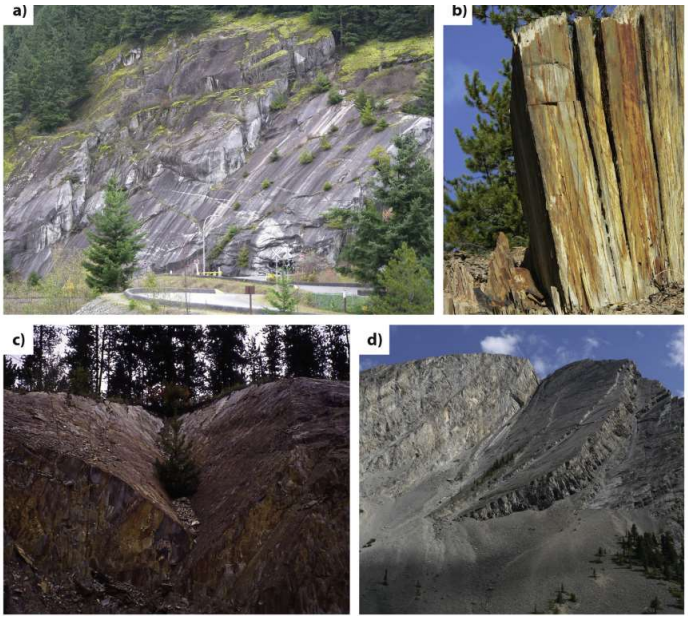
Por fim, a ruptura circular ocorre quando uma grande massa de rocha ou solo desliza em formato circular. Este mecanismo é muito comum em taludes formados por solos por causa de sua estrutura pouco coesa, porém também pode ocorrer em maciços rochosos quando estes estão muito fraturados. Outro fenômeno que pode acarretar em uma ruptura circular ocorre quando se leva em conta o fator da escala do maciço rochoso. Quando se analisa grandes taludes esse efeito de escala altera a forma como as descontinuidades se relacionam podendo causar ruptura. O efeito escala será abordado mais a fundo no próximo capítulo deste trabalho. Esses mecanismos estão representados na Figura 3 onde é possível ver como a orientação das descontinuidades (linha tracejada) se comporta em relação ao talude numa situação de ruptura.

Figura 3 - Mecanismos de ruptura com representação das descontinuidades (linha tracejada) e projeção da ruptura (ponto traço); a) Planar; b) Cunha; c) Tombamento; d) Circular.



O efeito dessas rupturas no ambiente pode ser visto na figura 4 onde é possível observar diferentes mecanismos atuando nos maciços rochosos em diferentes regiões do mundo.

Figura 4 Principais modos de ruptura de rochas considerados na análise de estabilidade de taludes: a) ruptura planar, b) tombamento e c) ruptura em cunha. d) Falha translacional multiplanar.



Fonte - (Stead & Wolter, 2015).

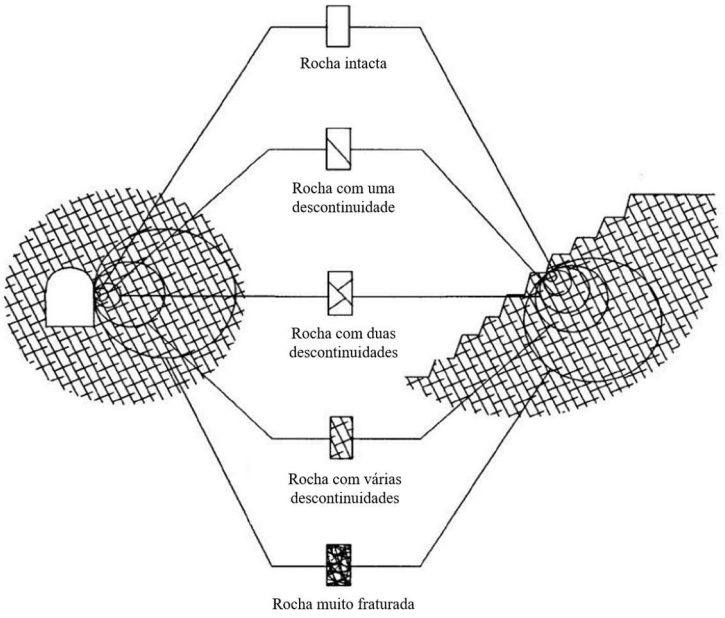
2.6. Efeito escala

O efeito escala em mecânica de rochas pode ser definido como a influência das descontinuidades no mecanismo de ruptura levando em consideração a escala do projeto. Portanto a depender da dimensão do talude a distribuição das descontinuidades determina os mecanismos de ruptura, possibilitando a interpretação do problema como um meio contínuo, meio fraturado, meio muito fraturado, meio extremamente fraturado e o meio contínuo equivalente.

Nesse contexto, a superfície de ruptura em um talude pode consistir de um sistema simples formado apenas por um plano continuo ou em casos mais complexos podem formar superfícies mais complexas com a interação de várias famílias de descontinuidades dentro do maciço rochoso. Portanto, segundo Hoek (2002), a seleção dos parâmetros apropriados para caracterização de um talude, vai depender da escala relativa entre o plano do talude e as estruturas de descontinuidade no maciço rochoso.

Baseado nos efeitos escala e as condições geológicas mencionados previamente, pode-se determinar as condições de estabilidade apropriadas adequando-se aos objetivos finais do talude. O efeito da escala pode ser visto na Figura 5, onde se nota como a proporção da análise das descontinuidades altera a validação final das condições de descontinuidade do maciço.

Figura 5 - Diagrama idealizado mostrando transição desde rocha intacta até o maciço rochoso fraturado com o incremento do tamanho de amostra.



Fonte - (adaptado de Hoek, 2002).

2.7. Aprendizado de máquinas

2.7.1. CONSIDERAÇÕES INICIAIS

O aprendizado de máquina é uma categoria da inteligência artificial, com um conjunto de técnicas que permitem a criação de modelos matemáticos usando como informações bases de dados. Os modelos podem ser preditivos ou descritivos.

Tradicionalmente a amostra original é particionada em duas partes, uma chamada de "dados de treinamento" e outra parte chamada de "dados de teste". Os dados de treinamento são utilizados para treinar o modelo e os dados de teste para testar o modelo. As métricas de avaliação dos modelos são realizadas utilizando os dados de teste.

Para a utilização deste método é importante determinar quais são as variáveis preditoras e qual é a variável *target* dos dados analisados. As variáveis preditoras são aquelas usadas como valores de entrada no modelo, que foram usadas para treinar, testar e estimar a variável *target.* Já a variável *target* é o valor, ou categoria, de saída do modelo, que determina a resposta final de cada amostra avaliada.

Os dados de treinamento para modelos preditivos são compostos pela união do conjunto de variáveis junto com um conjunto resposta para essas variáveis. Com essas informações é possível criar um modelo para prever o comportamento do conjunto resposta em relação à resposta das variáveis. Para validar esse modelo são usados os dados de teste para comparar os resultados experimentais do modelo com as respostas contidas nesse novo banco de dados (Zhang, 2020).

Os modelos preditivos, podem ser classificados em dois tipos: Modelos de Regressão e Modelos de Classificação. Os Modelos de Regressão possuem uma variável *target* composta por dados do tipo numéricos, ou seja, dados que aceitam valores fora do intervalo dos dados usados no treino do modelo. Já os Modelos de Classificação possuem uma variável *target* restrita aos valores predeterminados pelo conjunto de dados usados no seu treino, ou seja, os dados de saída são fatores onde não há meio termo em seus níveis de resposta. Os principais modelos deste tipo podem ser vistos na Tabela 1 além do tipo de modelo que eles são de acordo com suas funções.

Tabela 1 - Modelos e suas principais funções.

Modelo	Função
Algoritmo de aprendizado supervisionado	
Vizinho mais próximo	Classificação
Naive Bayes	Classificação
Árvore de Decisão	Classificação
Classificador por regras de aprendizagem	Classificação
Regressão Linear	Regressão numérica
Árvore de Regressão	Regressão numérica
Modelo de Árvores	Regressão numérica
Rede Neural	Ambos os usos
Máquina de Vetores de Suporte	Ambos os usos
Algoritmo de aprendizado não supervisionado	
Regra de associação	Detecção de padrões
K-*means* por análise de agrupamento	Análise de agrupamento

Fonte - Adaptado de Lantz, (2013).

2.7.2. TIPOS DE APRENDIZADO

De acordo com Zhang (2020), os tipos de aprendizado de máquina podem ser classificados dessas formas:

A. Aprendizagem regular ou euclidiana de dados estruturados.

- Aprendizagem supervisionada: a partir de um conjunto de dados de treinamento que apresenta as variáveis de entrada e a variável target, o algoritmo interpreta o conjunto de regras criado pelo banco de dados (conjunto de treino) e mapeia as informações rotuladas (variável *target*). A partir desse conjunto de regras o modelo consegue interpretar novas informações (conjunto de teste), como um "professor" ou supervisor dando a um aluno um problema e suas soluções e dizendo ao aluno para descobrir como resolver outros problemas semelhantes.

- Aprendizagem não supervisionada: a aprendizagem não supervisionada, usa como conjunto de treino um banco de dados sem a variável *target*, ou seja, apenas as informações das variáveis de entrada são usadas para o desenvolvimento do modelo. Usando a mesma metáfora da classificação anterior, o trabalho do aluno será encontrar uma solução própria para o problema tentando encontrar um padrão nas informações disponibilizadas sem a ajuda de um professor.

- Aprendizagem semi-supervisionada: um conjunto de dados é fornecido ao modelo, porém diferentemente dos dois métodos anteriores, este banco de dados possuí tanto

informações rotuladas quanto não rotuladas. No entanto, o primeiro tem uma frequência muito inferior se comparado com o segundo. Esta aprendizagem fica entre a aprendizagem não supervisionada (sem quaisquer dados de treinamento) e aprendizagem supervisionada (com dados de treinamento completamente rotulados). De acordo com a forma que as informações são dispostas e rotuladas há subclassificações para esta aprendizagem:

  - O autotreinamento utiliza suas próprias previsões para ensinar a si mesmo e aprimorar suas previsões.
  - O co-treinamento usa as suas previsões para ampliar o conhecimento sobre o próprio modelo criado pelo método.
  - A aprendizagem ativa faz com que o modelo preditivo solicite e consiga de forma ativa determinar quais pontos devem ser rotulados para constituir o conjunto de dados de treinamento.

- Aprendizagem por reforço: os dados de treinamento (na forma de recompensas e punições) são fornecidos apenas como resposta à previsão feita por algum sistema baseado em inteligência artificial em um sistema que se alimenta com cada novo *feedback*.

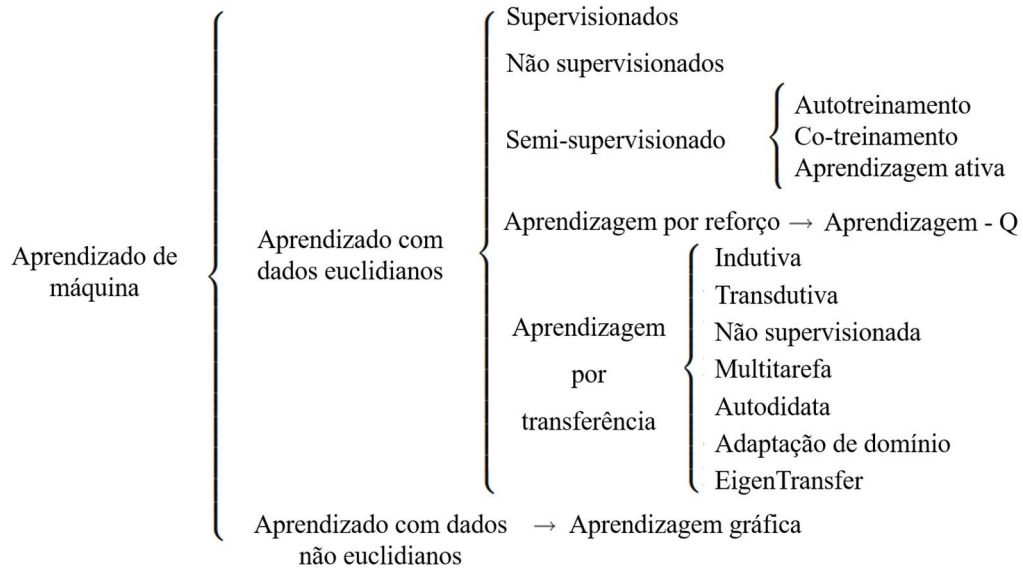
- Aprendizagem por transferência: as informações armazenadas podem ser consideradas datadas ou desatualizadas com o passar do tempo, sendo necessário a substituição dessas informações, esta aprendizagem leva em conta este problema e consegue selecionar os dados ainda relevantes de acordo com as novas informações geradas ou fornecidas. A aprendizagem por transferência inclui, mas não se limita a, aprendizagem por transferência indutiva, aprendizagem por transferência transdutiva, aprendizagem por transferência não supervisionada, multitarefa aprendizagem, aprendizagem de transferência autodidata, adaptação de domínio e *EigenTransfer*.

B. Aprendizado com dados não euclidianos

- O aprendizado de máquina gráfica é um aprendizado de dados estruturados irregular que não respeita as definições euclidianas clássicas dos outros métodos, esta aprendizagem funciona a partir da estruturação gráfica das informações usando o banco de dados de treinamento podendo esse ser formulado de forma semi-supervisonada ou não supervisionada.

Na Figura 6 está representada a organização dos tipos de aprendizagem com suas respectivas categorias em uma distribuição hierárquica em árvore.

Figura 6 - Árvore hierárquica da classificação de aprendizado de máquina.



Fonte - (adaptado de Zhang, 2020).

Após apresentar estes tipos de aprendizado é possível abordar os métodos usados para o modelamento dos dados de acordo com o problema abordado.

2.7.3. TÉCNICAS DE APRENDIZADO DE MÁQUINA

Existem muitos métodos para se aplicar no aprendizado de máquina para a modelagem dos dados, sendo que cada método é usado para se alcançar a solução para tipos de problemas específicos (Zhang, 2020). Esses métodos de aprendizado de máquina podem ser divididos em seguintes tipos:

1. Métodos de aprendizado de máquina baseados em estrutura de rede.

- Redes Neurais Artificiais (RNAs): este método consiste em simular uma comunicação entre neurônios, cada um desses recebe uma informação de entrada e calcula um valor de saída que é encaminhado para o próximo neurônio onde este mesmo processo será repetido, porém com outro tipo de cálculo até que este dado alcance a última camada neural onde a saída será a resposta do modelo.

- Rede Bayesiana: uma rede Bayesiana é um modelo gráfico probabilístico que representa as variáveis e as relações entre elas. A rede é formada por diversos nós que mantem a aleatoriedade dos estados das variáveis e a forma de condicional de probabilidade, além disso estes nós geram a função gráfica dirigindo está para um modelo acíclico para que haja uma interpretação deste. Pela complexidade deste método é necessário um conhecimento especializado no assunto ou o desenvolvimento de algoritmo eficiente para a geração e posicionamento desses nós.

2. Métodos de Aprendizado de Máquina baseados em Análise Estatística.

- Regras de associação: o objetivo da associação é descobrir regras que interligam as variáveis de um banco de dados. Este processo utiliza duas métricas que indicam a frequência de uma determinada relação e quão frequente o conjunto de itens aparece no banco de dados, com isso é gerada uma confiança em relação ao número de vezes que esta regra desenvolvida foi considerada verdadeira.

- *Clustering*: é um conjunto de técnicas que tenta adequar dados não rotulados de bancos de dados com grande número de informações. É uma abordagem de descoberta de padrões não supervisionada onde as informações são organizadas de acordo com similaridades entre as medidas obtidas.

- *Ensemble Learning*: neste método diversos algoritmos são usados para melhorar a capacidade preditiva, muitas vezes o modelo usa vários modelos fracos para conseguir criar modelos considerados fortes.

- Modelos ocultos de Markov (HMMs): um HMM é uma forma estatística de modelagem baseada no modelo de Markov onde os estados das variáveis estão ocultos. O principal desafio é determinar quais dos parâmetros devem permanecer observáveis e quais deles devem ser ocultados. Por ter uma saída de respostas com uma distribuição probabilística diferente em cada estado é possível o sistema se alterar ao longo do tempo.

- Aprendizagem indutiva: A aprendizagem indutiva visa aprender um modelo a partir de exemplos marcados e tenta prever os rótulos de um banco de dados com informações ainda não categorizadas. Após a realização de observações especificas, surgem padrões e regularidades usados pelo modelo para criar hipóteses experimentais a serem exploradas.

- Naive Bayes: Classificadores Naive Bayes são usados para lidar com banco de dados com um número arbitrário de recursos independentes, reduzindo uma tarefa de estimativa muito densa para uma estimativa com densidade unidimensional de Kernel partindo da independência dos dados.

3. Métodos de aprendizado de máquina baseados em evolução.

- Computação Evolutiva: é uma família de algoritmos para otimização global que usa como princípio básico a evolução biológica. O termo engloba algoritmos genéticos, programação genética, estratégias de evolução, otimização de enxame de partículas entre outros, sendo que todos utilizam preceitos da biologia e do comportamento natural da vida para desenvolver os algoritmos.

Conhecendo estes métodos, é possível utilizá-los em diversas áreas de atuação onde há a necessidade de previsões estatísticas para validar planos operacionais.

2.7.4. APLICAÇÕES NA ENGENHARIA DE MINAS

Como em diversas atividades, a aplicação de aprendizado de máquinas vem crescendo na mineração. Por ser uma atividade que envolve diversas variáveis operacionais e necessita de otimização de processos constantemente baseado nas alterações do ambiente e do mercado é necessário que haja um acompanhamento estatístico de variabilidade desses parâmetros. Em função disto o uso de aprendizado de máquina tem se popularizado neste ramo, desde a pesquisa mineral até as etapas finais de comercialização. Alguns estudos podem ser vistos na Tabela 2.

Tabela 2 - Relação de estudos com aprendizado de máquina na mineração.

ESTUDOS COM APRENDIZADO DE MÁQUINA APLICADOS NA MINERAÇÃO			
Título	**Autores**	**Modelos**	**Data de publicação**
A comparative study of empirical and ensemble machine learning algorithms in predicting air over-pressure in open-pit coal mine	Nguyen, *et al.* (2020)	*Gradient boosting Machine* (GBM); *Random Forest*; *Cubist.*	Janeiro de 2020
Prediction of flyrock in open pit blasting operation using machine learning method	Manoj *et al*. (2013)	*Máquinas de vetores de suporte.*	Maio de 2013
Machine Learning-Based Driving Style Identification of Truck Drivers in Open-Pit Mines	Wang, *et al*. (2019)	*Máquinas de vetores de suporte; Random Forest; K-nearest neighbor; Rede Neural.*	Dezembro de 2019
Automated lithological classification using UAV and machine learning on an open cast mine	Beretta *et al*.(2019)	*Máquinas de vetores de suporte; Random Forest; K-nearest neighbor; Gradient Tree Boost.*	Janeiro de 2019
Fuzzy Algorithm of discontinuity sets	Klen & Lana (2014)	*Fuzzy K-means.*	Dezembro de 2014

- Nguyen, *et al*. (2020)

Este estudo visou levar em consideração a viabilidade de três algoritmos de aprendizado de máquina de conjunto para prever a sobrepressão de ar induzida pela explosão (AOp) em mina a céu aberto, incluindo *Gradient boosting Machine* (GBM), *Random Forest* (RF) e *Cubist*. Uma técnica empírica também foi aplicada para predizer AOp e comparada com os modelos *ensemble*. Para empregar este estudo, 146 eventos de detonação foram investigados com 80% do banco de dados total (aproximadamente 118 eventos de detonação) sendo usados para desenvolver os modelos, enquanto o resto (20% ~ 28 detonações) foram usados para validar a precisão dos modelos. Além de resultar em modelos com boa confiabilidade, outras descobertas indicaram que a capacidade de carga explosiva, espaçamento, avanço, distância de monitoramento e umidade do ar foram as entradas mais importantes para os modelos preditivos AOp usando inteligência artificial (Nguyen, *et al*. ,2020).

- Manoj *et al.* (2013)

O Ultralançamento é um dos eventos mais perigosos na operação de detonação de minas de superfície que ocorre quando grandes fragmentos de rocha são lançados a grandes distâncias no momento da detonação. A existência de vários parâmetros eficazes e suas relações desconhecidas são as principais razões para a imprecisão dos modelos empíricos. Em seu artigo, os autores realizaram uma tentativa de prever o ultra lançamento em operações de detonação da Mina de Cobre Soungun, no Irã, incorporando propriedades da rocha e parâmetros de projeto de detonação usando o método de Máquina de Vetor de Suporte (SVM). Resultando num modelo com uma melhor acurácia que os modelos empíricos convencionais usados até o momento.

- Wang *et al.* (2019)

A importância da construção de um modelo de identificação do estilo de direção para os motoristas de caminhões de minas a céu aberto é reduzir o consumo de diesel e melhorar o treinamento. Com base nisso, os dados obtidos foram aplicados em modelos como *de Random Forest, K-nearest neighbor*, máquina de vetor de suporte e modelos de rede neural. Os dados foram otimizados e a precisão foi comparada por meio de uma pesquisa de grade de validação cruzada e, em seguida, um modelo de identificação de estilo de condução com base na *Random Forest* foi finalmente proposto pelos autores. Os resultados mostram que os modelos de identificação do estilo de direção baseados em *Random Forest* podem efetivamente identificar diferentes estilos de direção quando o caminhão de mineração está operando sob carga pesada e sem carga, e a precisão geral do modelo foi de 95,39% e 90,74%, respectivamente. O consumo de combustível do estilo de direção agressivo foi o maior e foi 10% maior do que o consumo médio de combustível (Wang*et al.* 2019).

- Beretta *et al*. (2019)

O planejamento da mina é diretamente dependente das características litológicas e da definição dos contatos entre os materiais. A modelagem geológica é uma tarefa contínua que é realizada usando dados de observação, que incluem informações de faces abertas. Veículos aéreos não tripulados (UAVs) são amplamente utilizados em projetos de mineração a céu aberto, com baixo risco para os operadores, para a aeronave ou terceiros. Dessa forma os autores desenvolveram um sistema de estimativa para detecção das litologias de uma mina brasileira de minério fosfático utilizando imagens áreas. Após a análise dos autores, um

modelo utilizando o *Random Forest* foi selecionado, resultando em uma precisão de 59%, porém novos estudos foram propostos por eles com o objetivo de utilizar uma Rede Neural para aumentar a precisão do modelo.

- Klen & Lana (2014)

O agrupamento de descontinuidades em famílias e a identificação de seu valor médio de orientação são tarefas importantes na engenharia geotécnica, sendo que as famílias de descontinuidades controlam o comportamento mecânico e hidráulico dos maciços rochosos e por consequência a sua estabilidade. O agrupamento nem sempre é uma tarefa trivial, particularmente quando se utiliza apenas o diagrama de densidade de polos, método clássico, porém muito subjetivo em sua interpretação. Com o objetivo de contornar essa dificuldade, é possível utilizar métodos numéricos para resolver esse dilema, para isso, os autores propuseram um algoritmo baseado no Método *Fuzzy K-means*, que permite reunir as descontinuidades em famílias eliminando o critério subjetivo dos métodos tradicionais. Ao fim da pesquisa, o algoritmo teve seus resultados comparados com dois conjuntos de fraturas estudados na literatura e demonstrou-se eficiente para a determinação das famílias de descontinuidades.

2.8.Árvores de Decisão

2.8.1. CONCEITOS INICIAIS

Árvores de decisão, em uma explicação simples, é uma abordagem usada para criar modelos a partir de um banco de dados com o objetivo de classificar as variáveis desses dados de acordo com um parâmetro resposta, podendo estar em classes (classificação) ou um dado numérico (regressão).

Uma das grandes utilidades deste método é a possibilidade de ser usado para criar padrões e regras extraídos de grandes bancos de dados. Esses parâmetros são muito importantes para a discriminação estatística e modelagem preditiva. Essas características, juntamente com suas interpretações intuitivas, são alguns dos motivos deste método ter se tornando amplamente usado em diversos modelamentos em diversas áreas de atuação por mais de duas décadas (Myles *et al*. 2004).

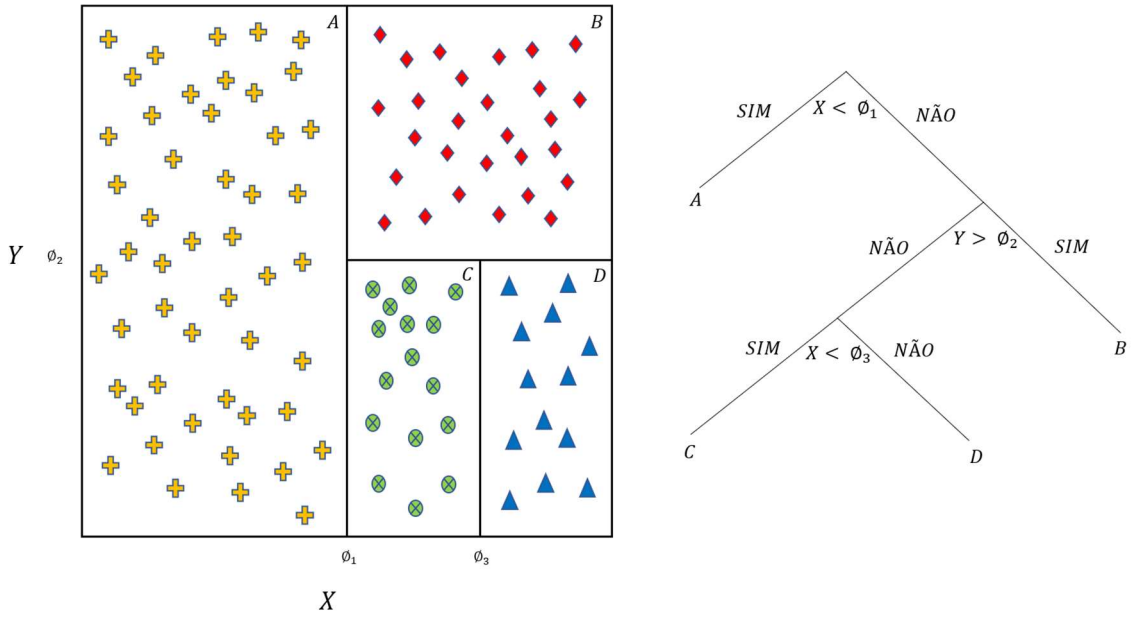
As árvores de decisão são consideradas uma categoria de aprendizagem de máquina e inteligência artificial sendo usado como algoritmo auxiliar de previsão ou como principal

meio preditivo na literatura. Além disso, há diversos estudos realizados em diferentes áreas principalmente nas ciências químicas e bioquímicas.

Uma vantagem que a modelagem de árvore de decisão tem sobre outros técnicas de reconhecimento de padrões reside na interpretabilidade do modelo construído, por apresentar uma visualização gráfica da análise dos parâmetros além da apresentação de uma árvore hierárquica de decisão para o modelo final obtido.

Na Figura 7 é possível analisar como este método funciona em relação a sua disposição gráfica dos dados. Usando como exemplo um banco de dados com duas variáveis preditivas X e Y e uma variável *target* com classes A, B, C e D é possível representar o funcionamento de uma árvore de decisão de classificação. Tendo isso em mente, o algoritmo da árvore consegue criar limites para cada fator e gerar *n* "perguntas" (ϕ_n) para classificar o banco de dados dividindo esses dados em um ponto chamado de "nó" onde uma nova pergunta pode ser feita.

Figura 7 - Distribuição dos dados modelados e funcionamento da árvore de decisões.



Para avaliar a precisão do modelo de árvore de decisão, a parte remanescente do banco de dados é convertido como um conjunto de teste, essas amostras são avaliadas pela regra de decisão no primeiro nó e é passada adiante percorrendo o caminho previsto para o próximo nó, este processo continua até a amostra atingir um nó sem ramificação, este nó é chamado de folha.

Após a classificação de conjunto de teste, os resultados experimentais do modelo são comparados com o rotulo do banco de dados e por meio da comparação é possível calcular os parâmetros de validação do modelo.

Com esse mesmo método de validação do modelo, é possível melhorar a característica preditiva "podando" as ramificações da árvore de acordo com a importância de cada variável analisada, isto reduz o tamanho do modelo. Para se determinar quando a árvore deve ser podada e quais variáveis são mais importantes, Breiman *et al.* (2017) desenvolveram em seu trabalho o Índice de Gini e a Entropia do nó.

2.8.2. ÍNDICE OU PUREZA DE GINI

Tomando um objeto *u* como hipótese, sendo este um conjunto finito não vazio, de acordo com a categoria de valor do atributo, que pode ser dividido em N categorias diferentes, o número do Índice Gini é expresso:

$$Gini(u) = 1 - \sum_{i=1}^{n} [p(c_i|u)]^2$$

O $p(c_i|u)$ expresso no nó *u* está condicionado à probabilidade de que o conjunto de objetos u pertença à classe c_i, quando o valor mínimo de $Gini(u)$ é 0, ou seja, neste nó todos os objetos pertencem à mesma categoria, obtendo-se o máximo de informação útil; quando todos os objetos no nó têm a distribuição uniforme para o campo da categoria, o valor $Gini(u)$ é o máximo, já que as médias podem obter o mínimo de informação útil. Se o conjunto de acordo com um subconjunto de atributos é dividido no número do conjunto *k* (u*, j* *k* = 1, 2 ... *k*), depois de dividir o nó o número $Gini(u)$ é expresso como:

$$Gini_{split}(u) = \sum_{i=1}^{k} \frac{n_j}{n} Gini(u_j)$$

Na equação, *n* é o número de objetos de *u*, n_j é o número do objeto no sub-nó. A ideia básica é que o Índice de Gini para cada atributo percorre todas as possíveis segmentações, até se alcançar o coeficiente Gini mínimo, sendo assim, este nó alcançou a máxima informação útil podendo finalmente finalizar o crescimento da árvore ou prosseguir para um novo sub-nó (Breiman *et al.*, 2017).

Porém, o Índice de Gini precisa de um complemento para determinar se há a necessidade de continuar crescendo a árvore. A Entropia do nó tem como ideia básica medir a

desordem de um agrupamento pela variável *target*. Em vez de utilizar probabilidades simples, este método usa o log_2 das probabilidades para determinar seu valor como pode ser visto na equação a seguir, onde p_j representa a probabilidade de fator *j* da variável *target* estar no nó analisado.

$$Entropia = -\sum_j p_j log_2(p_j)$$

Quanto mais próximo de 0, mais puro o nó vai ser e por consequência, mais próximo de 0 o Índice de Gini vai ser. A entropia é importante para limitar o crescimento da árvore pois acrescenta uma outra validação de impureza para o algoritmo usar como base de poda. Além disso esses parâmetros determinam como o crescimento da árvore se desenvolve de acordo com essa redução da impureza dos nós.

2.8.3. CRESCIMENTO DA ÁRVORE DE CLASSIFICAÇÃO

Breiman *et al.* (1984) desenvolveram dois distintos modelos de árvores de decisões, as de regressão e as de classificação. A principal diferença entre elas é que a primeira trata as informações como números, podendo haver intervalos reais entre as amostras conhecidas, uma variável plausível para este método seria a altura de uma planta, o ângulo do mergulho de camada mineralizada entre outros exemplos. Já a classificação trata de dados em classes ou fatores, ou seja, informações que não aceitam intervalos fora da sua amplitude original.

Como o banco de dados usado neste trabalho está padronizado em classes, o modelo de árvores de classificação é o mais adequado e por isso foi usado para a modelagem destas informações.

Neste método os nós são divididos de acordo com redução da impureza relativa calculada no próprio nó. Esta ideia de encontrar divisões de modo a dar nós "mais puros" a partir de índices como o Índice de Gini que foi implementada por Breiman *et al.* (1984) pode ser feita desta forma:

A. Define-se as proporções do nó $p\ (j|\ t)$, $j = 1....k$, para ser a proporção dos casos $x_n \subset t$ pertencentes à classe *j*, de modo que:

$$p(1|t) + \cdots + p(k|t) = 1$$

B. Define-se uma medida $i(t)$ da impureza de *t* como uma função não negativa ϕ do $p\ (j|\ t)$ de modo que:

$$i(t) = \phi(j_1, j_2, \ldots., j_k) = máximo,$$

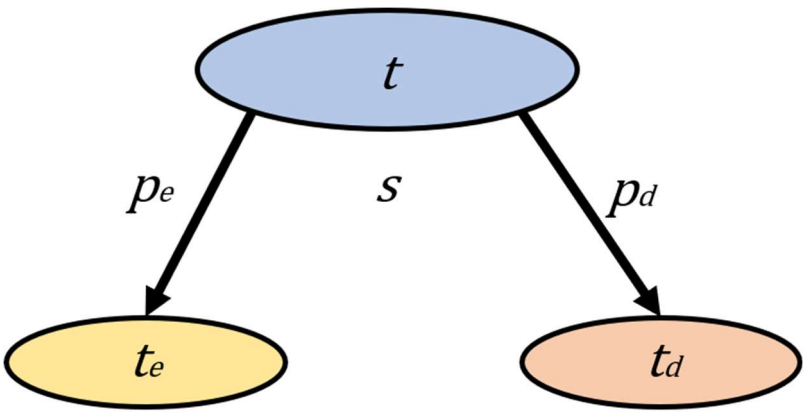
$$\phi(j_1, 0, \ldots.,0) = 0,$$

$$\phi(0,0, \ldots., j_k) = 0$$

Sendo assim, a impureza é máxima quando os dados de um conjunto estão todos misturados em apenas 1 nó, e a partir da separação desses dados a impureza diminui até seu mínimo 0.

Para qualquer nó *t*, há uma divisão candidata *s* do nó que o divide em t_e e t_d de modo que uma proporção p_e dos casos em *t* vai para t_e e uma proporção p_d vai para t_d. como pode ser visto na Figura 8.

Figura 8 - Representação da divisão candidata s do nó t .



Sendo assim, a divisão do nó é definida como a diminuição da sua impureza aumentando a segregação das amostras em cada nova divisão. Sendo assim, essa impureza pode ser demonstrada dessa forma:

$$\Delta i(s,t) = i(t) - p_e i(t_e) - p_d i(t_d)$$

C. Definir o possível candidato do novo conjunto S nas divisões binárias s em cada novo nó de acordo com uma pergunta Q como: Se Q for “Sim” k_n vai para t_e , caso Q for “Não” k_n vai para t_d.

A questão Q é definida a partir de um intervalo do mínimo e máximo local (A,B) sendo que $A \leq B$ e o seu alcance vai de 0 a 1 em passadas de 0,1. O crescimento da árvore continua após J nós candidatos serem criados até que as s^* divisões alcancem a impureza mínima do conjunto.

$$\Delta i(s^*, t_k) = \max_{s \in S} \Delta i(s, t_k)$$

A classe dos nós finais é determinada a partir da regra da pluralidade que determina qual a classe escolhida no nó final para estimar as amostras que passaram por esta divisão. Essa regra está especificada como:

$$\vartheta = p(j_0|t) - \max_j p(j|t)$$

Por fim, Breiman *et al.* (1984) chegaram em 2 possíveis regras gerais para a divisão dos nós a partir das informações apresentadas anteriormente. A primeira é baseada na impureza após os *j* candidatos serem testados:

$$i(t) = -\sum_j p(j|t) \log\left[p(j|t)\right]$$

O segundo método é baseado no Índice de Gini para medir a impureza do nó:

$$i(t) = \sum_{i \neq j} p(j|t) p(j|t)$$

Após a criação da árvore, a última etapa para otimizar o modelo é a poda dos nós da modelagem feita. O método de poda é uma sequência decrescente de subárvores, onde;

$$T_k = T(\alpha_k) * \alpha_1 = 0$$

$$\alpha_k < \alpha_{k+1}, \qquad k \geq 1$$

$$Para \quad k \geq 1, \qquad \alpha_k \leq \alpha \leq \alpha_{k+1}$$

$$T(\alpha) = T(\alpha_k) = T_k$$

O problema agora se reduz a selecionar a árvore de tamanho ideal. Se a estimativa de ressubstituição $R(T_k)$ for usada como critério, a maior árvore T_1 seria selecionada. Mas ao utilizar essa resposta não haveria uma diminuição da árvore final do modelo, portanto a melhor subárvore T_{k0} pode ser definida como o mínimo de $\hat{R}(T_{k0})$.

$$\hat{R}(T_{k0}) = \min_k \hat{R}(T_k)$$

Outro fator muito utilizado em modelamentos é o DLDA, que possui uma função muito parecida com o Índice de Gini. A grande diferença que existe entre estes métodos é que este representa o ganho de acurácia do modelo a partir das variáveis.

2.8.4. DIAGONAL LINEAR DISCRIMINANT ANALYSIS (DLDA)

DLDA é a regra discriminante de máxima verossimilhança, para densidades de classe normais multivariadas, quando as densidades de classe têm a mesma matriz diagonal de variância-covariância. Isso resulta em uma simples regra, onde uma amostra é atribuída à classe *k* que minimiza o resultado de:

$$\sum_{j=1}^{p} \frac{(x_j - \overline{x_{kJ}})^2}{\sigma_j^2}$$

Onde *p* é o número de variáveis, x_j é o valor da variável (gene) *j* da amostra teste, $\overline{x_{kJ}}$ é a média da amostra da classe variável *k* e σ_j^2 é a estimativa (agrupada) da variância do gene j(Díaz-Uriarte,2005). Com isso é possível determinar as variáveis de maior importância na Árvore de decisão e no *Random Forest* de acordo com este somatório, sendo que o resultado deste para cada variável explica o quão determinante este é para se alcançar o mínimo da equação.

2.9. Random Forest

Recentemente, tem havido muito interesse em "métodos *ensemble*", que são métodos que geram muitos classificadores e agregam seus resultados. Um método muito usado é o *bagging* desenvolvido por Breiman (1996a). Preditores de *bagging* geram várias versões de um preditor e os usam para obter um preditor que consiga agregar os dados originais.

Breiman (2001) uniu o método de *bagging* com as árvores de decisão para gerar uma nova metodologia de validação que adicione uma camada adicional de aleatoriedade ao

método. O *Random Forest* é uma combinação de preditores de árvore, gerando uma dependência entre os valores obtidos em cada árvore por uma vetorização aleatória que foi amostrada de forma a não interferir com as demais tentativas de criação de árvores obtidas pelo método. Essa geração de tentativas tende a convergir os seus erros relativos ao limite no decorrer do processo sendo que quanto mais árvores são criadas, mais próximo deste limite o modelo consegue alcançar.

Esses preditores são criados a partir da seleção de variáveis do banco de dados original (*Out-of-Bag* (OOB)) e a partir das estimativas internas também é possível usá-las para medir a importância das variáveis.

Para a validação dos métodos tradicionais de árvore, cada árvore anteriormente gerada recebe um peso para os pontos que foram previstos de forma incorreta, ao final deste processo uma contagem ponderada é entregue para a previsão. Com o *bagging*, as novas árvores não sofrem alterações, pois não dependem das árvores anteriores - cada uma é gerada usando uma amostragem de *bootstrap* do conjunto de dados estudado.

Além disso em árvores de decisão clássicas, cada nó é criado a partir da melhor adequação usando a melhor combinação entre todas as variáveis. No entanto no *Random Forest*, cada nó é dividido usando uma otimização entre um subconjunto de preditores aleatórios naquele nó. Esta técnica consegue gerar um desempenho muito bom em comparação com muitos outros classificadores comumente usados, incluindo análise discriminante, máquinas de vetor de suporte e redes neurais, e é robusto contra *overfitting*, um termo usado em estatística para descrever quando um modelo estatístico se ajusta muito bem ao conjunto de dados anteriormente observado, mas se mostra ineficaz para prever novos resultados (Breiman, 2001a).

Um exemplo inicial é o *Bagging* (Breiman, 1996b). Para gerar o modelo cada árvore deve crescer em uma seleção aleatória de variáveis sem que haja a sua substituição e isto é feito a partir dos exemplos no conjunto de treinamento.

Outro exemplo desenvolvido posteriormente é a *random split selection* (Dietterich, 1998) onde em cada nó criado no modelo final a sua divisão é baseada aleatoriamente na melhor divisão entre todas as outras árvores criadas. Breiman (1999) volta ao seu trabalho e propõe uma nova forma de treinamento ao randomizar as saídas no conjunto de treinamento original podendo adotar outra abordagem ao selecionar o conjunto de treinamento a partir de um conjunto aleatório de pesos baseado no primeiro conjunto de treinamento.

Para entender a base do funcionamento do *Random Forest* será discutido um exemplo criado por Breiman (2001). Dado um conjunto de variáveis preditivas X com uma variável

target Y = h_1**(x),** h_2**(x),. . . ,** h_j**(x),** um conjunto de treino é criado aleatoriamente a partir desses vetores Y e X, define-se a função de contorno, que é um espaço matricial onde as médias das importâncias ($E_{X,Y}$) calculadas pelo Índice de Gini em X não excedam a média para as demais classes. A função de contorno pode ser descrita da seguinte maneira

$$mr(X,Y) = P_{\Theta}(h(X,\Theta) = Y) - \max_{j \neq Y} P_{\Theta}\,(h(X,\Theta) = j)$$

Nesta função, os valores subscritos de X, Y indicam que a probabilidade está acima do espaço X, Y gerado pela função de contorno no modelo de *Random Forests*, h_k *(X) = h (X,* Θk*)*. Para representar essas importâncias é necessário obter o peso da classificação (*s*) E assumindo que $s \geq 0$, por causa da Desigualdade Chebychev (Bienaymé, 1853) a probabilidade das importâncias para mais de duas classes (PE^*) pode ser representada por:

$$s = E_{X,Y} mr(X,Y)$$

$$PE^* \leq \frac{var(mr)}{s^2}$$

Para um grande número de árvores, todas as sequências de probabilidade P_{Θ} convergem para resultando na probabilidade de classificar os dados de X corretamente no vetor Y ($P_{X,Y}$):

$$P_{X,Y}(P_{\Theta}(h(X,\Theta) = Y) - \max_{j \neq Y} P_{\Theta}\,(h(X,\Theta) = j) < 0)$$

Este resultado explica porque os RFs não ajustam os modelos de forma a causar o efeito de *overfitting* pois o acréscimo de árvores faz com que a generalização do erro se estabiliza em um valor limite. Usando essas equações Breiman (2001) chegou em uma função para combinar o efeito de probabilidade das importâncias para mais de duas classes (PE^*). Essa fórmula pode ser escrita da seguinte forma:

$$PE^* = \left(P_{\Theta}(h(X,\Theta) = Y) - \max_{j \neq Y} P_{\Theta}\,(h(X,\Theta) = j) < 0\right)$$

$$\leq \sum_{j} (P_{\Theta}(h(X,\Theta) = Y) - P_{\Theta}(h(X,\Theta) = j) < 0))$$

Com isso definiu-se como calcular a força de cada variável de um conjunto de classificadores $\{h\,(x, \Theta)\}$ em relação à classe *j*. Além disso observou-se que esta definição de força não depende da floresta em si, podendo ser aplicado em outros modelos também. Usando a Desigualdade de Chebyshev (Bienaymé, 1853) para os *j* fatores respostas do conjunto Y (*sj)* e que assumindo todo *sj* > 0 é possível tirar mais informações da fórmula:

$$s_j = P_{X,Y}(P_\Theta(h(X,\Theta) = Y) - P_\Theta(h(X,\Theta) = j) < 0))$$

$$PE^* \leq \sum_j \text{var}(P_\Theta(h(X,\Theta) = Y) - P_\Theta(h(X,\Theta) = j) < 0)){s_j}^2$$

Dessas fórmulas é possível extrair as variâncias e a correlação média das árvores criadas para conjuntos de dados com mais de duas classes. Para dar um auxílio à acurácia do modelo Breiman (2001) aconselha o uso do *Bagging* para a seleção de variáveis. Existem duas razões para usar o *Bagging* de acordo com o autor, a primeira é que o uso deste método tende a aumentar a precisão quando se aumenta a aleatoriedade dos recursos utilizados, além disso é que o *Bagging* pode ser usado para dar estimativas do erro generalizado (*PE**) do conjunto das árvores criadas além de gerar dados em relação a força e correlação das variáveis, podendo ser usado como seletor de parâmetros de acordo com a importância desses para o modelo final.

Para cada Y, X no conjunto de treinamento, os votos ou pesos são agregados apenas sobre aqueles classificadores em T_k que não contém Y, X; essa metodologia se chama de *out-of-bag*. Tibshirani (1996) propôs o uso de estimativas *out-of-bag* como um auxiliar nas estimativas de erro de generalização. Tibshirani (1996) usou estimativas *out-of-bag* de variância para estimar o erro de generalização para classificadores arbitrários. O estudo das estimativas de erro para classificadores OOB em Breiman (1996b), fornecem evidências empíricas para mostrar que a estimativa *out-of-bag* é tão precisa quanto usar um conjunto de teste do mesmo tamanho do conjunto de treinamento.

2.10. Análise da Componente Principal (PCA)

De acordo com Wold & Geladi (1987) a Análise de Componentes Principais (PCA) é uma das grandes bases para a análise multivariada de dados. O PCA fornece uma aproximação dos dados de uma matriz *X*, em termos do produto de duas matrizes menores *T* e *P*. Essas matrizes, *T* e *P*, adquirem os padrões essenciais formadores do conjunto *X* original.

Traçar as colunas de *T* fornece uma representação das componentes ou variáveis dominantes de *X* e, analogamente, traçar as linhas de *P* representam as demais informações complementares da matriz *X*. PCA foi formulado pela primeira vez em estatísticas por Pearson (1901), que formulou a análise como encontrar "linhas e planos de ajuste mais próximo aos sistemas de pontos no espaço".

Para realizar esta análise é necessário que ocorra primeiro a subtração do vetor médio das componentes para garantir que os primeiros componentes principais descrevam a direção de máxima variância. Usando como princípio de uma média empírica nula, a componente principal w_1 de um conjunto *X* é definido como:

$$w_1 = arg \max_{||w||=1} \mathrm{Var}\{\left(w^\top X\right)^2\} = arg \max_{||w||=1} E\{\left(w^\top X\right)^2\}$$

Com os primeiros *k-1* componentes selecionados, a componente *k* pode ser alcançada subtraindo *k-1* das demais componentes de *X*. O resultado disso gera esta condição:

$$\hat{X}_{k-1} = X - \sum_{i=1}^{k-1} w_i w_i^\top X$$

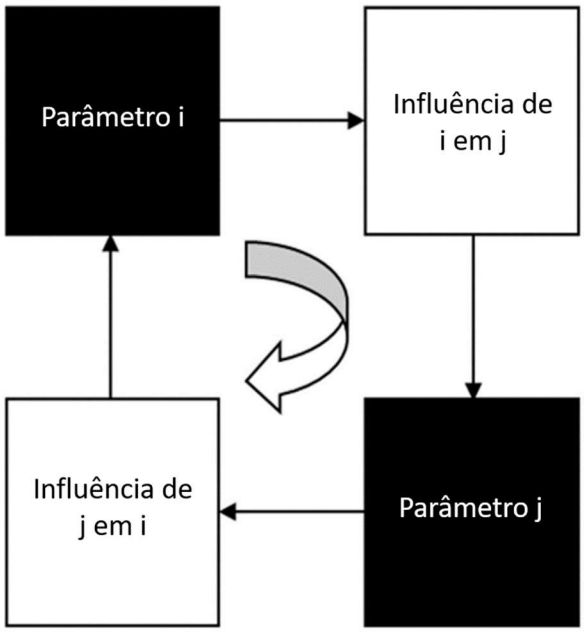
Por fim, juntando estas duas fórmulas é possível determinar o novo conjunto de dados adaptados para essa nova condição. Nesse novo banco de dados a componente principal é obtida por:

$$w_k = arg \max_{||w||=1} E\{\left(w^\top \hat{X}_{k-1}\right)^2\}$$

2.11. MSII - Mine Slope Instability Index

A abordagem de *Rock Engineering Systems* (RES) (Hudson, 1992) é usado para a análise de estabilidade de taludes, para isso as variáveis selecionadas são parametrizadas numa única métrica para melhorar a interpretação do modelo. O RES usa um modelo analítico que consegue alcançar, as condições de contorno como um sistema completo, interativo e dinâmico. Neste sistema é possível estipular os objetivos considerando os parâmetros (variáveis preditivas) relevantes para chegar nessa meta (variável *target*) de acordo com o comportamento do maciço rochoso. O funcionamento desse método pode ser visto na Figura 9.

Figura 9 - Ilustração da matriz de interação em RES para dois fatores.



Fonte - (adaptado de Zare Naghadehi *et al*. 2013).

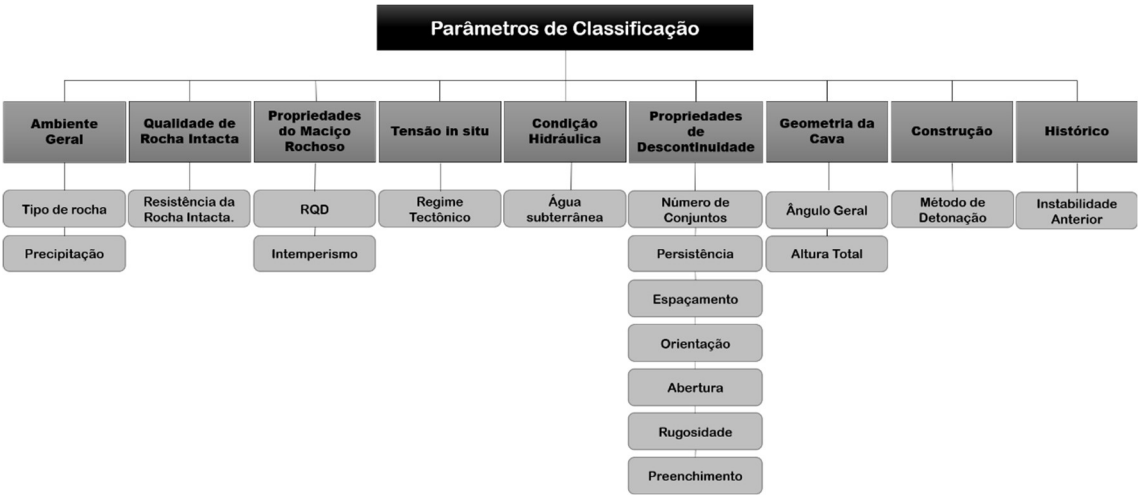
As interações entre os parâmetros na abordagem RES são representadas por meio de uma "matriz de interação" e a partir dessa relação é possível extrair os dados necessários para a obtenção dos seus determinantes.

Então, utilizando este princípio, um índice de risco para estabilidade de taludes em minas a céu aberto pode ser calculado usando os valores obtidos em cada variável associando um valor de peso para cada uma correspondendo a sua importância e distribuição parametrizada; este processo foi desenvolvido por Zare Naghadehi *et al*. (2013) e o resultado foi chamado de *Mine Slope Instability Index* (MSII). O método utilizado pelos autores para atribuir um "peso" a cada parâmetro foi proposto por Hudson (1992).

No sistema desenvolvido por Zare Naghadehi *et al*.(2013), os autores consideraram diversos parâmetros geomecânicos como pode ser visto na Figura 10. A escolha desses parâmetros já

foi discutida anteriormente, porém é importante lembrar que estas variáveis já são muito utilizadas por diversos autores para predição de modelos de estabilidade de maciços rochosos. Por isto é que estas informações são aplicáveis em novos estudos desta área.

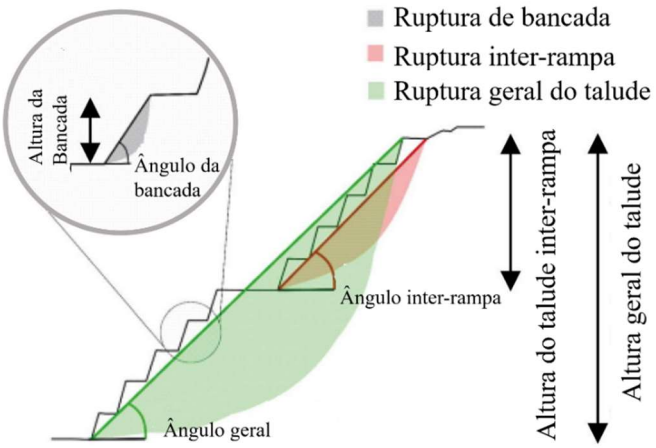
Figura 10 - As categorias selecionadas e os principais parâmetros do sistema.



Fonte - (adaptado de Zare Naghadehi *et al*., 2013).

Como variável *target,* a estabilidade do talude é classificada de acordo com 3 classes distintas: a classe de Talude Estável (ST); a classe de Talude Instável (OF) e a classe de Instabilidade Pontual de Bancada (FSB). Na figura 11 é possível visualizar o efeito das rupturas em diferentes níveis de instabilidade, desde rupturas completas (OF), até rupturas menores em bancadas (FSB).

Figura 11 Esquema de ruptura de taludes em minas a céu aberto.



Fonte - (Adaptado de Santos *et al.* 2019)

2.12. Q-Slope

Desenvolvido por Bar & Barton (2017) o Q-slope é um modelo empírico para engenharia de taludes de rocha com o objetivo de avaliar a sua estabilidade. O método pode ser aplicado em taludes para usos gerais. O Q-slope permite ajustar os ângulos de taludes conforme as condições do maciço rochoso para se adequar de acordo com a estabilidade desejada para cada tipo de operação.

Este método é derivado do Q-system (Barton *et al*., 1974) que é muito utilizado para o planejamento de suportes para tuneis e classificação de estabilidade de maciços rochosos. O Q-slope utiliza os parâmetros RQD, J_n, J_r, J_a, J_w e SRF. No entanto, o par de resistência ao atrito J_r e J_a pode se aplicar, quando necessário, para os lados individuais de cunhas potencialmente instáveis. O termo J_w, que é agora denominado J_{wice} que leva em consideração uma gama mais ampla de fatores ambientais e condições mecânicas para taludes.

- RQD - *Rock Quality Designation* (Deere, 1963; Deere *et al.*, 1967).
- SRF_{slope} - Resultado equivalente de 3 fatores de redução da resistência.
    - SRF_a - Fator da condição física;
    - SRF_b - Fator da tensão e resistência
    - SRF_c - Fator da descontinuidade principal
- J_n - Fator das famílias de descontinuidades.
- J_r - Fator da rugosidade das descontinuidades.
- J_a - Fator da alteração das descontinuidades.
- Jwice - Fator das condições ambientais e geológicas.

A fórmula para o Q-slope pode ser escrita da seguinte forma:

$$Q_{slope} = \frac{RQD}{J_n} \text{ x } (\frac{J_r}{J_a})_0 \text{ x } \frac{J_{wice}}{SRF_{slope}}$$

Da mesma forma que no *Q-system*, a qualidade do maciço rochoso no *Q-slope* pode ser considerado a partir de 3 parâmetros medidos da seguinte forma:

1. Tamanho do bloco: $\frac{RQD}{J_n}$.
2. Força cisalhante: menos favorável $\frac{J_r}{J_a}$ ou pela força cisalhante média para cunhas $(\frac{J_r}{J_a})_1 \mathrm{x}\ (\frac{J_r}{J_a})_2$.
3. Fatores externos $\frac{J_{wice}}{SRF_{slope}}$.

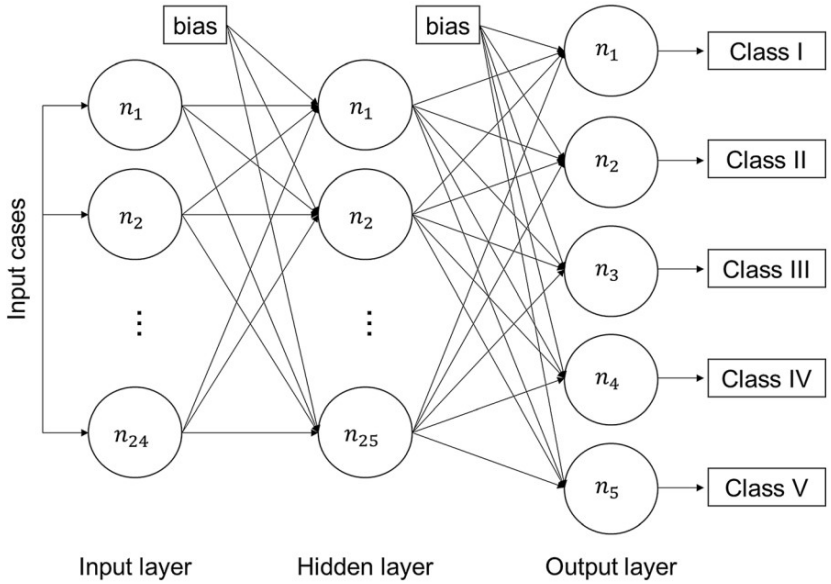
A classificação usada nesse modelo utiliza o *Rock Quality Design*, RQD (Deere, 1963; Deere et al., 1967), número de famílias de descontinuidade (J_n), número da rugosidade das descontinuidades (J_r,), o número de alteração da descontinuidade (J_a), número das condições ambientais e geológicas (J_{wice}) e por fim o fator de redução das forças SRF_{slope}.

2.13. Rock Mass Classification by Multivariate Statistical Techniques and Artificial Intelligence (Santos et al. 2021)

O estudo de Santos *et al*. (2021) é uma proposta de aprimoramento do sistema de classificação RMR de Bieniawski (1989) para maciços rochosos em minas a céu aberto. O banco de dados utilizado nesta pesquisa se baseia em levantamentos realizados em maciços rochosos em minerações a céu aberto brasileiras, com informações dos parâmetros de classificação de maciços rochosos, baseados no RMR. As variáveis relacionadas ao maciço rochoso levam em conta a resistência da rocha intacta, alteração do maciço rochoso, RQD e condição de água no maciço rochoso. Em relação a estruturação geométrica e física das descontinuidades, têm-se espaçamento, persistência, abertura, preenchimento e rugosidade.

As técnicas aplicadas na pesquisa de Santos *et al*. (2021) são provenientes das áreas da estatística multivariada e inteligência artificial. Em relação às estatísticas multivariadas, análise fatorial foi utilizada para identificar fatores subjacentes não observáveis nas variáveis originais, sendo as variáveis compostas por esses fatores utilizadas no sistema de classificação, em vez de todas as variáveis RMR de Bieniawski (1989). Na Figura 12 está representado o sistema de Rede Neural Artificial utilizado por Santos *et al*. (2021). As principais características consideradas para a escolha desse modelo estão relacionadas com aprendizado e adaptação constante, paralelismo massivo, robustez do método, armazenamento associativo de informações e rápido processamento de informação.

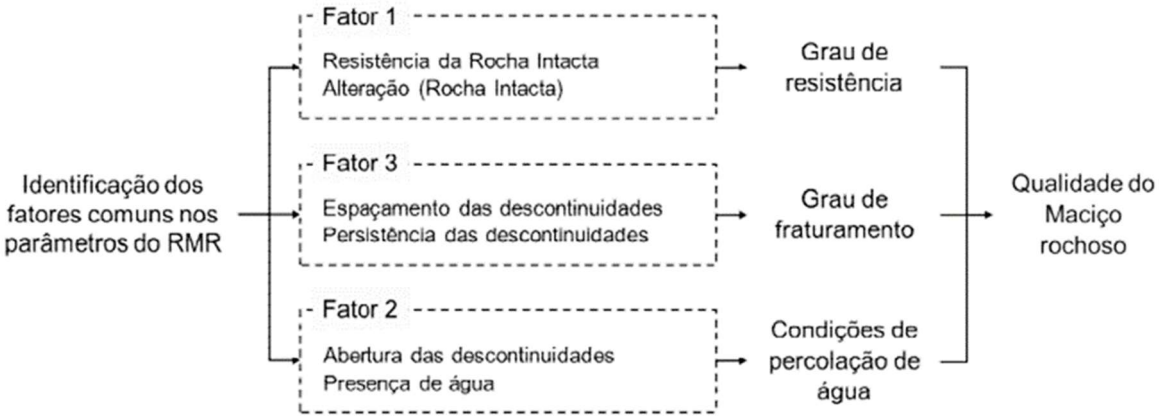
Figura 12 - Esquema da Rede Neural Artificial.



Fonte - (Santos *et al*. 2021).

Santos *et al*. (2021) identificaram 3 fatores pelos resultados da análise fatorial. Fator 1 representando a resistência e alteração do maciço rochoso, o Fator 2 representando as condições do fluxo de água e por fim o Fator 3 representando o grau de fraturamento do maciço rochoso. A seleção de variáveis está representada na Figura 13.

Figura 13 - Representação dos pesos fatoriais rotacionados para cada Fator.



Fonte - (Santos *et al*. 2021).

3. METODOLOGIA

3.1.Materiais

O trabalho foi desenvolvido e implementado no freeware R (R CORE TEAM, 2016), uma linguagem de programação estruturada que tem como objetivo a manipulação, análise e visualização de dados. Este *software* é muito usado dentro das áreas estatísticas e analíticas de dados para o desenvolvimento de modelos estatísticos e análise de dados sendo que os scripts criados para o desenvolvimento desta pesquisa estão no Apêndice 1. A metodologia foi aplicada ao banco de dados compilado e organizado por Zare Naghadehi *et al.* (2013).

O banco de dados organizado por Zare Naghadehi *et al.* (2013) apresenta 84 amostras com 18 variáveis preditivas. Essas amostras foram tiradas de diversas minas espalhas pelo mundo como pode ser visto na Tabela 3.

Tabela 3 - Relação da localização das 84 amostras do banco de dados.

RELAÇÃO DAS AMOSTRAS DOS TALUDES DO BANCO DE DADOS DE ZARE NAGHADEHI *ET AL*. (2013)					
Número de dados	**País**	**Mina**	**Número de dados**	**País**	**Mina**
4	Irã	Angooran	5	Australia	Cadia-Hill
5	Irã	Chadormalou	6	Suécia	Aitik
5	Irã	Choghart	7	Chile	Escondida
4	Irã	Golegohar	5	Espanha	Aznalcollar
4	Irã	Sarcheshmeh	5	EUA	Betze-Post, Goldstrike
4	Irã	Sungun	2	Espanha	La Yesa
7	África do Sul	Venetia	1	Chile	Ujina, Collahuasi
5	Brasil	Águas Claras	1	Canadá	Panda, Ekati
5	Chile	Chuquicamata	1	EUA	Esperanza, Phelps-Dosge
6	África do Sul	Sandsloot	2	Papua Nova-Guiné	Ok-Tedi

Além disso, os autores parametrizaram essas informações em intervalos de 0 até 1, como pode ser visto nas Tabelas 4 e 5 com o objetivo de sistematizar numa mesma escala essas informações e melhorar a interpretação desses dados. A variável *target* tem 3 níveis de resposta: ST para classificar taludes estáveis; FSB para taludes com instabilidades pontuais de bancada e OF para instabilidades totais de talude.

Tabela 4 - Relação dos pesos atribuídos para cada variável de acordo com sua categoria A.

Parâmetro	**Classificação categórica e seus pesos**					
Peso	**0**	**0.2**	**0.4**	**0.6**	**0.8**	**1**
Tipo de Rocha (litologia)	Ígnea: Granito, granodiorito, diorito e gabro; Metamórfica: gnaisse, quartzito e anfibolito.	Sedimentar: Brecha, grauvaca, arenito e conglomerado; Metamórfica: Hornfels; Ígnea: Obsidiana, andesito, norito e aglomerado.	Sedimentar: Anidrito e Gipsito; Ígnea: Tufo basalto, brecha, dacito e riolito.	Sedimentar: Calcário, folhelho, dolomito, gesso, siltito; Metamórfica : ardósia, filito e mármore.	Metamórfica : Xisto e milonito.	Sedimentar : Xisto argiloso, Lamito, Argilito.
Resistencia à compressão simples da rocha intacta (MPa)	> 150	150 - 100	100 - 75	75 - 50	50 - 25	< 25
Peso	**0**	**0.3**	**0.6**	**0.8**	**1**	
RQD (%)	100 - 75	75 - 50	50 - 25	05 - 10.	< 10	
Alteração	Sem alteração (rocha fresca)	Suavemente alterada	Moderadamente alterada	Intensamente alterada	Completamente alterada	
Regime Tectônico	Fraco (ausência de eventos tectônicos)	Moderado (presença de foliações, xistosidade e clivagens)	Forte (presença de dobras, falhas e descontinuidades)	Muito Forte (zonas altamente fraturadas)	Intenso (imbricações e falha de calvagamento)	
Condições de Água Subterrânea	Completamente seco	Úmido	Molhado	Gotejando	Corrente	
Número de Famílias	0	1	2	3	> 3	

Tabela 5 - Relação dos pesos atribuídos para cada variável de acordo com sua categoria B.

	Parâmetro	Classificação categórica e seus pesos				
	Peso	**0**	**0.3**	**0.6**	**0.8**	**1**
Propriedades das descontinuidades	**Persistência (m)**	< 5	05 - 10.	05 - 25.	25 - 40	> 40
	Espaçamento (hb = altura do banco)	> 3hb	2hb - 3hb	1hb - 2hb	1/5hb - 1hb	< 1/5hb
	Orientação *	Muito favorável βd > βs e αd - αs > 30°	Favorável βd > βs e αd - αs < 30°	Regular ≤ βd < βs/4 ou αd - αs > 30°	Desfavorável βs/4 ≤ βd < βs/2 e αd - αs < 30°	Muito desfavorável βs/2 ≤ βd < βs e αd - αs < 30°
	Abertura (mm)	Sem abertura	< 0.1	0.1 - 1	01 - 05.	> 5
	Rugosidade	Muito rugoso	Rugoso	Levemente Rugoso	Macio	Escorregadio
	Preenchimento	Inexistente	Muito duro	Duro	Macio	Muito macio
Talude	**Ângulo**	< 30°	30 - 40°	41 - 50°	51 - 60°	> 60°
	Altura (m)	< 50	50 - 100	100 - 200	200 - 300	> 300
	Método de Desmonte	Pré-corte	Pós-corte	Desmonte controlado, paredes lisas	Desmonte modificado	Desmonte regular/mecânico
	Instabilidade passada	Inexistente	Inativo	Em repouso	Relativamente ativo	Altamente ativo
	Precipitação (mm/ano)	< 150	150 - 300	300 - 450	450 - 600	> 600

* αs = direção de mergulho do talude; αd = direção de mergulho da descontinuidade; βd é o mergulho da descontinuidade; βs é o mergulho do talude

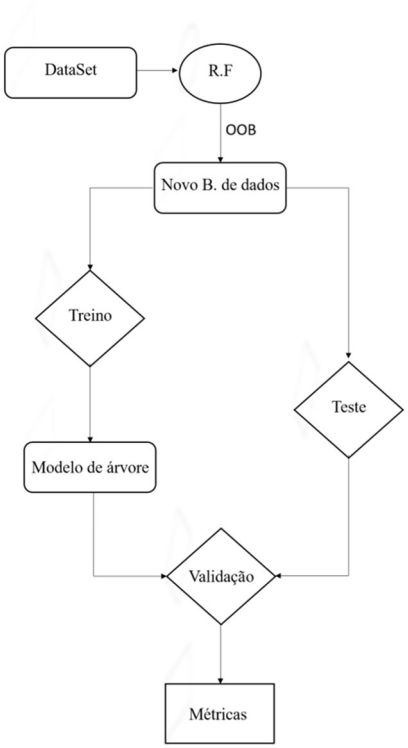
Com estes dados, foram desenvolvidos 6 modelos de árvores de decisão de classificação, que foram nomeadas de Modelo Geral (MG), Modelo Matemático (MM), Modelo Q-slope (MQ), Modelo Santos (MS), Modelos Matemático Sem erros (MMS) e por fim Modelo Santos Sem erros (MSS). Cada modelo possuí suas características que serão descritas na sequência do texto.

3.2.Métodos

Para o desenvolvimento desta pesquisa foram utilizados os métodos de *Random Forest* (RF), Árvore de Decisões para classificação, Análise das Componentes Principais (PCA) e *Bootstrap*. Cada um destes possuí sua função especifica, o RF foi usado para a seleção de variáveis para serem usadas nos dois modelos matemáticos (MM e MMS); as Árvores de Decisões foram utilizadas para o desenvolvimento dos 6 modelos de predição (MG; MM; MQ; MS; MMS e MSS) que foram comparados de acordo com os métodos de validação escolhidos; O PCA foi utilizado para identificar e classificar as amostras de acordo com os seus erros de estimativa usadas para o treino dos modelos.

Para os modelos de *Random Forest*, a estrutura de *Bagging* foi usada de acordo com Breiman (2001). Na Figura 14 pode ser observado o fluxograma de funcionamento desse método e como cada OOB é usado no método.

Figura 14 - Esquema de funcionamento do método de Bagging para árvores de decisões.



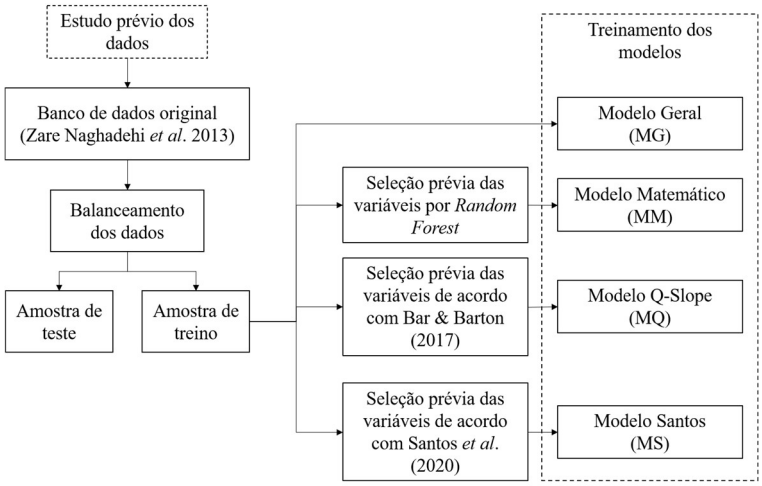
O *Bagging* está presente dentro do algoritmo usado no *Random Forest*, ou seja, ao aplicar o método automaticamente o OOB é feito. Já a criação dos modelos de Árvores de Decisão se baseia nos métodos de Breiman, (2001), cada modelo foi criado com uma adaptação do banco de dados de Zare Naghadehi *et al*. (2013) de acordo com a seleção de variáveis feita para cada modelo.

3.3. Metodologia Geral

Cada modelo foi obtido a partir de um conjunto diferente de parâmetros baseado nas especificações de cada grupo. O primeiro deles é o Modelo Geral (MGMG Modelo Geral
MM Modelo Matemático
MG Modelo Geral
MM Modelo Matemático
MG Modelo Geral
MM Modelo Matemático
MG Modelo Geral
MM Modelo Matemático
MG Modelo Geral
MM Modelo Matemático
MG Modelo Geral
MM Modelo Matemático
MG Modelo Geral
MM Modelo Matemático
), que tem os 18 parâmetros geoestatísticos do banco de dados balanceados; o segundo é o Modelo Matemático (MM), que tem as variáveis selecionadas pelo método de *Random Forest* que determina quais parâmetros possuem uma maior importância para a determinação da resposta da variável *target*.

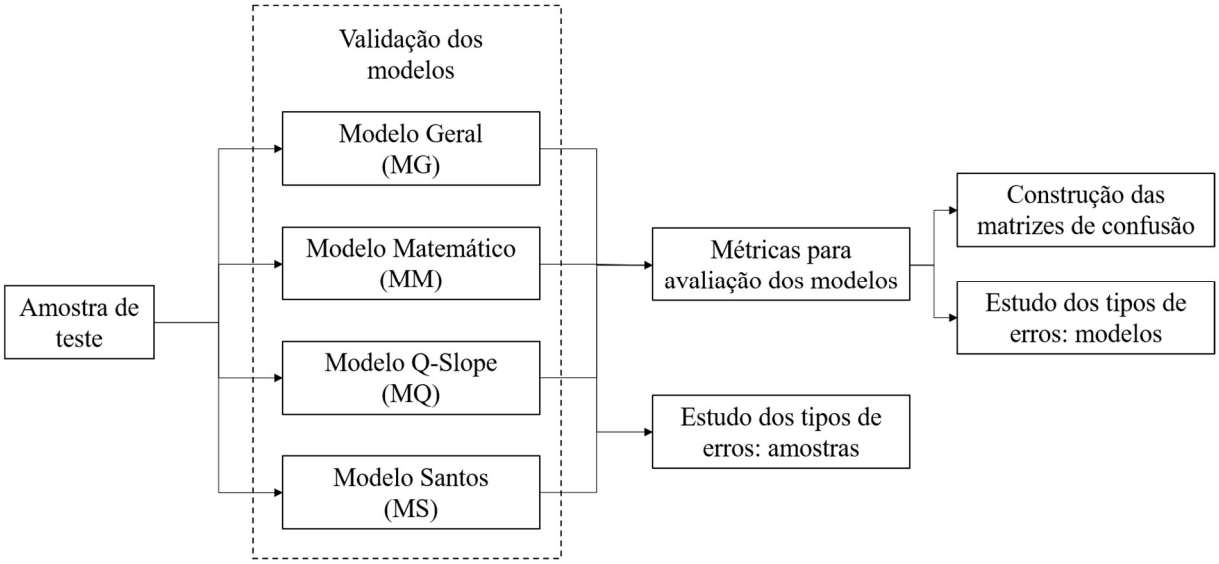
O terceiro modelo é baseado nas variáveis do modelo Q-Slope (MQ) proposto por Bar e Barton (2017). Essas variáveis serão destrinchadas para se alcançar seus equivalentes referentes ao banco de dados usado. O quarto modelo é feito a partir dos resultados obtidos por Santos *et al.* (2020). As etapas feitas para o desenvolvimento desses modelos podem ser vistas na Figura 15.

Figura 15 - Desenvolvimento dos 4 primeiros modelos de árvores de decisão.



Após o treino desses 4 modelos, estes foram validados usando as amostras de teste e como resultado foram obtidas as métricas para o estudo dos erros das amostras e dos modelos. Esta etapa pode ser vista na Figura 16.

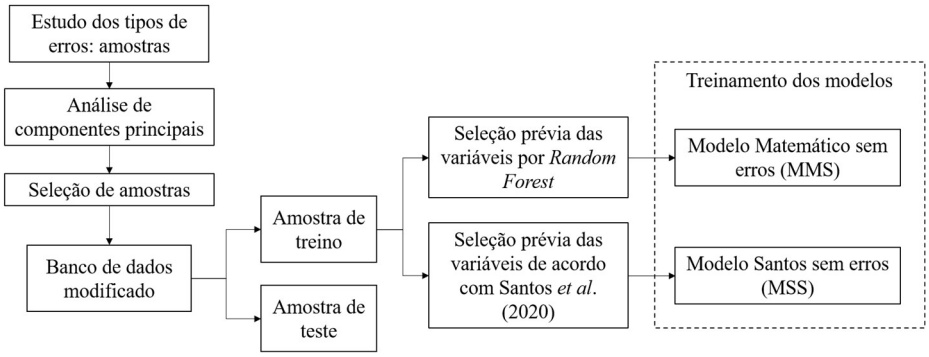
Figura 16 - Esquema de validação dos 4 modelos e estudo dos erros.



Por fim, os dois últimos modelos foram criados a partir dos resultados obtidos com o PCA do estudo dos erros das amostras. Com a identificação dos erros dos 4 primeiros modelos, foi percebida a presença de amostras estimadas erradas que se repetiam em mais de

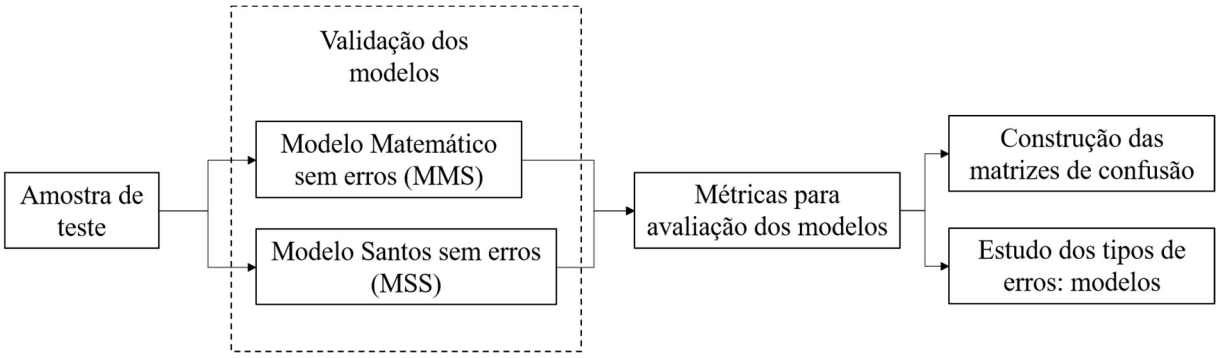
um modelo. Com isso, um novo banco de dados foi criado com a remoção dessas amostras e inspirados nos modelos MS e MM, o Modelo Matemático Sem erros (MMS) e o Modelo Santos Sem erros (MSS) foram criados usando os mesmos princípios usados para o desenvolvimento dos seus antecessores. O esquema desenvolvido para a criação desses dois novos modelos pode ser visto na Figura 17.

Figura 17 - Esquema do desenvolvimento dos modelos sem erros a partir da análise dos erros do PCA.



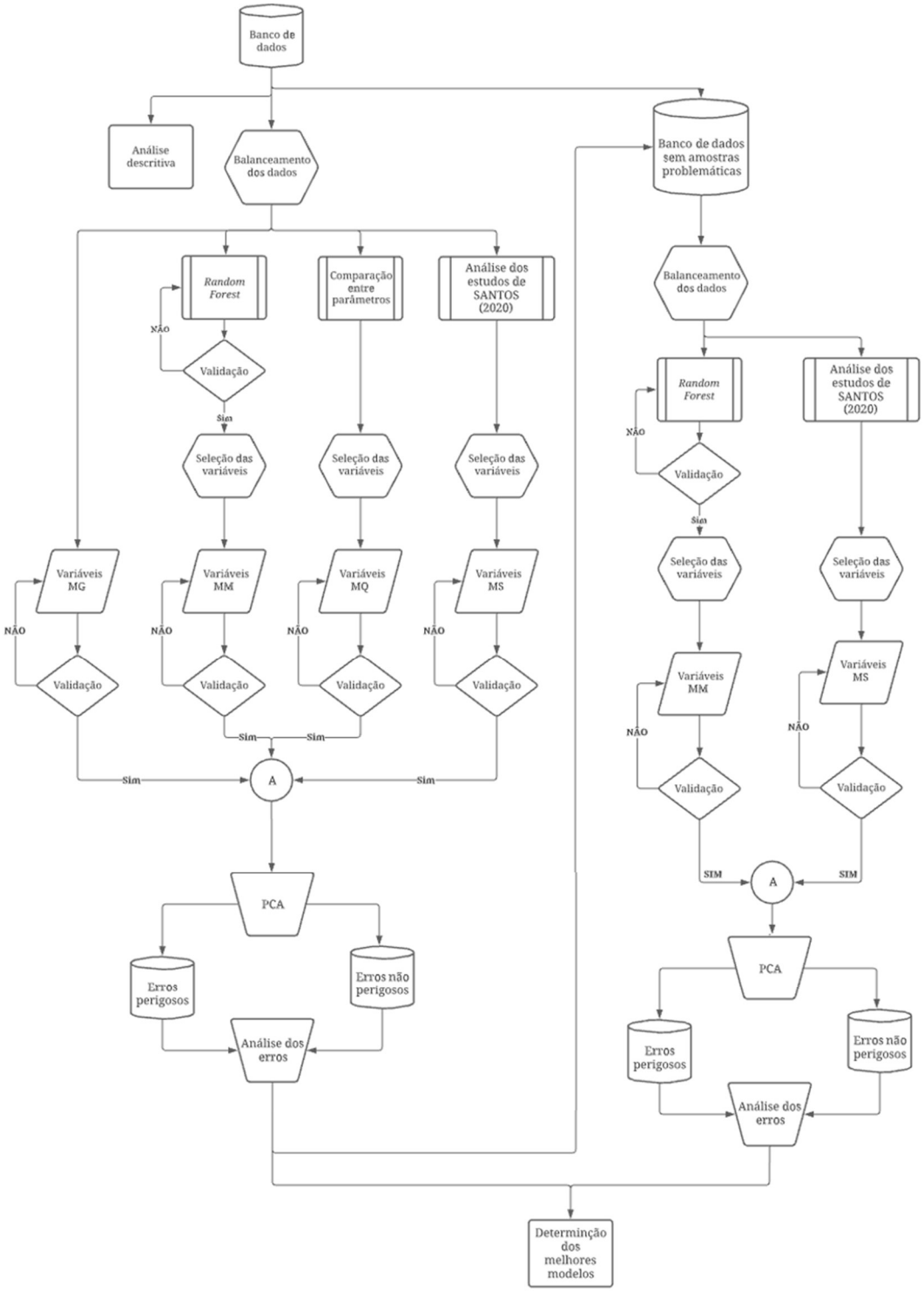
Por fim, para se alcançar as métricas de avaliação dos modelos, os dois novos modelos foram submetidos à novas amostras de teste para sua validação resultando em parâmetros comparativos. A etapa de validação destes últimos modelos está demonstrada na Figura 18.

Figura 18 - Validação e obtenção das métricas dos modelos sem amostras problemáticas.



Utilizando as matrizes de confusão de cada modelo foi possível determinar quais destes obtiveram uma melhor resposta e consequentemente qual deles será escolhido como a resposta para este trabalho. O fluxograma completo deste trabalho está representado na Figura 19.

Figura 19 - Fluxograma metodológico do trabalho.



Porém antes de começar a modelagem é importante garantir alguns requisitos para que o resultado final deste processo seja otimizado. O primeiro critério que deve ser preenchido é o balanceamento dos dados usados na criação dos modelos.

3.4. Considerações especificas da metodologia

3.4.1. BALANCEAMENTO DOS DADOS

Zare Naghadehi *et al*. (2013) em seu trabalho classificaram os taludes estudados em três categorias distintas: a primeira são os taludes estáveis (ST), o segundo são os taludes instáveis (OF) e por fim temos os taludes com falhas de bancada pontuais (FSB).

De acordo com Vladislavleva *et al.* (2010) a necessidade de balanceamento de dados para regressão se origina da pesquisa aplicada. O objetivo de qualquer abordagem de modelagem empírica é inferir dependências ocultas de dados fornecidos. Para um problema de classificação a tarefa é usar os dados fornecidos para representar a saída de variáveis como funções analíticas de algumas ou todas as entradas. Se a abordagem de modelagem não consegue encontrar uma convincente relação entre as entradas e as saídas, pode ser que os dados não contenham as informações necessárias para prever essas saídas.

Uma forma de garantir que haja dados suficientes entre todos os fatores de saída é garantir, além dum número grande de dados, que estes tenham o mesmo número de amostras para serem analisadas.

Para realizar este processo foi utilizado o pacote *ROSE* (Lunardon *et al.*2014) no software R (R CORE TEAM, 2016). O *Random Over-Sampling Examples* (ROSE) fornece funções para lidar com a classificação binária de problemas na presença de classes desequilibradas. Amostras sintéticas balanceadas são gerados de acordo com o *ROSE* (Menardi e Torelli, 2013).

Funções que implementam soluções mais tradicionais para o desequilíbrio de classe também são fornecidas, bem como diferentes métricas para avaliar a precisão do modelo, que são estimadas por métodos de validação*, bootstrap* ou validação cruzada.

Para o método é considerado um domínio χ incluído em R^d , isto é, P(x) = f (x) é uma densidade de probabilidade função em χ. Sem perda de generalidade, podemos considerar que $n_j < n$ é o tamanho de Υ_j , j = 0,1. O procedimento ROSE para gerar um novo exemplo artificial consiste nos seguintes passos:

I. Seleciona-se $y = \Upsilon j \in \Upsilon$ com probabilidade de 1/2

II. Seleciona-se (x_i, y_i) em T_n de modo que $y_i = y$ com probabilidade $pi = 1/n_j$

III. Amostra-se x de $K_{Hj}(\cdot, x_i)$, com K_{Hj} uma distribuição de probabilidade centrada em xi e H_j uma matriz de parâmetros de escala.

Extrai-se do conjunto de treinamento uma observação pertencente a uma das duas classes (escolhido dando a mesma probabilidade para Υ_0 e Υ_1) gerando um novo exemplo em sua vizinhança, onde a largura da vizinhança é determinada por H_j. Normalmente, K_{Hj} é escolhido no conjunto das distribuições simétricas unimodais (Menardi e Torelli, 2013). Vale a pena notar que, uma vez que uma classe de rótulo foi selecionada:

$$\hat{f}(x|y = \Upsilon j) = \sum_{i=1}^{n_j} p_i Pr\,(x|x_i)$$

$$= \sum_{i=1}^{n_j} \frac{1}{n_j} Pr\,(x|x_i)$$

$$= \sum_{i=1}^{n_j} \frac{1}{n_j} K_{H_j}\,(x - x_i)$$

Conclui-se que a geração de novos exemplos da classe Υ_j, segundo ROSE, corresponde à geração de dados a partir da estimativa da densidade do kernel de $\hat{f}(x|y = \Upsilon j)$. A implementação repetida das etapas, um novo conjunto de treinamento balanceado é criado T_m^*, com tamanho m tendo aproximadamente o mesmo número de exemplos pertencentes às duas classes (Menardi e Torelli, 2013). Após esta etapa foi possível dar início ao desenvolvimento dos 6 modelos propostos.

3.4.2. DESENVOLVIMENTO DO RANDOM FOREST

O *Random Forest* (RF) é comumente usado para a criação de estimadores mais robustos baseados nos princípios das árvores de decisões, no entanto como a proposta deste trabalho é criar um modelo de fácil aplicação em campo o RF teve outro propósito, determinar as variáveis do banco de dados mais importantes que serão usadas para gerar o modelo de árvore de decisão do modelo MM e MMS.

Esta técnica só foi utilizada nestes modelos específicos (MM e MMS), pois os demais já possuem suas variáveis determinadas de acordo com a literatura (MS e MSS), com sua interpretação equivalente dos parâmetros (MQ) ou utilizando todas as variáveis preditivas originais (MG Modelo Geral

MM Modelo Matemático

Para esta tarefa foi utilizado o pacote *varSelRF* (Dias-Uriarte, 2005), o *Variable Selection using Random Forests* tem como objetivo selecionar quais as variáveis são mais importantes para a determinação do modelo final por RF.

Para utilizar o *Random Forest* foi criada uma regra de partição do banco de dados para criar dois sub-ramos dos dados originais, um conjunto para o treino do modelo com 80% dos dados e outro para o teste com 20% dos dados restantes de verificação deste, é importante salientar que o algoritmo garante que estes dois subconjuntos possuam total compatibilidade estatística em relação ao banco de dados original.

Os parâmetros usados para gerar o RF foram o número de árvores que deverão ser criadas além do número de interações entres elas, além é claro dos dados de treino selecionados pela partição das amostras. Para selecionar as variáveis do modelo o pacote *varSelRF* utiliza o método DLDA para as importâncias além do Índice de Gini.

3.5.Desenvolvimento das árvores de decisões

Após selecionar os parâmetros para os 4 modelos propostos (Tabela 6) foi possível gerar as árvores de decisões baseado nessas informações. Para o modelamento dessas árvores foram usados dois pacotes principais, o primeiro foi o *rpart* (Therneau & Atkinson, 2019) e o segundo foi o *partykit* (Hothorn *et al*. 2021).

Tabela 6 - Relação das variáveis selecionadas para cada modelo.

Modelo Geral (MG)	**Modelo Matemático (MM)**	**Modelo Q-slope (MQ)**	**Modelo SANTOS *et.al* (2020) (MS)**
Tipo de Rocha (V1)	Resistência da rocha intacta (V2)	Resistência da rocha intacta (V2)	Resistência da rocha intacta (V2)
Resistência da rocha intacta (V2)	RQD (V3)	RQD (V3)	Alteração (V4)
RQD (V3)	Alteração (V4)	Alteração (V4)	Água subterrânea (V6)
Alteração (V4)	Número de famílias (V7)	Número de famílias (V7)	Persistência (V8)
Regime tectônico (V5)	Orientação (V10)	Espaçamento (V9)	Espaçamento (V9)
Água subterrânea (V6)	Ângulo geral (V14)	Orientação (V10)	Abertura (V11)
Número de famílias (V7)	Instabilidade prévia (V18)	Abertura (V11)	
Persistência (V8)		JRC (V12)	
Espaçamento (V9)		Preenchimento (V13)	
Orientação (V10)		Método de desmonte (V16)	
Abertura (V11)		Precipitação anual (V17)	
JRC (V12)			
Preenchimento (V13)			
Ângulo geral (V14)			
Altura geral (V15)			
Método de desmonte (V16)			
Precipitação anual (V17)			
Instabilidade prévia (V18)			

O *Recursive Partitioning and Regression Trees* (*rpart)* é um pacote voltado para o particionamento recursivo para classificação, árvores de regressão e classificação utilizando os conceitos implementados do livro de 1984 de Breiman *et al.*.O *Toolkit for Recursive Partytioning* (*partykit)* é um conjunto de ferramentas com funções para representar, resumir e visualizar modelos de regressão e classificação estruturados em árvore.

O conjunto de treino usado em todos os modelos seguiu a mesma regra de partição usada no RF, 80% dos dados foram usados no treino e os demais 20% foram usados para o teste do modelo. Para a análise do PCA, um novo conjunto de treino foi feito a partir dos dados originais para se obter a relação dos erros de cada modelo.

Sabendo disso é possível desenvolver os 6 modelos de árvore de decisão. Porém é necessário que haja algum método de validação dos resultados finais. Os próprios pacotes utilizados para o desenvolvimento das árvores (*rpart* e *varSelRF*) já fazem este trabalho utilizando validações cruzadas como o K-*Fold* e o *bootstrap,* porém além destes um outro método é utilizado, a Análise das Componentes Principais (PCA).

3.6.Análise das Componentes Principais

Para realizar esta análise, foram utilizados dois pacotes. O primeiro é o *factoextra* (Kassambara, 2020) e o segundo o *FactoMineR* (Sebastien *et al.* 2008). Estes pacotes tem como objetivo fornecer algumas funções fáceis de usar para extrair e visualizar a saída de análises de dados multivariados. Para esta análise foram utilizadas as 84 amostras originais do banco de dados como amostras de teste para os modelos.

Após a realização destes passos, foi possível alcançar os resultados das análises e modelagens feitas no banco de dados para classificar a natureza de cada erro, essa classificação foi feita baseada em uma regra muito simples, dividindo os erros em duas classes distintas, os Erros Perigosos e os Erros Não Perigosos.

O Erros Perigosos são aquelas amostras que foram estimadas por um dos modelos com uma estabilidade superior a real, ou seja, são amostras que tiveram suas estabilidades superestimadas. Já os Erros não Perigosos são aquelas amostras que foram classificadas com uma estabilidade inferior a real, sendo assim esta amostra teve sua estabilidade subestimada pelo modelo analisado. As estimativas reais de estabilidade estão apresentadas no Anexo 1.

Essa classificação é importante de ser feita pois, em análises de estabilidade de talude, é importante reduzir principalmente os Erros Perigosos, já que estes são muito mais problemáticos para modelos dessa natureza já que estes podem causar acidentes graves se um modelo estimar uma amostra com uma estabilidade superior a real.

Com a relação dos erros foi possível determinar quais amostras foram estimadas erradas em mais de um modelo. Como cada modelo utiliza variáveis diferentes é incomum a presença de muitas amostras sendo estimadas erradas em mais de um modelo. Sendo assim para averiguar um possível caráter problemático dessas amostras dois novos modelos foram criados usando como base os dois melhores modelos obtidos originalmente, porém usando um banco de dados alterado com a remoção dessas possíveis amostras problemáticas (veja Tópicos 4.7 e 4.8). Caso esses novos modelos fossem melhores que os 4 modelos originais (MG; MM; MQ e MS), seria comprovada a incoerência dessas amostras removidas.

Para a criação desses dois modelos novos, os mesmos scripts usados para o desenvolvimento dos seus antecessores foram usados para manter as características básicas de cada um para assim, garantir que as mudanças na sua modelagem sejam justificadas apenas pela mudança nos dados usados para o desenvolvimento destes modelos.

3.7. Validação dos modelos

Para determinar a acurácia, além de outros parâmetros, dos modelos as seguintes formulas podem ser usadas. Todas as equações utilizam os dados obtidos a partir dos resultados obtidos na matriz de confusão (Tabela 7).

Tabela 7 - Matriz de confusão

	Classe predita	
Classe verdadeira	Positiva	Negativa
Positiva	V_p	F_n
Negativa	F_p	V_n

$$Especificidade = \frac{V_n}{V_n + F_p}$$

$$Acurácia = \frac{V_n + V_p}{V_n + F_p + V_p + F_n}$$

$$Sensibiildade = \frac{V_p}{V_p + F_n} \quad Especificidade = \frac{V_n}{V_n + F_p}$$

Com a sensibilidade e a especificidade calculada para o modelo, é possível calcular a eficiência do modelo a partir da seguinte formula. Com ela será possível finalmente determinar qual é a capacidade preditiva do modelo desenvolvido neste trabalho junto com a acurácia.

$$Eficiência = \frac{(Sensibilidade + Especificidade)}{2}$$

No caso das árvores de decisão criadas, o conjunto de dados de teste foi usado para validar o modelo e gerar os dados estatísticos pela matriz de confusão. Porém, para a análise do PCA o conjunto total de dados não balanceados foi usado como conjunto de teste para classificar as amostras estimadas erradas como Erros Perigosos ou Não Perigosos. A utilização desse conjunto de teste maior com os dados originais foi necessária para se alcançar uma melhor estimativa da eficiência e acurácia dos modelos.

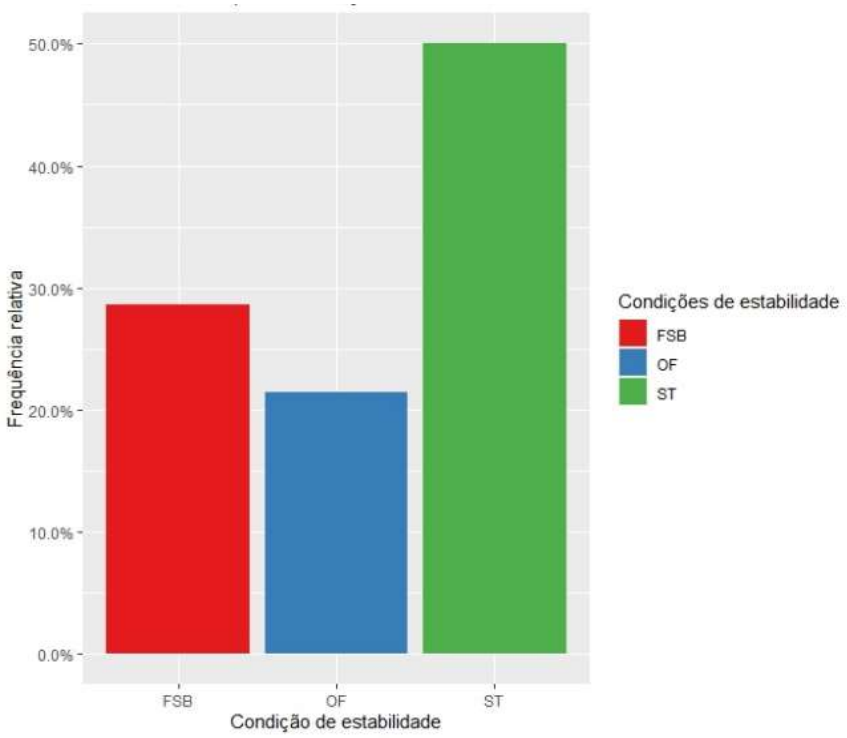
No entanto, para os modelos que usaram RF, uma outra validação foi feita, utilizando o *Bootstrap* um novo conjunto de amostras foi criado para ser testado no modelo que selecionou as variáveis para o Modelo Matemático. Esse método foi feito a partir de uma função que existe dentro do pacote *varSelRF* (Dias-Uriarte, 2005), que apresenta estes resultados como uma probabilidade de cada amostra ser classificada de acordo com os fatores presentes na variável *target* utilizada na criação do modelo RF. Neste caso, o resultado desse *Bootstrap* é um gráfico com essas probabilidades para as classes ST, FSB e OF.

4. RESULTADOS E DISCUSSÕES

4.1. Análise estatística dos dados

Inicialmente foi feita a análise descritiva das informações do banco de dados. A primeira análise foi feita em relação ao balanceamento de classes no banco de dados. A Figura 20 apresenta a distribuição original dos dados em relação às classes. A partir da Figura 20 é possível visualizar a predominância da classe estável (ST) em relação às classes OF e FSB.

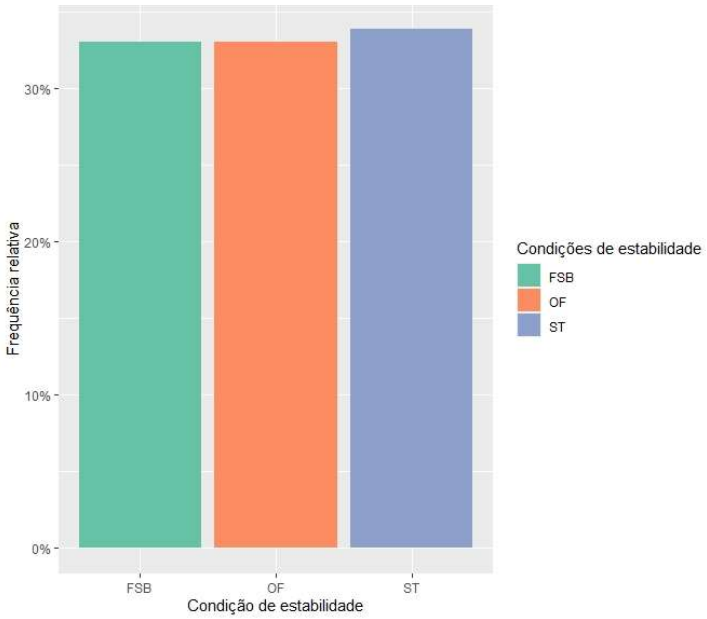
Figura 20 - Gráfico de barras para os fatores de estabilidade não balanceados.



Fonte - R CORE TEAM ,(2016).

A Figura 21 apresenta a nova distribuição dos dados para as três classes. Sendo assim, os dados nas classes minoritárias foram completados para se alcançar o mesmo nível de informações da classe ST, o banco de dados passou de 84 amostras para 124.

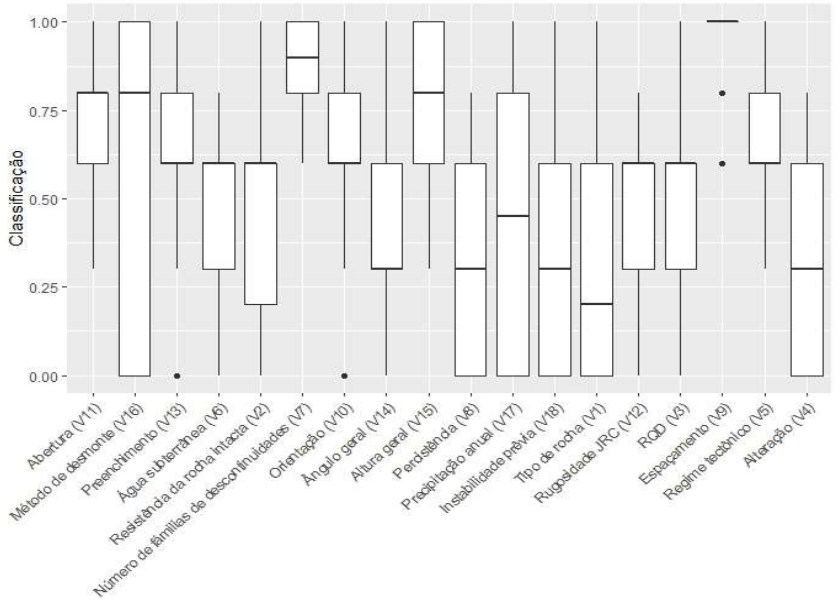
Figura 21 - Gráfico de barras para os fatores de estabilidade não balanceados.



Fonte - R CORE TEAM ,(2016).

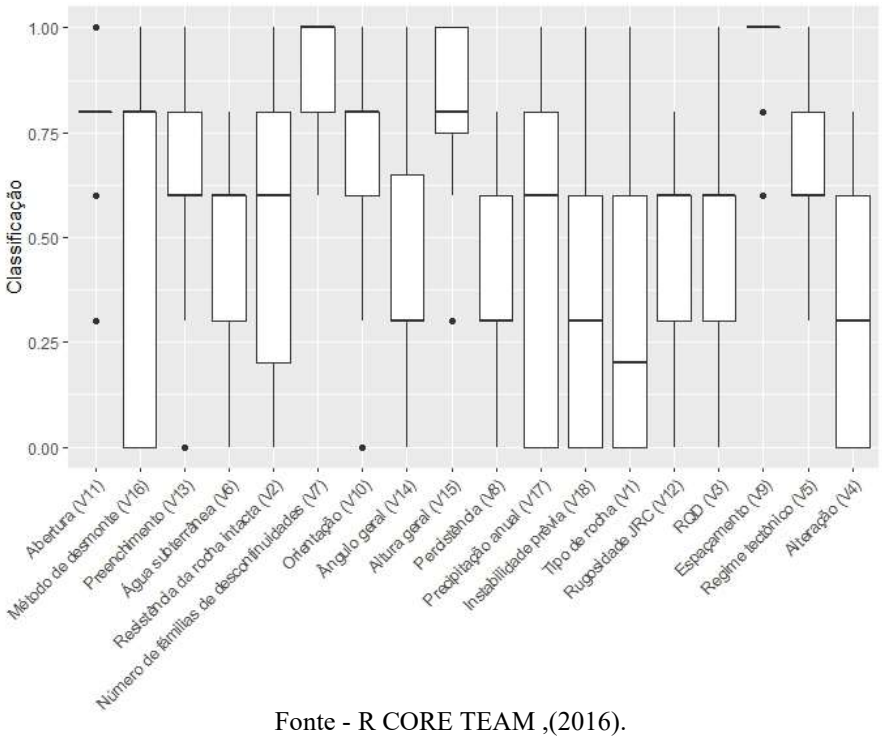
Além disso, foram feitos os gráficos de *boxplot* para cada variável para demonstrar a eficácia do modelo *ROSE* em preservar as características de distribuição do banco de dados original como pode ser visto nas Figuras 22 e 23.

Figura 22 - *Boxplot* dos dados não balanceados com todas as variáveis.



Fonte - R CORE TEAM ,(2016).

Figura 23 - *Boxplot* dos dados balanceados com todas as variáveis.

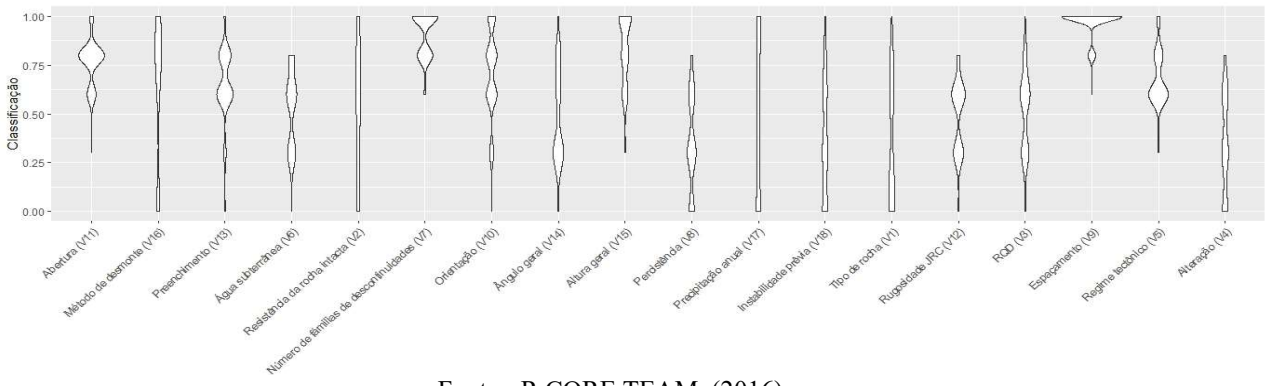


Fonte - R CORE TEAM ,(2016).

Mesmo preservando as características originais do banco de dados, houve uma alteração na distribuição de algumas variáveis, isso ocorreu devido à proporcionalidade das classificações do próprio banco de dados. Isso é esperado do processo de balanceamento, porém o balanceamento não cria dados, pelo contrário reamostra. Portanto, isso não afeta o modelo preditivo.

Variáveis como a abertura (V11) e espaçamento (V9) possuem originalmente uma grande concentração de informações localizadas em um único ponto, essa situação pode ser vista também na Figura 24. Porém, mesmo com essas pequenas alterações entre os dois bancos de dados, os dados balanceados ainda são uma ótima representação expandida das informações originais.

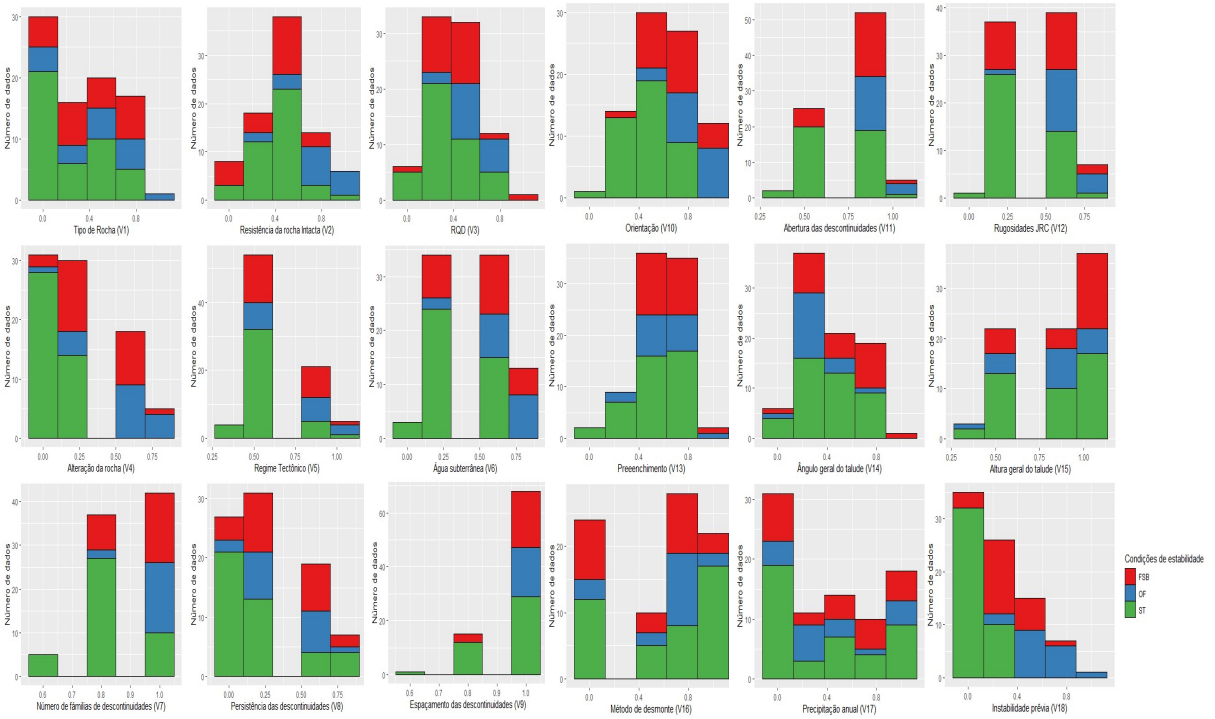
Figura 24 - Gráfico de Violino para o banco de dados não balanceado.



Fonte - R CORE TEAM ,(2016).

Após esta etapa, foi possível visualizar a distribuição das amostras de cada classe de instabilidade dentro de cada variável preditiva como pode ser visto na Figura 25. Após realizar o estudo estatístico dos dados, foi possível dar início ao desenvolvimento e criação dos 6 primeiros modelos propostos.

Figura 25 - Histograma dos dados não balanceados.



Fonte - R CORE TEAM ,(2016).

4.2. Modelo Geral (MG)

4.2.1. TREINO DO MODELO

Para o Modelo Geral, todas as variáveis do banco de dados balanceado foram utilizadas como pode ser visto na Tabela 8, no entanto por uma questão de aplicação dos modelos no software, é necessário que haja uma formatação específica dos nomes das variáveis para que seja possível a interpretação deste pelo modelo e por isso foi necessário criar essa relação.

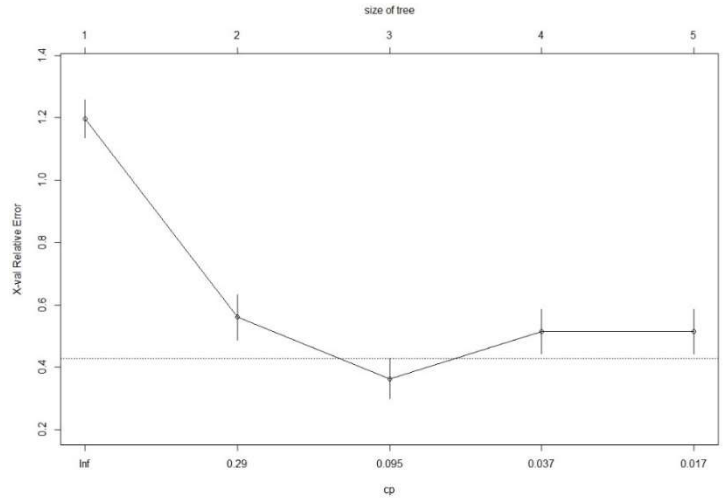
Tabela 8 - Variáveis do Modelo Geral e suas identificações para o modelamento.

Modelo Geral (MG)	**Relação das identificações**
Tipo de Rocha (V1)	Rock_type_V1
Resistência da rocha intacta (V2)	Intact_Rock_Strength_V2
RQD (V3)	RQD_V3
Alteração (V4)	Weathering_V4
Regime tectônico (V5)	Tectonic_Regime_V5
Água subterrânea (V6)	Groundwater_V6
Número de famílias (V7)	Number_of_sets_V7
Persistência (V8)	Persistence_V8
Espaçamento (V9)	Spacing_V9
Orientação (V10)	Orientation_V10
Abertura (V11)	Aperture_V11
JRC (V12)	Roughness_JRC_macro_V12
Preenchimento (V13)	Filling_V13
Ângulo geral (V14)	Overall_angle_degrees_V14
Altura geral (V15)	Overall_Height_meters_V15
Método de desmonte (V16)	Blasting_Method_V16
Precipitação anual (V17)	Precipitation_mmperyear_V17
Instabilidade prévia (V18)	Previous_Instability_V18

O Modelo Geral (MG) foi criado utilizando uma partição do banco de dados balanceado para criar o conjunto de treino para o modelo, esse conjunto utilizou 80% dos dados do banco respeitando as distribuições não apenas das variáveis, mas também das classes de resposta de estabilidade, os demais 20% foram utilizados para criar um conjunto de dados para testar a qualidade do modelo final.

O Controle de Poda (CP) que determinou o número de nós criados é uma relação que existe entre as importâncias das variáveis e a Pureza de Gini em cada nó a ponto de minimizar o erro do modelo, esta etapa pode ser vista na Figura 26 onde o CP com o menor erro é selecionado para a poda da árvore.

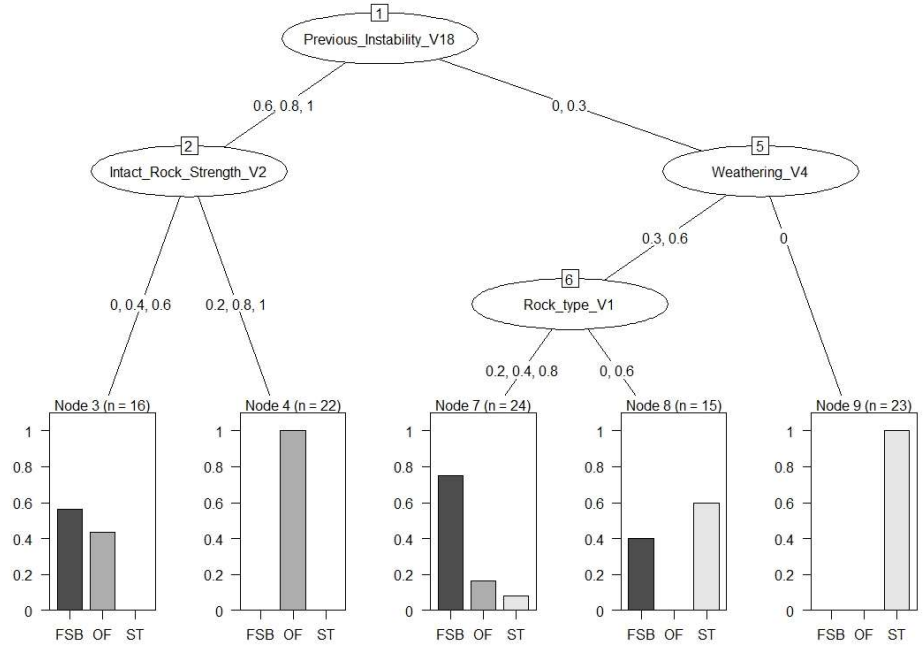
Figura 26 - Gráfico da relação do CP com erro relativo do Modelo Geral para determinação do tamanho da árvore.



Fonte - R CORE TEAM ,(2016).

A árvore de decisão final do MG está apresentada na Figura 27, sendo possível notar que o modelo selecionou apenas as variáveis V1, V2, V4 e V18 para a criação da árvore. Neste caso, a poda foi feita de acordo com as importâncias e a Pureza de Gini que minimize o erro do modelo ao mesmo tempo que diminui o número de nós criados como já foi explicado anteriormente.

Figura 27 - Árvore de decisão para o Modelo Geral.



Fonte - R CORE TEAM ,(2016).

A interpretação da árvore de decisão é muito simples, a amostra analisada passa pelo primeiro nó e a variável dos dois é comparada de acordo com as especificações do ramo

esquerdo e direito. Dependendo do valor contido na amostra este é encaminhado para uma direção. Esse processo se repete até que a amostra alcance o nível mais inferior da árvore.

O tamanho da árvore é determinado de acordo com o número de níveis de nós que existem. As variáveis V2 e V4 estão no mesmo nível. Elas são consideradas como apenas um nível. Para validar o modelo criado, o conjunto de amostras de teste foi utilizado para se determinar a eficácia do modelo final.

4.2.2. TESTE DO MODELO MG

Para testar o modelo, o conjunto de teste constituído por 20% do banco de dados balanceado foi submetido à interpretação da árvore de decisão. A partir da comparação das estimativas dessas amostras pela árvore com os valores reais desses dados é possível determinar a acurácia do modelo por meio de uma matriz de confusão. A Figura 28 está representando todos os dados obtidos pela estimativa dos dados de teste além da matriz de confusão sendo que as colunas representam as estimativas e as linhas representam os valores reais. A acurácia do Modelo Geral foi 83,33% podendo ser considerado uma boa estimativa se levar em conta que o número de amostras testados é pequeno.

Figura 28 - Matriz de confusão e dados estatísticos do teste com 20% dos dados balanceados como teste.

```
Confusion Matrix and Statistics

     predict_unseen2
      FSB OF ST
  FSB   6  0  2
  OF    1  6  1
  ST    0  0  8

Overall Statistics

               Accuracy : 0.8333
                 95% CI : (0.6262, 0.9526)
    No Information Rate : 0.4583
    P-Value [Acc > NIR] : 0.0001808

                  Kappa : 0.75

 Mcnemar's Test P-Value : 0.2614641

Statistics by Class:

                     Class: FSB Class: OF Class: ST
Sensitivity              0.8571    1.0000    0.7273
Specificity              0.8824    0.8889    1.0000
Pos Pred Value           0.7500    0.7500    1.0000
Neg Pred Value           0.9375    1.0000    0.8125
Prevalence               0.2917    0.2500    0.4583
Detection Rate           0.2500    0.2500    0.3333
Detection Prevalence     0.3333    0.3333    0.3333
Balanced Accuracy        0.8697    0.9444    0.8636
```

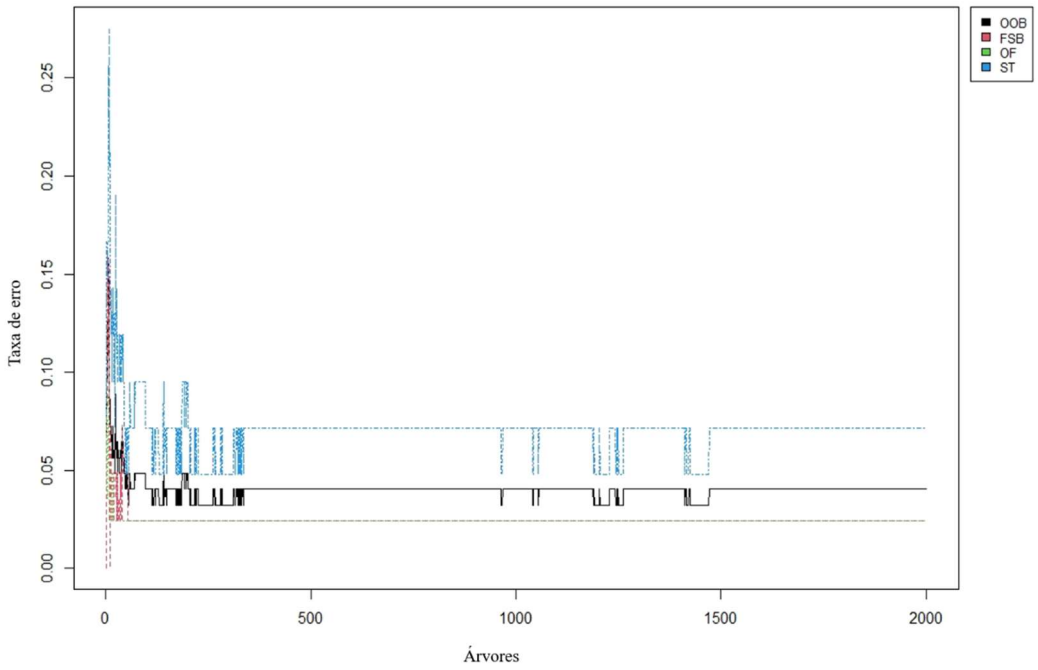
Fonte - R CORE TEAM ,(2016).

4.3. Modelo Matemático (MM)

4.3.1. SELEÇÃO DE VARIÁVEIS PELO *RANDOM FOREST*

O modelo matemático é constituído das variáveis selecionadas pelo *Random forest*. Para a seleção das variáveis o RF foi realizado com os dados balanceados. 3000 árvores foram criadas com 2000 interações entre elas. O objetivo do modelo é buscar o conjunto de variáveis que minimizem o erro OOB, *Out-of-Bag*, e os erros relativos a cada classe. A Figura 29 apresenta as estimativas dos erros para cada interação, sendo evidenciada a estabilização dos erros a partir de 1500 árvores testadas.

Figura 29 - Relação do número de árvores criados em função do erro relativo dos fatores no RF;



Fonte - R CORE TEAM ,(2016).

De acordo com a Figura 30 é possível ver o decréscimo médio da acurácia de cada variável, com este valor é possível determinar quais são mais relevantes já que este número representa a redução esperada na acurácia relativa do modelo com a remoção dessa variável.

Figura 30 - Relação dos parâmetros do banco de dados com a importância média obtida pelo RF.

	MeanDecreaseAccuracy	MeanDecreaseGini
Rock_type_V1	0.029615914	3.5846030
Intact_Rock_Strength_V2	0.086345034	9.0535160
RQD_V3	0.057325064	5.1384864
Weathering_V4	0.102107111	9.8304047
Tectonic_Regime_V5	0.009664798	1.0925922
Groundwater_V6	0.026552094	3.1870285
Number_of_sets_V7	0.038213618	3.9275982
Persistence_V8	0.021835597	2.5302425
Spacing_V9	0.001123038	0.4093364
Orientation_V10	0.053147871	5.4199651
Aperture_V11	0.020003181	2.2275910
Roughness_JRC_macro_V12	0.017541011	1.7495659
Filling_V13	0.015542123	2.8561544
Overall_angle_degrees_V14	0.043871160	4.2933095
Overall_Height_meters_V15	0.016160614	2.3199408
Blasting_Method_V16	0.025546150	2.8262231
Precipitation_mmperyear_V17	0.021130969	2.5201984
Previous_Instability_V18	0.205091338	19.0074531

Fonte - R CORE TEAM ,(2016).

De acordo com o algoritmo, o valor de corte foi de aproximadamente 0,038 e 3,875 para o decréscimo de acurácia e Índice de Gini respectivamente, ou seja, todos os parâmetros com valor superior a estes entraram na seleção como pode ser visto na Tabela 9.

Tabela 9 - Variáveis selecionadas pelo RF.

Modelo Matemático (MM)
Resistência da rocha intacta (V2)
RQD (V3)
Alteração (V4)
Número de famílias (V7)
Orientação (V10)
Ângulo geral (V14)
Instabilidade prévia (V18)

No entanto, antes de prosseguir para a criação da árvore MM é importante demonstrar a acurácia do modelo RF. Para isso a matriz de confusão e o *bootstrap* são usados para esta tarefa.

4.3.2. VALIDAÇÃO E TESTE DO RANDOM FOREST

A primeira validação foi feita usando a matriz de confusão da mesma forma que foi feita anteriormente, com 20% dos dados balanceados como conjunto de teste e os resultados podem ser vistos na Figura 31.

Figura 31 - Matriz de confusão e dados estatísticos do teste com 20% dos dados balanceados como teste para o RF.

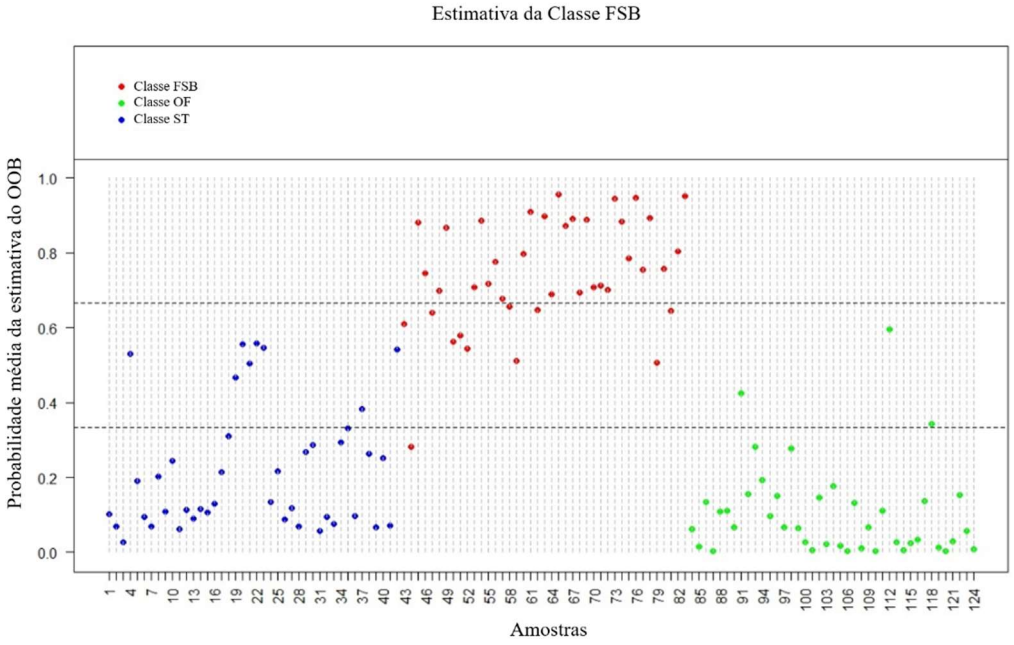
```
          Reference
Prediction FSB OF ST
       FSB   8  0  0
       OF    0  8  0
       ST    0  0  8

Overall Statistics

               Accuracy : 1
                 95% CI : (0.8575, 1)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : 3.541e-12

                  Kappa : 1

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: FSB Class: OF Class: ST
Sensitivity              1.0000    1.0000    1.0000
Specificity              1.0000    1.0000    1.0000
Pos Pred Value           1.0000    1.0000    1.0000
Neg Pred Value           1.0000    1.0000    1.0000
Prevalence               0.3333    0.3333    0.3333
Detection Rate           0.3333    0.3333    0.3333
Detection Prevalence     0.3333    0.3333    0.3333
Balanced Accuracy        1.0000    1.0000    1.0000
```

Fonte - R CORE TEAM ,(2016).

Com uma acurácia de 100%, todas as 24 amostras foram classificadas corretamente comprovando a eficácia do modelo final para a seleção das variáveis mais importantes, porém para uma estimativa mais robusta do modelo é possível utilizar o *bootstrap* para criar novas amostras e testá-las no modelo. Para esta técnica foram criadas 124 novas amostras com 3000 árvores e 2000 interações e 100 interações do tipo *bootstrap.* Com esse método é possível calcular a probabilidade de uma amostra ser classificada em cada fator de estabilidade e dessa forma calcular o erro preditivo de um modelo já criado anteriormente.

Na Figura 32 é possível ver a probabilidade de as 124 amostras serem classificadas com FSB. Como esperado, já que a classificação FSB é um intermediário de estabilidade entre o estável (ST) e o instável (OF) naturalmente se espera que haverá dados podendo ser classificados erroneamente nessa zona de transição, porém mesmo nestas condições o modelo alcançou um bom resultado. Esta zona de transição será novamente abordada quando o PCA for discutido. Algo importante a acrescentar é que a cor de cada ponto representa a classificação real da amostra e o eixo Y do gráfico representa a probabilidade de o modelo estimar, neste caso, a amostra com sendo FSB.

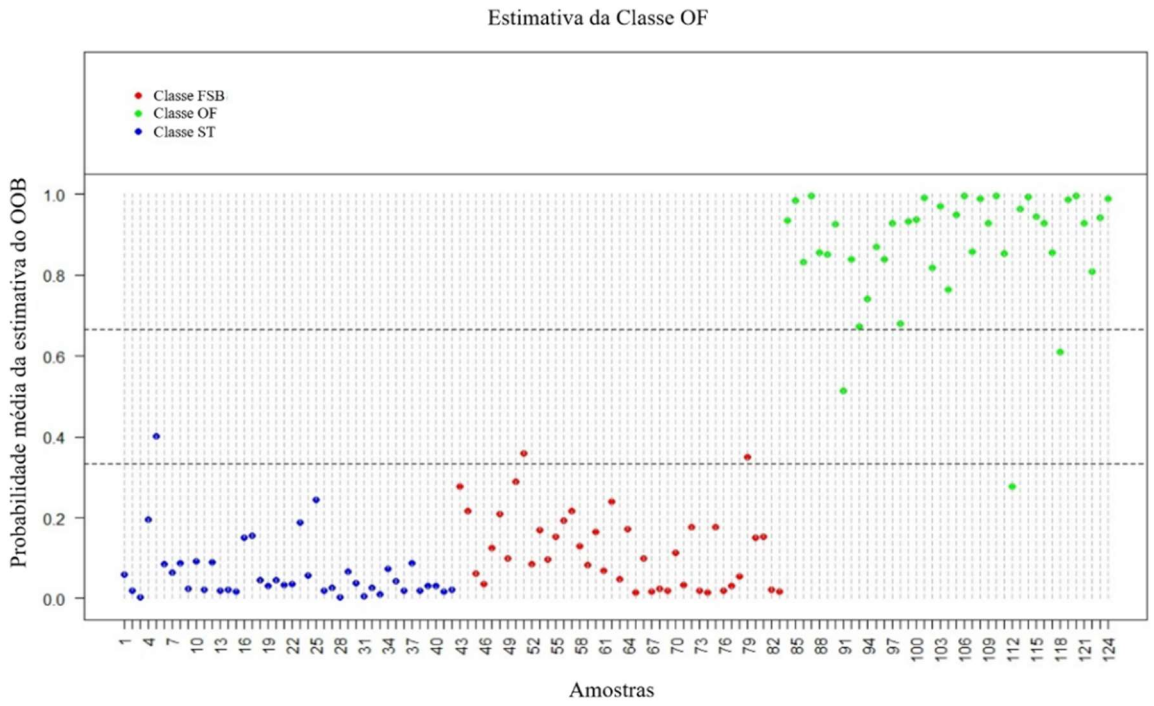
Figura 32 - Gráfico da estimativa probabilística das amostras do bootstrap para a classe de estabilidade FSB.



Fonte - R CORE TEAM ,(2016).

Partindo para a próxima classe, na Figura 33 é possível analisar os mesmos aspectos vistos anteriormente, porém para o OF, neste caso é perceptível a diferença que existe entre a predição já que os pontos foram muito mais segregados em suas reais classes.

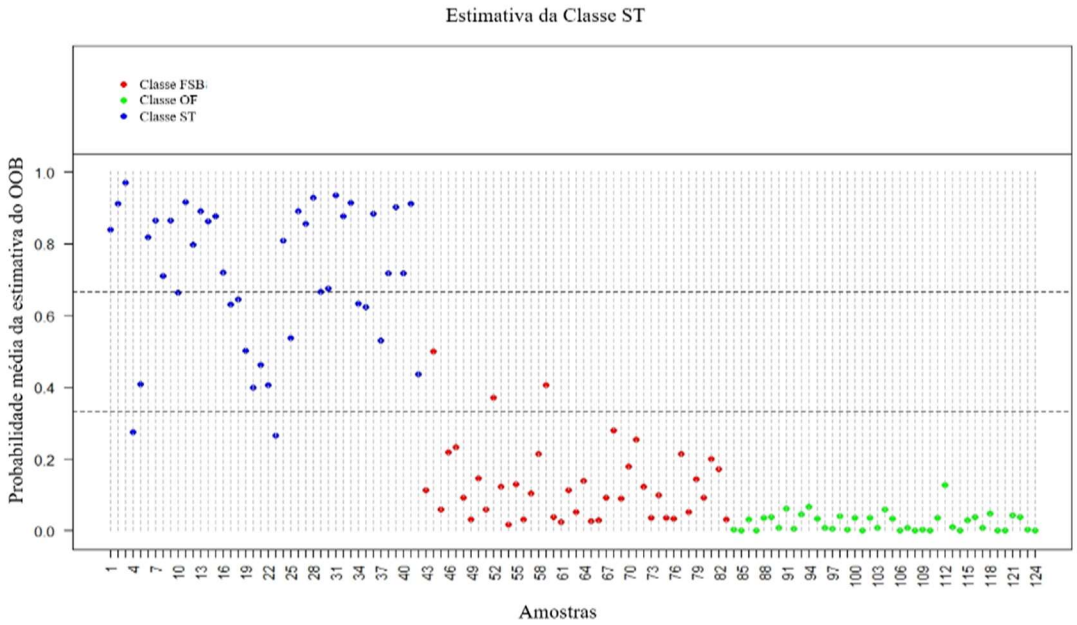
Figura 33 - Gráfico da estimativa probabilística das amostras do bootstrap para a classe de estabilidade OF.



Fonte - R CORE TEAM ,(2016).

Por fim, na Figura 34 está representado a classe ST de estabilidade, neste caso é possível ver a existência de amostras azuis com uma menor chance de classificação correta se mesclando com as amostras do tipo FSB. Isso ocorre, pois, estes pontos estão em uma zona de transição com o FSB muito mais sobrepostos se comparado com a mesma interação das amostras FSB e OF.

Figura 34 - Gráfico da estimativa probabilística das amostras do *bootstrap* para a classe de estabilidade ST.



Fonte - R CORE TEAM ,(2016).

ZareNaghadehi *et al*. (2013) também observaram esta condição em seus estudos. No entanto, mais sobre essa condição foi abordado no PCA, além de outros resultados obtidos pela manipulação do banco de dados de acordo com os resultados dos 4 modelos de árvore de decisão neste trabalho. Com todos estes dados foi possível calcular o erro preditivo do modelo do RF como pode ser visto na Figura 35.

Figura 35 - Resultados do bootstrap do RF para 100 interações.

Resultados do *Bootstrap*

Estimativa do erro da predição do Bootstrap para 100 interações:

```
0.08250251
```

Número de variáveis em cada floresta no Bootstrap:

```
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  3.00    5.00    6.00    6.79    7.00   18.00
```

Sobreposição das florestas do Bootstrap com todos os dados:

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.5774  0.6804  0.7778  0.7785  0.8819  1.0000
```

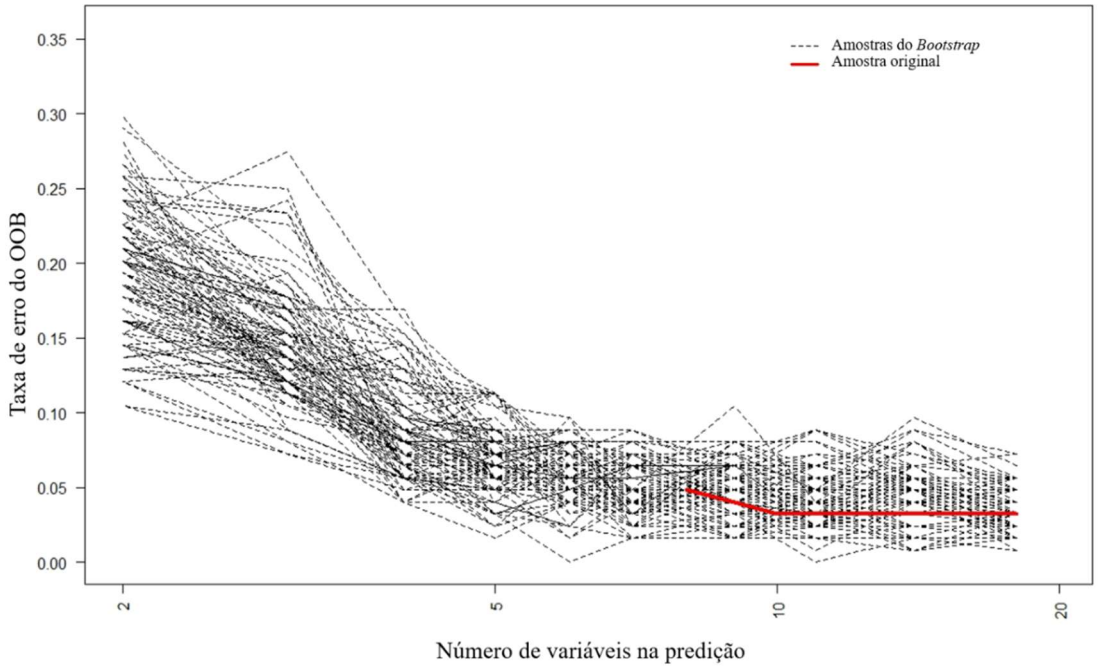
Fonte - R CORE TEAM ,(2016).

O erro estimado pelo *bootstrap* foi de 8,25% para o modelo, sendo assim o resultado do RF foi satisfatório e a seleção das variáveis mais importantes pode ser considerada para a modelagem de MM. Mesmo tendo saído um pouco da acurácia calculada pelo conjunto de

teste, é possível afirmar que este é próximo por causa da natureza do método do *boostrap* de calcular o erro preditivo.

Na Figura 36 é possível ver o resultado do erro de cada uma das 100 interações feitas para o desenvolvimento das 124 amostras. Como cada interação cria 124 amostras novas, cada uma delas possuí um erro relativo que é somado para determinar a probabilidade de classificação visto nas Figuras 32, 33 e 34. As linhas tracejadas representam as 100 interações e a linha vermelha representa o erro original associado as amostras balanceadas usadas para a criação do RF. Como nesses métodos cada variável possuí uma importância associada que diminui o erro relativo do modelo, é esperado que o acréscimo de variáveis diminua o erro relativo do preditor, como pode ser visto na Figura 36.

Figura 36 - Erro relativo das interações em função do número de variáveis usadas em cada *bootstrap*.



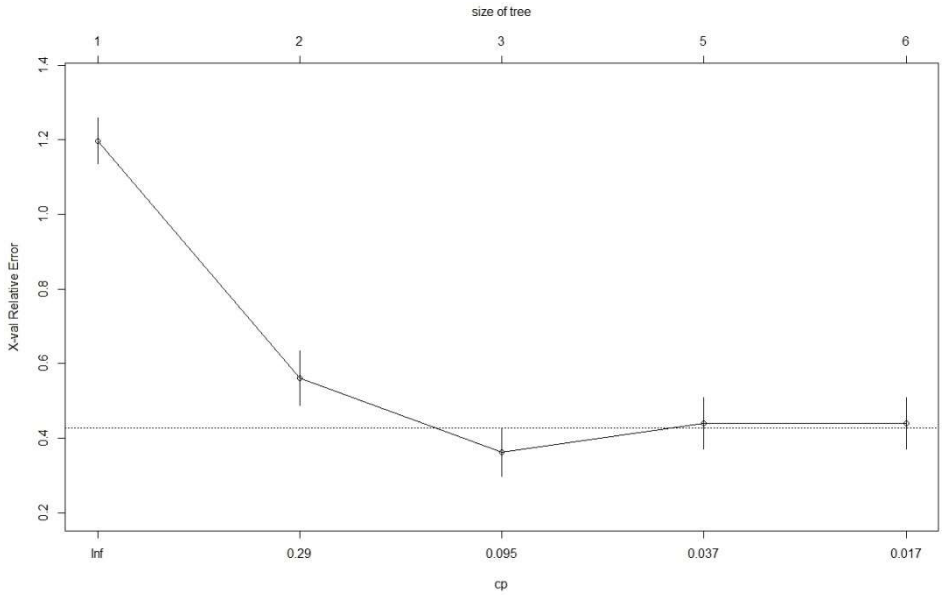
Fonte - R CORE TEAM ,(2016).

Levando em conta tudo que foi discutido anteriormente, foi possível dar andamento e criar a árvore de decisão para o Modelo Matemático.

4.3.3. TREINO DO MODELO MM

Para treinar o MM as mesmas regras de partição do banco de dados foram usadas, no entanto desta vez as amostras só possuem as informações referentes as variáveis selecionadas apresentadas no tópico 4.3.1. O Controle de Poda (CP) que determinou o número de nós criados é uma relação que existe entre as importâncias das variáveis e a Pureza de Gini em cada nó a ponto de minimizar o erro do modelo ao máximo. Esta etapa pode ser vista na Figura 37 onde o CP com o menor erro é selecionado para a poda da árvore.

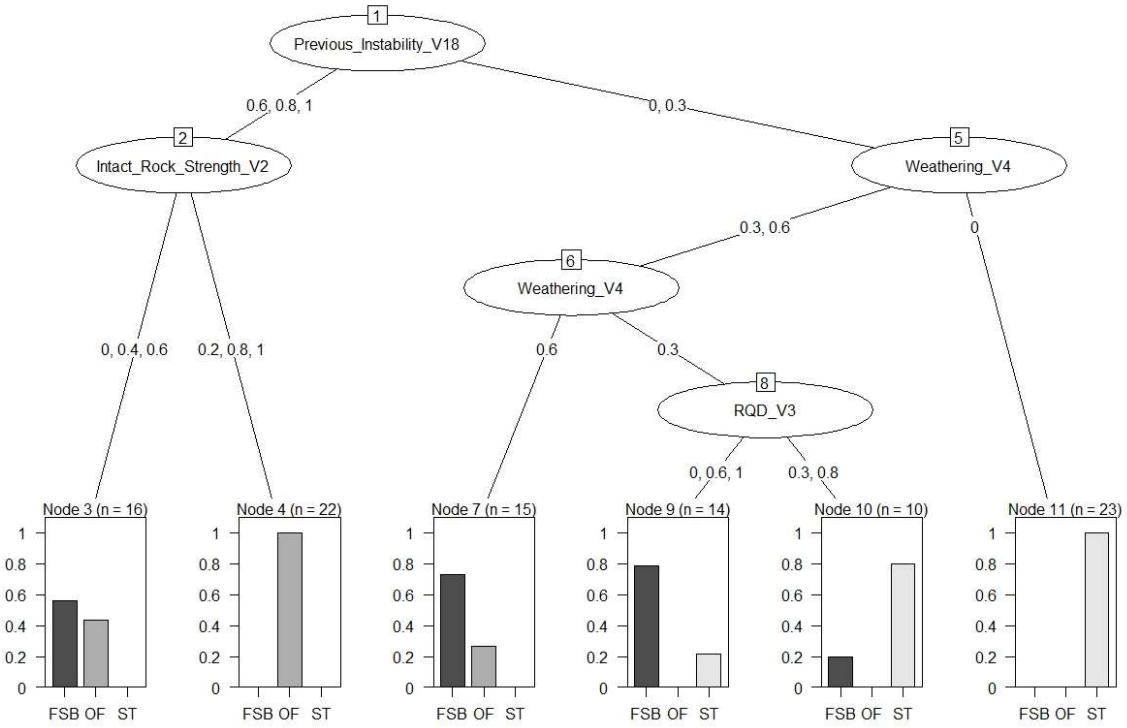
Figura 37 - Gráfico do CP em função do erro relativo da árvore e seu tamanho.



Fonte - R CORE TEAM ,(2016).

O conjunto de treino com 80% dos dados foi usado e gerou a árvore apresentada na Figura 38. Desta vez as variáveis usadas após a poda foram as V2, V3, V4 e V18. Além disso, é perceptível que no MM a classificação do modelo tende apenas a errar entre fatores da zona de transição, diferente do que ocorreu no MG, ou seja, os erros do modelo matemático são do tipo FSB/ST ou FSB/OF. No entanto, há uma diferença na interpretação do CP nesta árvore. Como os nós 5 e 6 usam a mesma variável para classificar as amostras e eles estão conectados diretamente, o CP os considera como dois nós do mesmo nível, por isso que o seu valor do CP foi o mesmo do MG.

Figura 38 - Modelo de árvore para o MM com os dados balanceados e variáveis selecionadas pelo RF.



Fonte - R CORE TEAM ,(2016).

Além disso, em alguns ramos da árvore existem classificações fora do intervalo lógico de distribuição, exemplo disso é o que correu no nó 8, o ramo da esquerda tem como resposta os dados de RQD iguais à 0, 0.6 e 1 já o ramo da direita tem como resposta os valores 0.3 e 0.8. Pela interpretação superficial das informações esperava-se que os valores menores estivessem juntos para classificar um talude estável e os valores maiores para classificar taludes menos estáveis, no entanto houve uma mescla desses valores.

O mesmo ocorreu no MG e esta situação irá se repetir nos demais modelos. Porém esta situação será também explicada nas discussões do PCA. Como feito para o MG, a validação do MM será feito com a predição dos 20% dos dados restantes da partição

4.3.4. TESTE DO MODELO MG

Na figura 39 é possível ver os resultados do teste do modelo e como foi sua acurácia. Se comparado com a do MG, a acurácia do MM para a predição dos dados de teste foi igual alcançando um valor de 83,33%. Isto mostra o poder preditivo das variáveis selecionadas pelo RF, ou seja, o MG com 18 variáveis alcança acurácia de 0.83, e o MM com 7 variáveis apenas alcança a mesma acurácia.

Figura 39 - Matriz de confusão e dados estatísticos do teste com 20% dos dados balanceados como teste.

```
Confusion Matrix and Statistics

          predict_unseen1
           FSB OF ST
      FSB    5  0  3
      OF     1  7  0
      ST     0  0  8

Overall Statistics

                    Accuracy : 0.8333
                      95% CI : (0.6262, 0.9526)
        No Information Rate : 0.4583
        P-Value [Acc > NIR] : 0.0001808

                       Kappa : 0.75

     Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: FSB Class: OF Class: ST
Sensitivity              0.8333    1.0000    0.7273
Specificity              0.8333    0.9412    1.0000
Pos Pred Value           0.6250    0.8750    1.0000
Neg Pred Value           0.9375    1.0000    0.8125
Prevalence               0.2500    0.2917    0.4583
Detection Rate           0.2083    0.2917    0.3333
Detection Prevalence     0.3333    0.3333    0.3333
Balanced Accuracy        0.8333    0.9706    0.8636
```

Fonte - R CORE TEAM ,(2016).

Estas questões para adequação de todos os modelos serão discutidas no capítulo onde será abordado o PCA. Importantes descobertas foram feitas com esta técnica que alteraram os resultados finais dos modelos. Dando continuidade aos modelos, o próximo será o MQ, que se baseou nas variáveis usadas no Q-slope desenvolvido por Bar & Barton (2017).

4.4. Modelo Q-slope (MQ)

4.4.1. DETERMINAÇÃO DAS VARIÁVEIS

Para selecionar as variáveis equivalentes no banco de dados, cada componente da equação do Q-slope foi comparada e relacionada com suas equivalências.

$$Q_{slope} = \frac{RQD}{J_n} \text{ x } (\frac{J_r}{J_a})_0 \text{ x } \frac{J_{wice}}{SRF_{slope}}$$

- RQD → RQD (V3)
- J_n → Número de famílias de descontinuidade (V7)
- J_r → Espaçamento (V9) + Rugosidade (V12) + Orientação (V10) + Abertura (V11)
- J_a→ Alteração (V4) + Preenchimento (V13)
- J_{wice} → Precipitação anual (V17) + Resistência da rocha intacta (V2)
- SRF_a → Alteração (V4) + Método de desmonte (V16)
- SRF_b → Resistência da rocha intacta (V2)
- SRF_c → Espaçamento (V9) + Rugosidade (V12) + Orientação (V10) + RQD (V3)

Organizando essas variáveis e retirando as repetidas os seguintes parâmetros foram selecionados de acordo com a Tabela 10. Com esta seleção foi possível repetir os mesmos passos dos modelos anteriores e criar o MQ.

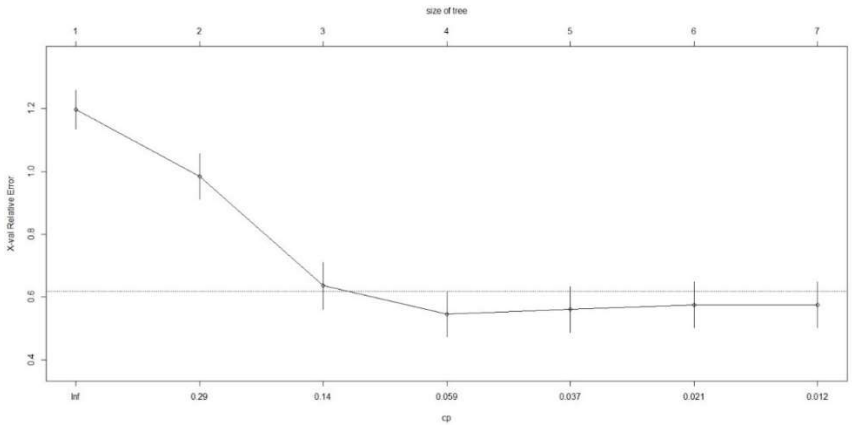
Tabela 10 - Variáveis selecionadas para MQ.

Modelo Q-slope (MQ)
Resistência da rocha intacta (V2)
RQD (V3)
Alteração (V4)
Número de famílias (V7)
Espaçamento (V9)
Orientação (V10)
Abertura (V11)
JRC (V12)
Preenchimento (V13)
Método de desmonte (V16)
Precipitação anual (V17)

4.4.2. TREINO DO MODELO MQ

A mesma regra de partição para os conjuntos de treino e teste foi usada. Já o CP dessa árvore pode ser visto na Figura 40, nada fora do normal foi detectado no gráfico, apenas a forma com que o erro se comporta de acordo com o aumento do tamanho da árvore.

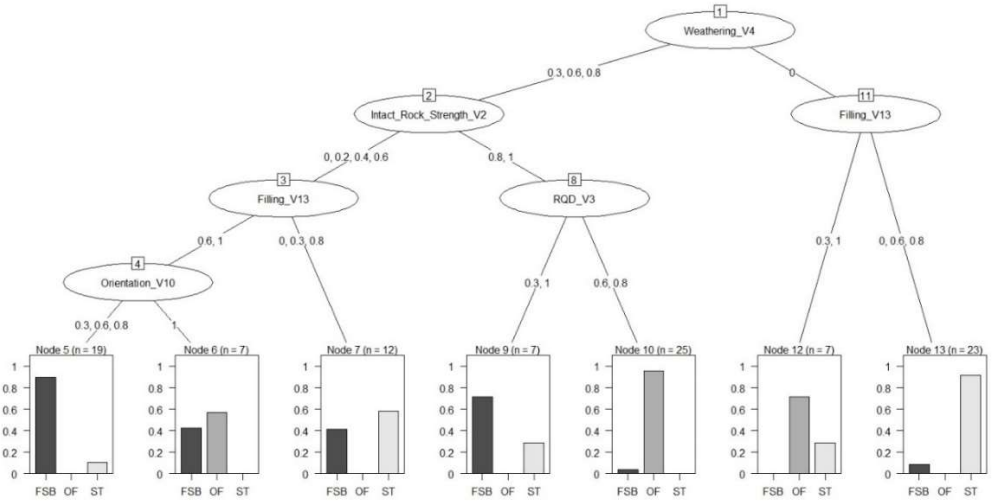
Figura 40 - Gráfico do CP para o Modelo Q-slope.



Fonte - R CORE TEAM ,(2016).

Com o CP definido, a árvore foi criada e no MQ as variáveis usadas na árvore após o controle de poda foram as V2, V3, V4, V10 e V13 como pode ser visto na Figura 41. Outro aspecto importante de notar é que o mesmo efeito visto nos modelos anteriores ocorre também neste, há ramos da árvore que possuem classificadores misturados entre a ordem lógica esperada na classificação. Isso pode ser visto no nó 8 em que o ramo esquerdo aceita como respostas 0.3 e 1 sendo que seria esperado que o segundo valor estivesse no ramo da direita.

Figura 41 - Árvore de decisões do Modelo Q-slope.



Fonte - R CORE TEAM ,(2016).

A presença dessas incoerências já poderia ser considerado algo anômalo com a análise apenas dos dois modelos anteriores e para entrar a fundo neste caso é que o PCA foi feito. Após a modelagem da árvore foi necessário realizar a etapa de teste do modelo.

4.4.3. TESTE DO MODELO MQ

Nada foi alterado em relação ao conjunto de teste usado para calcular a acurácia do MQ. 20% dos dados balanceados foram usados e o resultado pode ser visto na Figura 42. Desta vez, o valor da acurácia foi de 79,17%, inferior se comparado aos demais modelos.

Figura 42 - Matriz de confusão e dados estatísticos do teste com 20% dos dados balanceados como teste.

```
Confusion Matrix and Statistics

          predict_unseen3
           FSB OF ST
  FSB    6  1  1
  OF     1  7  0
  ST     2  0  6

Overall Statistics

               Accuracy : 0.7917
                 95% CI : (0.5785, 0.9287)
    No Information Rate : 0.375
    P-Value [Acc > NIR] : 3.82e-05

                  Kappa : 0.6875

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: FSB Class: OF Class: ST
Sensitivity              0.6667    0.8750    0.8571
Specificity              0.8667    0.9375    0.8824
Pos Pred Value           0.7500    0.8750    0.7500
Neg Pred Value           0.8125    0.9375    0.9375
Prevalence               0.3750    0.3333    0.2917
Detection Rate           0.2500    0.2917    0.2500
Detection Prevalence     0.3333    0.3333    0.3333
Balanced Accuracy        0.7667    0.9062    0.8697
```

Fonte - R CORE TEAM ,(2016).

Com os resultados da validação do MQ foi possível dar prosseguimento para o último modelo antes da análise do PCA.

4.5. Modelo SANTOS *et al.* (2021) (MS)

4.5.1. SELEÇÃO DAS VARIÁVEIS

Nesta seleção foram usadas as variáveis selecionadas por Santos *et al*. (2020) de acordo com os resultados obtidos pelos autores. Sua seleção foi baseada nos métodos explicados anteriormente e pode ser visto na Tabela 11. Além disso este conjunto possuí o menor número de parâmetros entre todos os modelos, sendo interessante para analisar o efeito disso no resultado final da árvore.

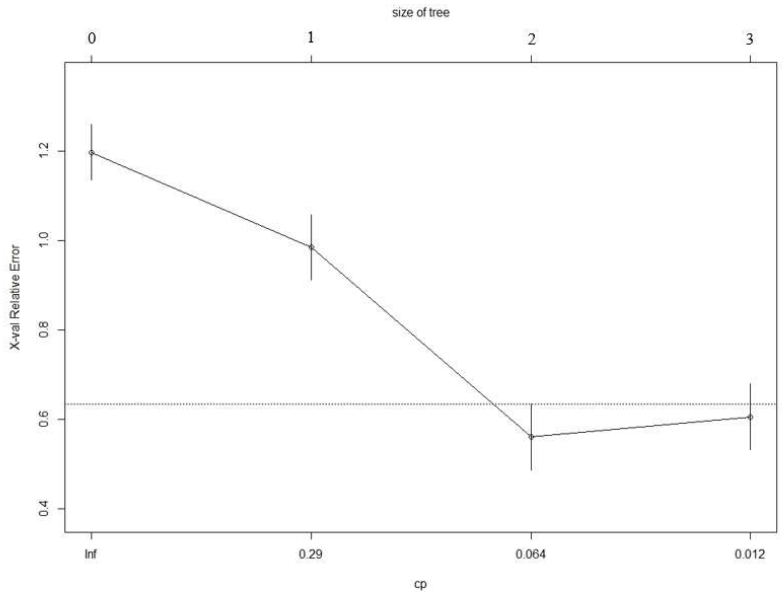
Tabela 11 - Seleção das variáveis para a árvore MS.

Modelo SANTOS *et al.* (2020) (MS)
Resistência da rocha intacta (V2)
Alteração (V4)
Água subterrânea (V6)
Persistência (V8)
Espaçamento (V9)
Abertura (V11)

4.5.2. TREINO DO MODELO MS

Para o treino, novamente, 80% dos dados balanceados e contendo apenas as variáveis selecionadas foram usadas para treinar o modelo. O CP desta árvore pode ser visto na Figura 43 acompanhado do erro relativo do modelo.

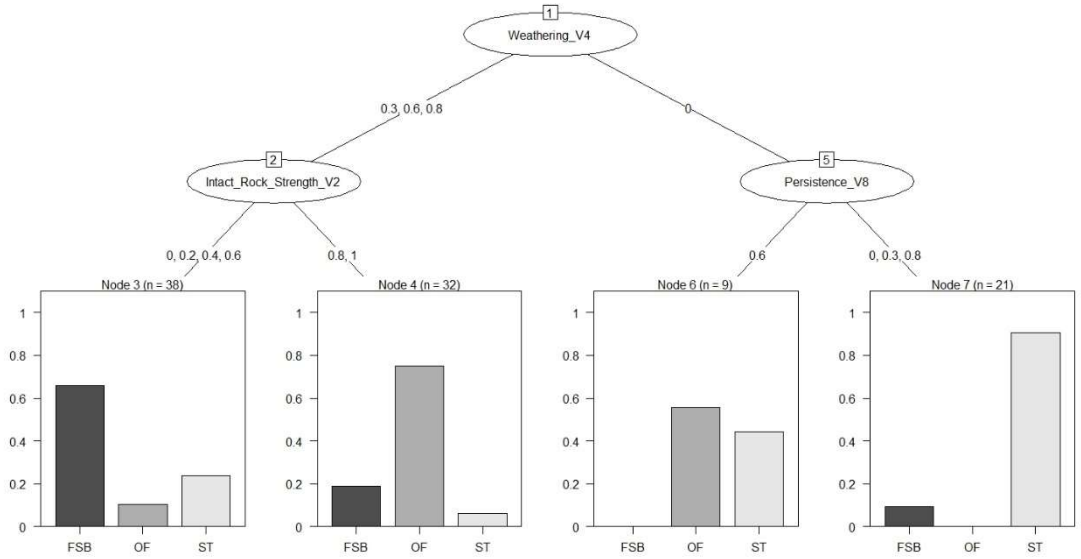
Figura 43 - Gráfico do CP para o modelo SANTOS.



Fonte - R CORE TEAM ,(2016).

A árvore do MS pode ser vista na Figura 44. A primeira impressão clara a se notar foi a diminuição da árvore, desta vez apenas as variáveis V2, V4 e V8 foram utilizadas, provavelmente reflexo do menor número de parâmetros que compõem o banco de dados usados nesta modelagem.

Figura 44 - Árvore de decisões do MS.



Fonte - R CORE TEAM ,(2016).

Além disso, novamente a mescla entre os valores dos ramos se repetiu neste modelo, isso enfatiza mais ainda a possibilidade de algum tipo de desvio nas amostras do banco de dados que será mais discutido posteriormente.

4.5.3. TESTE DO MODELO MS

Para finalizar os resultados do MS, o teste do modelo foi feito usando os 20% dos dados restantes como pode ser visto na Figura 45. Para o teste, com o conjunto dos dados de teste, a acurácia foi de 83,33%. Esta acurácia alcançou um resultado próximo dos demais modelos tendo uma capacidade de estimação satisfatória.

Figura 45 - Matriz de confusão e dados estatísticos do teste com 20% dos dados balanceados como teste.

```
          predict_unseen6
           FSB OF ST
      FSB   8  0  0
      OF    1  7  0
      ST    3  0  5

    Overall Statistics

                   Accuracy : 0.8333
                     95% CI : (0.6262, 0.9526)
        No Information Rate : 0.5
        P-Value [Acc > NIR] : 0.0007719

                      Kappa : 0.75

     Mcnemar's Test P-Value : NA

    Statistics by Class:

                         Class: FSB Class: OF Class: ST
    Sensitivity              0.6667    1.0000    1.0000
    Specificity              1.0000    0.9412    0.8421
    Pos Pred Value           1.0000    0.8750    0.6250
    Neg Pred Value           0.7500    1.0000    1.0000
    Prevalence               0.5000    0.2917    0.2083
    Detection Rate           0.3333    0.2917    0.2083
    Detection Prevalence     0.3333    0.3333    0.3333
    Balanced Accuracy        0.8333    0.9706    0.9211
```

Fonte - R CORE TEAM ,(2016).

Após a apresentação dos 4 modelos propostos é possível dar prosseguimento e iniciar a discussão dos resultados do PCA, no entanto é preciso discutir um aspecto dos erros dos modelos para compreender a importância da análise da componente principal neste trabalho.

4.6. Resultados da Análise das Componentes Principais (PCA)

4.6.1. CLASSIFICAÇÃO DOS ERROS EM PROBLEMAS DE ESTABILIDADE DE TALUDE

Diferente do que em outros estudos usando modelos preditivos, os erros para o modelamento para estabilidade de taludes podem ser classificados como: erros perigosos e erros não perigosos. Além disso, um aspecto relacionado com a validação utilizando uma parcela dos dados como amostra de teste pode causar uma falsa interpretação da capacidade do modelo, caso haja um viés associado com este conjunto, principalmente quando há poucas amostras nesses dados.

Como a seleção das amostras de teste é feita de forma aleatória, e cada modelo criado possuí uma interpretação diferente dos dados, os diferentes modelos podem receber conjuntos de testes mais “fáceis” de se interpretar.

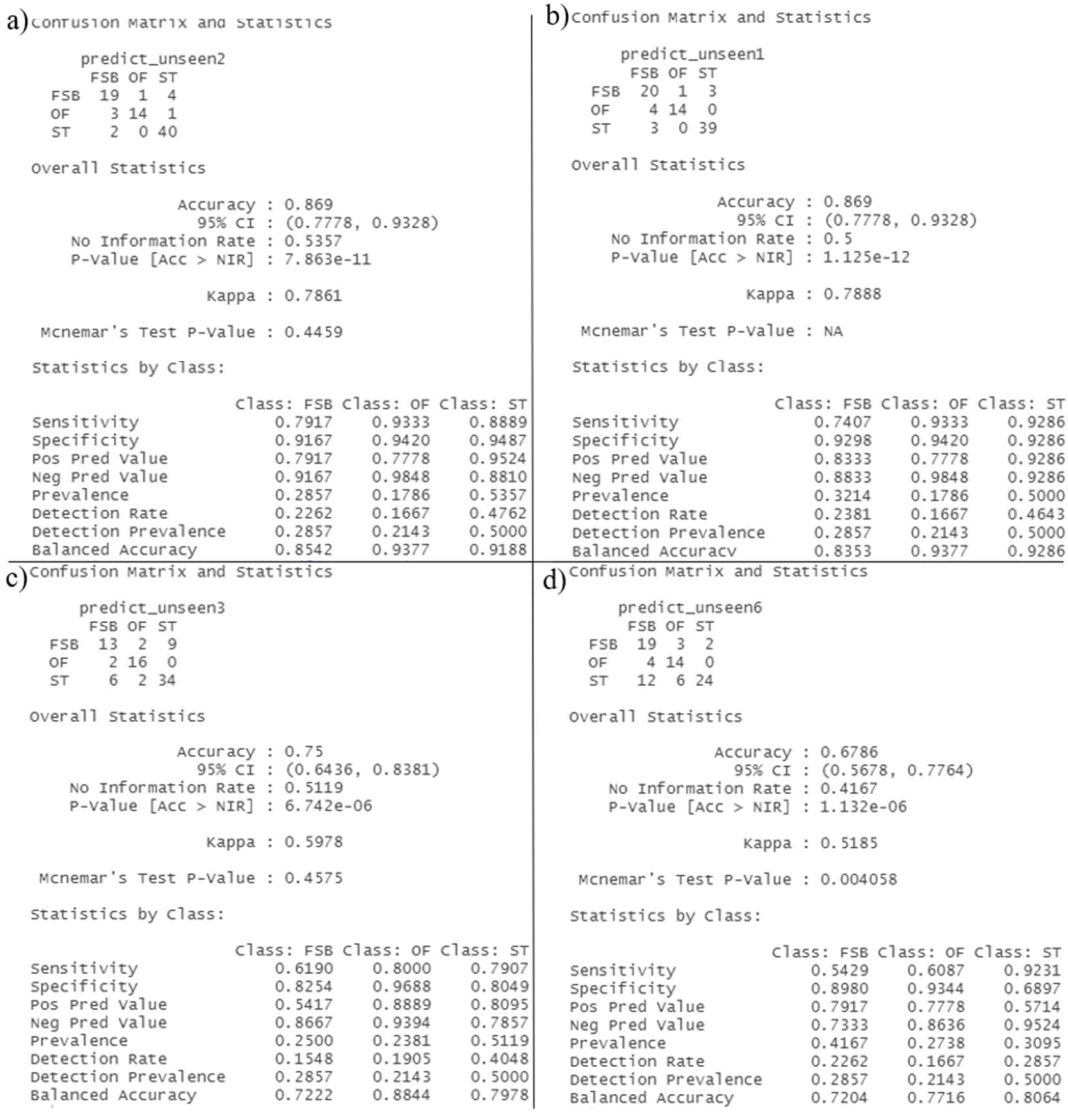
Sendo assim, para se obter uma acurácia mais representativa, novas estimativas foram feitas utilizando as 84 amostras originais para cada modelo estimar suas estabilidades. Importante frisar que os dados balanceados foram utilizados unicamente para treinar os modelos, e não para o estudo dos erros. Pois, como as amostras adicionadas na sua composição são cópias das originais, haveria amostras sendo estimadas mais de uma vez.

Usando estas ideias como princípio, foi possível identificar quais erros de cada modelo eram de cada tipo e além de comparar a acurácia de cada um de forma mais abrangente, também é importante analisar qual é aquele com menor número de erros perigosos, já que esses são muito mais prejudiciais num estimador dessa natureza. Nas novas matrizes de confusão é possível ver a presença de erros que estavam “escondidos” no banco de dados que não estavam sendo contabilizadas por causa do critério de aleatoriedade e pelo número baixo de amostras no conjunto de teste. Por isso, as acurácias usadas para validar os modelos além dos erros usados no PCA foram obtidos nestas novas matrizes de confusão.

No entanto, é importante salientar que os valores obtidos anteriormente com as matrizes de confusão antigas não foram descartados. A utilização delas teve como principal função averiguar a qualidade do modelamento das árvores. Ou seja, estas informações obtidas foram usadas para, principalmente, determinar se as árvores estavam conseguindo aprender com as amostras em seu treino e se não estava ocorrendo problemas como *overfitting* e *underfitting*, muito comuns em aplicações de *machine learning.*

Neste caso, o primeiro é caracterizado por uma alta performance nos dados de treino e uma baixa generalização em outros dados e o segundo é caracterizado por uma baixa performance nos dados de treino e baixa generalização em outros dados. Como todos os modelos alcançaram acurácias satisfatórias para a validação com as amostras de teste, foi possível comprovar a qualidade do modelamento usando os primeiros dados obtidos nas primeiras matrizes de confusão. As novas acurácias obtidas foram apresentadas na Figura 46,

Figura 46 - Matrizes de confusão dos modelos a) MG ; b) MM ; c) MQ e d) MS utilizando 100% dos dados originais como amostras de teste.



a)
```
Confusion Matrix and Statistics

          predict_unseen2
           FSB OF ST
      FSB  19  1  4
      OF    3 14  1
      ST    2  0 40

Overall Statistics

               Accuracy : 0.869
                 95% CI : (0.7778, 0.9328)
    No Information Rate : 0.5357
    P-Value [Acc > NIR] : 7.863e-11

                  Kappa : 0.7861

 Mcnemar's Test P-Value : 0.4459

Statistics by Class:

                     Class: FSB Class: OF Class: ST
Sensitivity              0.7917     0.9333     0.8889
Specificity              0.9167     0.9420     0.9487
Pos Pred Value           0.7917     0.7778     0.9524
Neg Pred Value           0.9167     0.9848     0.8810
Prevalence               0.2857     0.1786     0.5357
Detection Rate           0.2262     0.1667     0.4762
Detection Prevalence     0.2857     0.2143     0.5000
Balanced Accuracy        0.8542     0.9377     0.9188
```

b)
```
Confusion Matrix and Statistics

          predict_unseen1
           FSB OF ST
      FSB  20  1  3
      OF    4 14  0
      ST    3  0 39

Overall Statistics

               Accuracy : 0.869
                 95% CI : (0.7778, 0.9328)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : 1.125e-12

                  Kappa : 0.7888

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: FSB Class: OF Class: ST
Sensitivity              0.7407     0.9333     0.9286
Specificity              0.9298     0.9420     0.9286
Pos Pred Value           0.8333     0.7778     0.9286
Neg Pred Value           0.8833     0.9848     0.9286
Prevalence               0.3214     0.1786     0.5000
Detection Rate           0.2381     0.1667     0.4643
Detection Prevalence     0.2857     0.2143     0.5000
Balanced Accuracy        0.8353     0.9377     0.9286
```

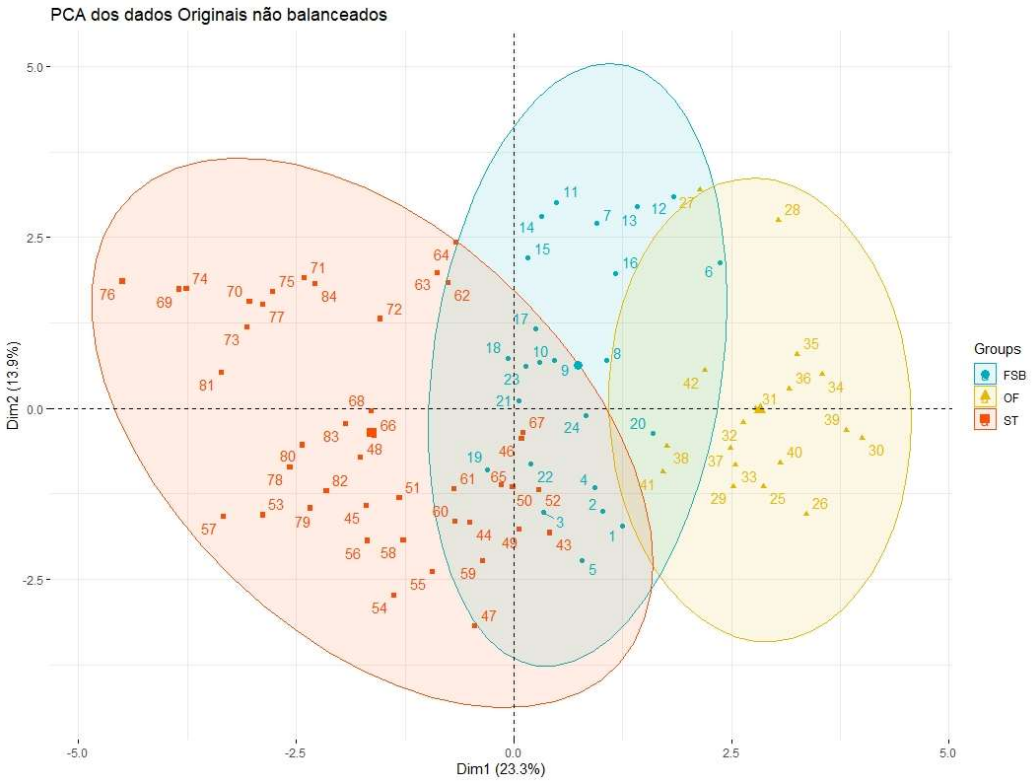
c)
```
Confusion Matrix and Statistics

          predict_unseen3
           FSB OF ST
      FSB  13  2  9
      OF    2 16  0
      ST    6  2 34

Overall Statistics

               Accuracy : 0.75
                 95% CI : (0.6436, 0.8381)
    No Information Rate : 0.5119
    P-Value [Acc > NIR] : 6.742e-06

                  Kappa : 0.5978

 Mcnemar's Test P-Value : 0.4575

Statistics by Class:

                     Class: FSB Class: OF Class: ST
Sensitivity              0.6190     0.8000     0.7907
Specificity              0.8254     0.9688     0.8049
Pos Pred Value           0.5417     0.8889     0.8095
Neg Pred Value           0.8667     0.9394     0.7857
Prevalence               0.2500     0.2381     0.5119
Detection Rate           0.1548     0.1905     0.4048
Detection Prevalence     0.2857     0.2143     0.5000
Balanced Accuracy        0.7222     0.8844     0.7978
```

d)
```
Confusion Matrix and Statistics

          predict_unseen6
           FSB OF ST
      FSB  19  3  2
      OF    4 14  0
      ST   12  6 24

Overall Statistics

               Accuracy : 0.6786
                 95% CI : (0.5678, 0.7764)
    No Information Rate : 0.4167
    P-Value [Acc > NIR] : 1.132e-06

                  Kappa : 0.5185

 Mcnemar's Test P-Value : 0.004058

Statistics by Class:

                     Class: FSB Class: OF Class: ST
Sensitivity              0.5429     0.6087     0.9231
Specificity              0.8980     0.9344     0.6897
Pos Pred Value           0.7917     0.7778     0.5714
Neg Pred Value           0.7333     0.8636     0.9524
Prevalence               0.4167     0.2738     0.3095
Detection Rate           0.2262     0.1667     0.2857
Detection Prevalence     0.2857     0.2143     0.5000
Balanced Accuracy        0.7204     0.7716     0.8064
```

Fonte - R CORE TEAM ,(2016).

Para identificar e classificar cada amostra errada em cada modelo, o PCA foi feito para os 4 estimadores. Primeiramente, será apresentado o gráfico das componentes principais com os elipsoides representando cada classificação real das estabilidades dos pontos. Isso pode ser visto na Figura 47. Pode-se ver facilmente a zona de transição entre as estabilidades ST/FSB e OF/FSB com a intersecção dos elipsoides. Esse entrelaçamento é muito mais presente entre ST/FSB o que já tinha sido visto em alguns modelos de árvores de decisões onde o erro entre essas duas classes é mais acentuado. Os dados usados em todas as análises de PCA para a comparação utiliza apenas as amostras originais, pois um dos objetivos desta etapa foi averiguar algumas nuances desses dados sem nenhuma alteração.

Figura 47 - Gráfico do PCA dos dados originais não balanceados.



Fonte - R CORE TEAM ,(2016).

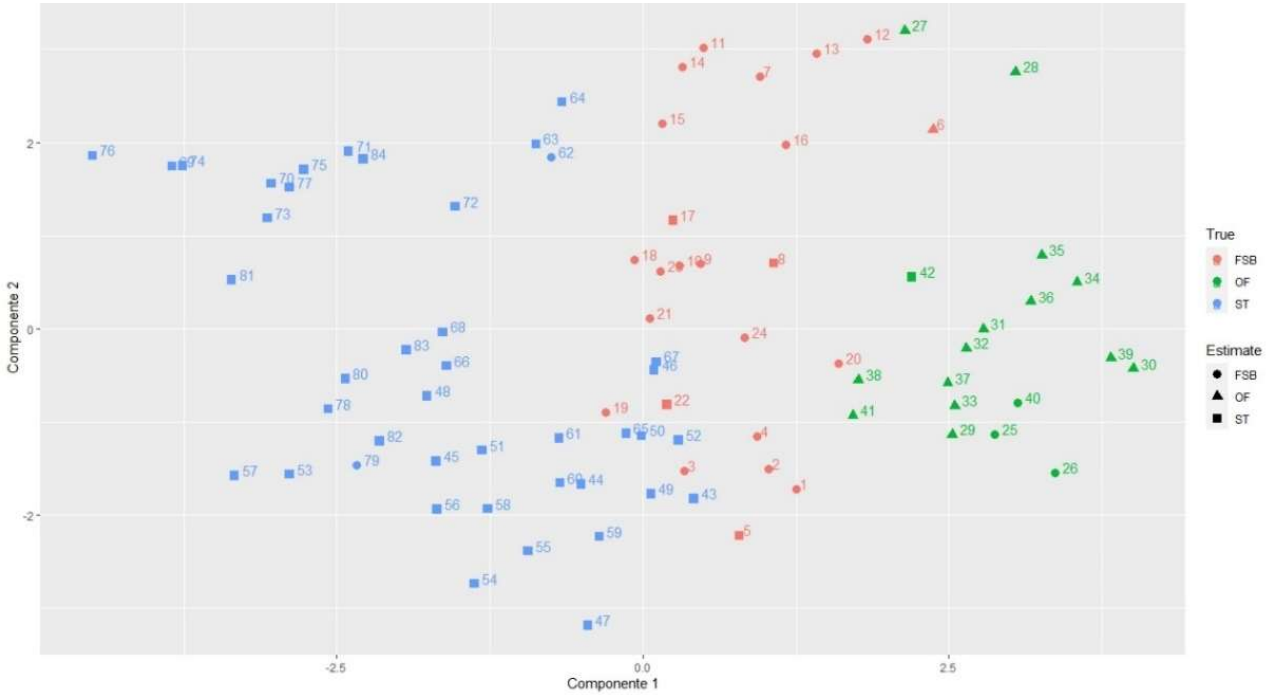
Tendo em mente a posição espacial de cada amostra original é possível determinar facilmente quais pontos foram estimados errados e qual seria sua classificação em cada modelo.

4.6.2. CLASSIFICAÇÃO DOS ERROS

O mesmo gráfico de PCA foi feito novamente, porém desta vez ele apresenta duas classes distintas, a *True* que representa a classificação de estabilidade real pela cor do ponto e a *Estimate* que representa a estimativa do ponto pelo modelo usado representado pelo formato do ponto.

Na Figura 48 estão representados os pontos do Modelo Geral. Este mesmo procedimento foi realizado para todos os demais modelos de árvore de decisões. Além disso, a predição usada nestes erros também foi feita a partir dos dados originais não balanceados e é por isso que nos capítulos anteriores no teste de cada modelo havia duas matrizes de confusão. A segunda foi usada para esta comparação.

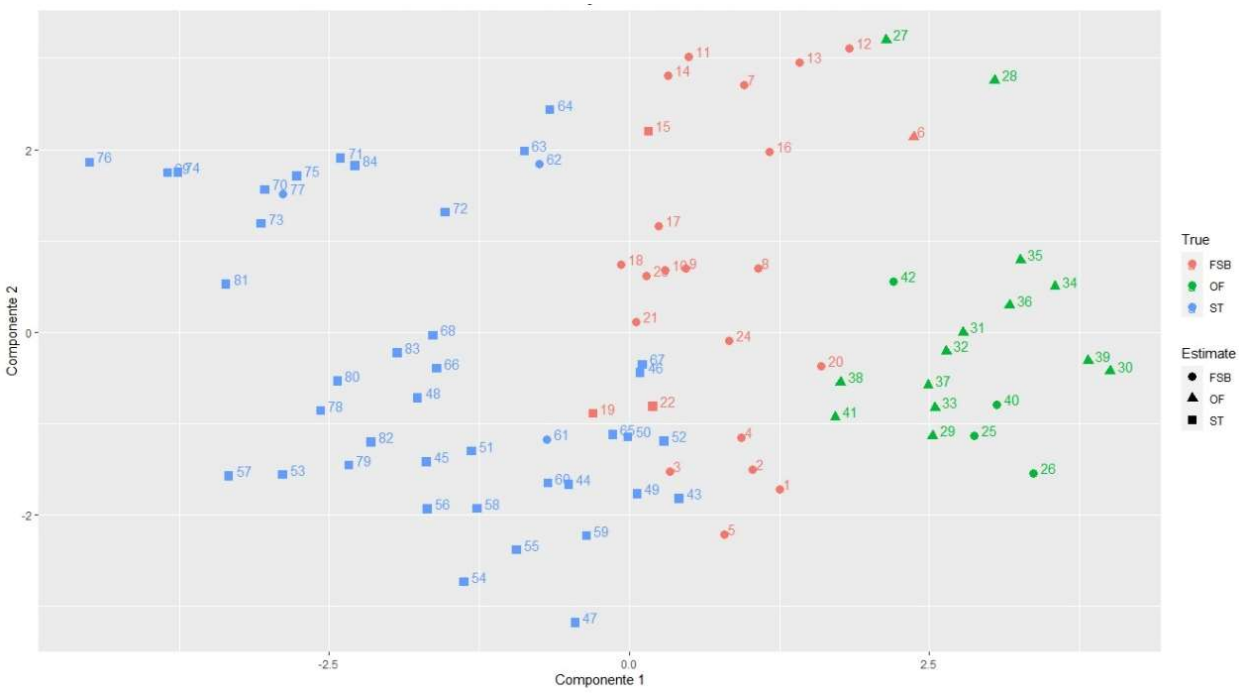
Figura 48 - Gráfico de PCA comparativo entre a estabilidade verdadeira e estimada do Modelo Geral.



Fonte - R CORE TEAM ,(2016).

Analisados os erros do MG, foi feito o mesmo para o próximo. O Modelo Matemático teve seus erros avaliados e o resultado pode ser visto na Figura 49.

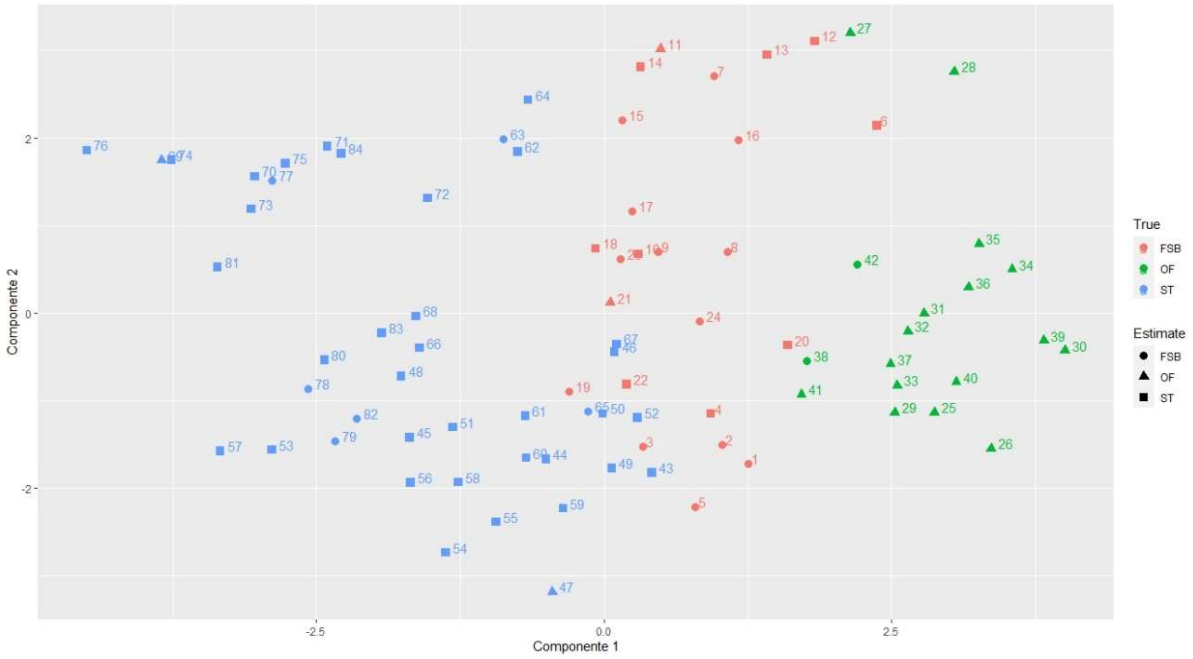
Figura 49 - Gráfico de PCA comparativo entre a estabilidade verdadeira e estimada do Modelo Matemático.



Fonte - R CORE TEAM ,(2016).

Prosseguindo para o Modelo do Q-slope, as mesmas análises foram feitas e os resultados foram expostos na Figura 50.

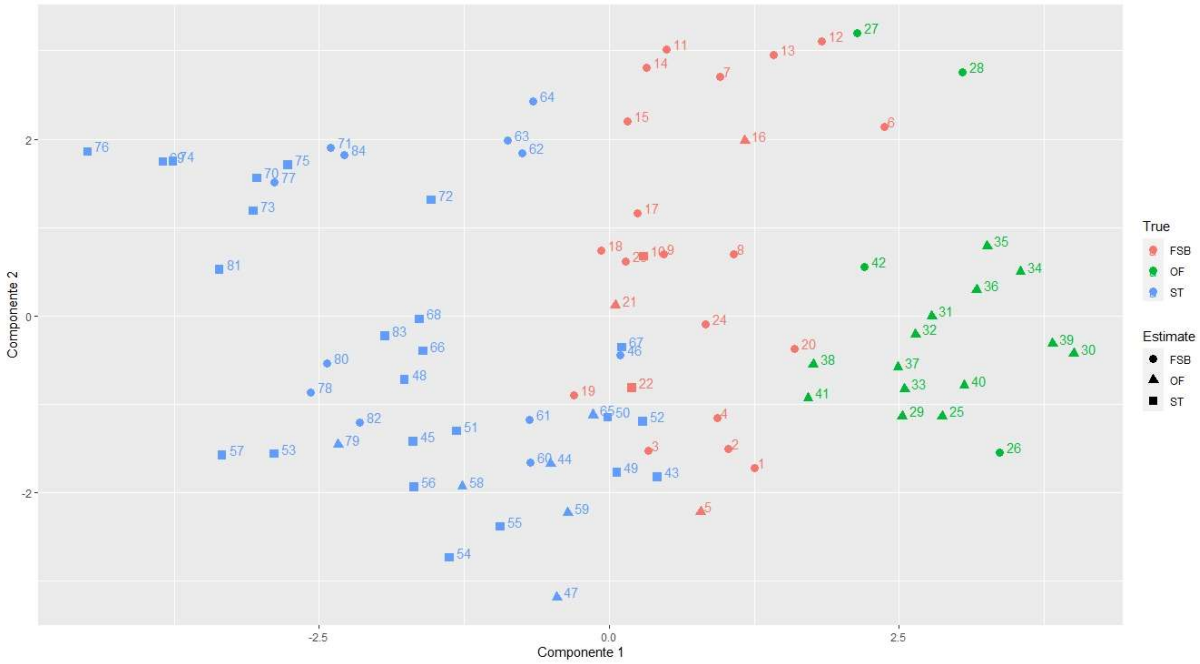
Figura 50 - Gráfico de PCA comparativo entre a estabilidade verdadeira e estimada do Modelo Q-slope.



Fonte - R CORE TEAM ,(2016).

Por fim, o último estimado foi estudado, os erros do MS foram classificados de acordo com os dados da Figura 51.

Figura 51 - Gráfico de PCA comparativo entre a estabilidade verdadeira e estimada do Modelo SANTOS.



Fonte - R CORE TEAM ,(2016).

Com todos estes dados coletados, a relação dos erros para cada modelo foi organizada e apresentada na Tabela 12. Nela é possível ver quais foram as amostras que cada modelo estimou errado, aqueles que estão em vermelho são os classificados como erros perigosos e os demais são considerados erros não perigosos.

Tabela 12 - Relação dos erros de cada modelos de árvore de decisão.

Modelo	Amostras classificadas erradas														Total de erros
Geral	**5**	6	**8**	**17**	21	**22**	**25**	**26**	**42**	62	79				11
Matemático	6	**21**	**22**	**25**	**26**	**40**	**42**	71	77	82	84				11
Q-slope	**4**	**6**	**10**	11	**12**	**13**	**14**	**18**	**20**	21	**22**	**38**	**42**	47	21
	63	65	69	77	78	79	82								
SANTOS	5	**10**	16	21	**22**	**26**	**27**	**28**	**42**	44	46	47	58	59	27
	60	61	62	63	64	65	71	77	78	79	80	82	84		
Repetidos	5	6	21	22	25	26	42	10	16	47	65	71	77	79	16
	82	84													

Além disso, na última linha estão representados quais amostras foram estimadas de forma errada mais de uma vez entre todos os modelos. O número de vezes que cada amostra se repetiu pode ser visto na Tabela 13.

Tabela 13 - Número de vezes que cada amostra foi estimada errada em todos os modelos.

Original							
ID	**Nº de repetições**	**ID**	**Nº de repetições**	**ID**	**Nº de repetições**	**ID**	**Nº de repetições**
1	0	28	1	54	0	81	0
2	0	29	0	55	0	82	3
3	0	30	0	56	0	83	0
4	1	31	0	57	0	84	2
5	2	32	0	58	1		
6	3	33	0	59	1		
7	0	34	0	60	1		
8	1	35	0	61	1		
9	0	36	0	62	2		
10	2	37	0	63	2		
11	1	38	1	64	1		
12	1	39	0	65	2		
13	1	40	1	66	0		
14	1	41	0	67	0		
15	0	42	4	68	0		
16	1	43	0	69	1		
17	1	44	1	70	0		
18	1	45	0	71	2		
19	0	46	1	72	0		
20	1	47	2	73	0		
21	4	48	0	74	0		
22	4	49	0	75	0		
23	0	50	0	76	0		
24	0	51	0	77	3		
25	2	52	0	78	2		
26	3	53	0	79	3		
27	1	54	0	80	1		

Não se espera que tantas amostras sejam classificadas erradas em tantos modelos que utilizam diferentes variáveis para a sua composição, amostras como a 21, 22 e 42 não conseguiram ser estimadas corretamente em nenhum dos 4 modelos.

Com isso uma nova pergunta surge: "O que aconteceria se os dados errados repetidos fossem retirados do modelamento dos modelos?". Para analisar esta situação primeiramente é necessário comparar os erros de todos os modelos. Esta informação pode ser vista na Tabela 14. As formulas usadas para calcular cada porcentagem estão representadas a seguir respectivamente para as duas colunas. Como a predição foi feita com as amostras originais, o número total de amostras neste caso será de 84.

$$\%\ erros\ perigosos = \frac{Erros\ perigosos}{N°\ de\ erros} * 100$$

$$\%\ geral\ dos\ erros = \frac{N°\ de\ erros}{N°\ total\ de\ amostras} * 100$$

Tabela 14 - Porcentagem dos erros para cada modelo criado.

Modelo	COM ERROS REPETIDOS				
	Nº de erros	**Erros aceitáveis**	**Erros perigosos**	**Porcentagem de erros perigosos**	**Porcentagem geral dos erros**
Geral	11.00	4.00	7.00	63.64%	13.10%
Matemático	11.00	5.00	6.00	54.55%	13.10%
Qslope	21.00	10.00	11.00	52.38%	25.00%
SANTOS	27.00	21.00	6.00	22.22%	32.14%

Dos 4 modelos, temos dois deles com resultados mais discrepantes, o MM e o MS. Enquanto o primeiro tem uma menor taxa de erros gerais, porém uma elevada taxa de erros perigosos, o segundo está na situação oposta. Por causa disso eles foram selecionados para o modelamento com os novos dados.

Neste momento então, estes dois modelos foram refeitos com o novo banco de dados sem as amostras que foram retiradas de acordo com o proposto anteriormente. Estes dados foram novamente balanceados seguindo os mesmos princípios explicados anteriormente no Tópico 3.4.1 e respeitando as características originais das amostras. Os resultados de cada modelagem estão apresentados a seguir.

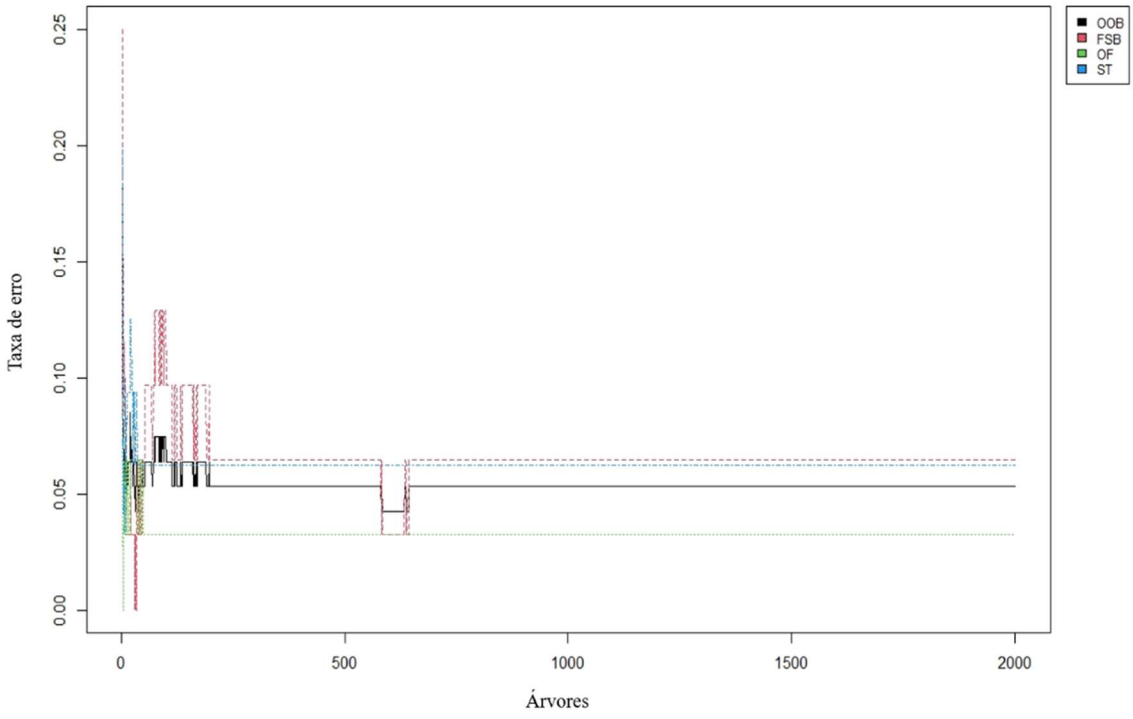
4.7. Modelo Matemático sem erros (MMS)

4.7.1. SELEÇÃO DAS VARIÁVEIS POR *RANDOM FOREST*

Como é necessário repetir todo o processo realizado para o desenvolvimento dos dois modelos, é preciso que a seleção de variáveis pelo *Random Forest* no MM seja revisitado. Para a seleção das variáveis o RF foi realizado com os dados balanceados, 3000 árvores foram criadas com 2000 interações entre elas. Além disso, é importante salientar que para este treino 80% e os demais 20% foram utilizados para o treino do modelo para a sua verificação.

Os erros podem ser vistos na Figura 52, sendo que após aproximadamente 700 árvores testadas, o erro relativo de cada fator inclusive o erro associado ao OOB se estabilizam. Desta forma as importâncias e Pureza de Gini para cada variável podem ser usadas para determinar quais são as mais determinantes para o modelo RF. Como o erro relativo do modelo se estabilizou com muito menos árvores, é possível dizer que a retirada das amostras contribuiu positivamente para a modelagem.

Figura 52 - Gráfico dos erros relativos para cada fator de estabilidade e OOB em função do número de árvores criadas.



Fonte - R CORE TEAM ,(2016).

De acordo com a Figura 53 é possível visualizar a importância e o Índice de Gini de cada variável. Variáveis como a V18, V4 e V2 neste novo RF obtiveram importâncias muito elevadas neste novo banco de dados.

Figura 53 - Importância e Índice de Gini para cada variável do banco de dados.

	MeanDecreaseAccuracy	MeanDecreaseGini
Rock_type_V1	0.024281761	2.1555418
Intact_Rock_Strength_V2	0.162510744	14.0383119
RQD_V3	0.014623047	1.1494327
Weathering_V4	0.177913958	14.7136238
Tectonic_Regime_V5	0.013803521	1.1086896
Groundwater_V6	0.013141919	1.5526918
Number_of_sets_V7	0.016958155	1.5463192
Persistence_V8	0.006663062	0.7609436
Spacing_V9	0.002121086	0.2793471
Orientation_V10	0.014666774	1.4302100
Aperture_V11	0.006433057	0.7704523
Roughness_JRC_macro_V12	0.002449188	0.2559635
Filling_V13	0.014459343	1.7566747
Overall_angle_degrees_V14	0.041742643	2.9917527
Overall_Height_meters_V15	0.005301421	1.1562360
Blasting_Method_V16	0.019464396	2.2445981
Precipitation_mmperyear_V17	0.020014359	1.8030658
Previous_Instability_V18	0.149964824	12.2799932

Fonte - R CORE TEAM ,(2016).

As variáveis selecionadas pelo RF estão apresentadas na Tabela 15. Antes de dar início à criação do MMS, é preciso validar o modelo com a matriz de confusão e o *bootstrap* novamente.

Tabela 15 - Novas variáveis selecionadas pelo RF.

Modelo Matemático sem erros (MMS)
Resistência da rocha intacta (V2)
Alteração (V4)
Número de famílias (V7)
Ângulo geral (V14)
Método de desmonte (V16)
Instabilidade prévia (V18)

4.7.2. VALIDAÇÃO DO *RANDOM FOREST*

Primeiramente o teste do modelo foi feito com a predição do estimador utilizando os 20% dos dados restantes como pode ser visto na Figura 54. Novamente a matriz de confusão resultou numa acurácia de 100%. Para forçar mais as capacidades do modelo o método de *bootstrap* foi usado para determinar o erro esperado da predição dos dados.

Figura 54 - Matriz de confusão para o teste do novo modelo de RF.

```
Confusion Matrix and Statistics

          Reference
Prediction FSB OF ST
       FSB   6  0  0
       OF    0  6  0
       ST    0  0  6

Overall Statistics

               Accuracy : 1
                 95% CI : (0.8147, 1)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : 2.581e-09

                  Kappa : 1

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: FSB Class: OF Class: ST
Sensitivity              1.0000    1.0000    1.0000
Specificity              1.0000    1.0000    1.0000
Pos Pred Value           1.0000    1.0000    1.0000
Neg Pred Value           1.0000    1.0000    1.0000
Prevalence               0.3333    0.3333    0.3333
Detection Rate           0.3333    0.3333    0.3333
Detection Prevalence     0.3333    0.3333    0.3333
Balanced Accuracy        1.0000    1.0000    1.0000
```

Fonte - R CORE TEAM ,(2016).

Os mesmos parâmetros usados no primeiro *bootstrap* foram usados neste, 3000 árvores com 2000 interações, porém desta vez, 94 amostras foram criadas para determinar a probabilidade de classificação em cada fator de estabilidade.

Na Figura 55 estão apresentados os dados obtidos do método com 100 interações de *bootstrap.* O erro resultante foi de 4,85%, valor este inferior ao primeiro teste feito, um outro ótimo indicio da melhora do modelamento sem as amostras problemáticas retiradas do banco de dados.

Figura 55 - Resultados estatísticos do novo *bootstrap*.

Resultados do *Bootstrap*

Estimativa do erro da predição do Bootstrap para 100 interações:

```
0.04853001
```

Número de variáveis em cada floresta no Bootstrap:

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   2.00    3.00    4.00    4.94    5.00   18.00
```

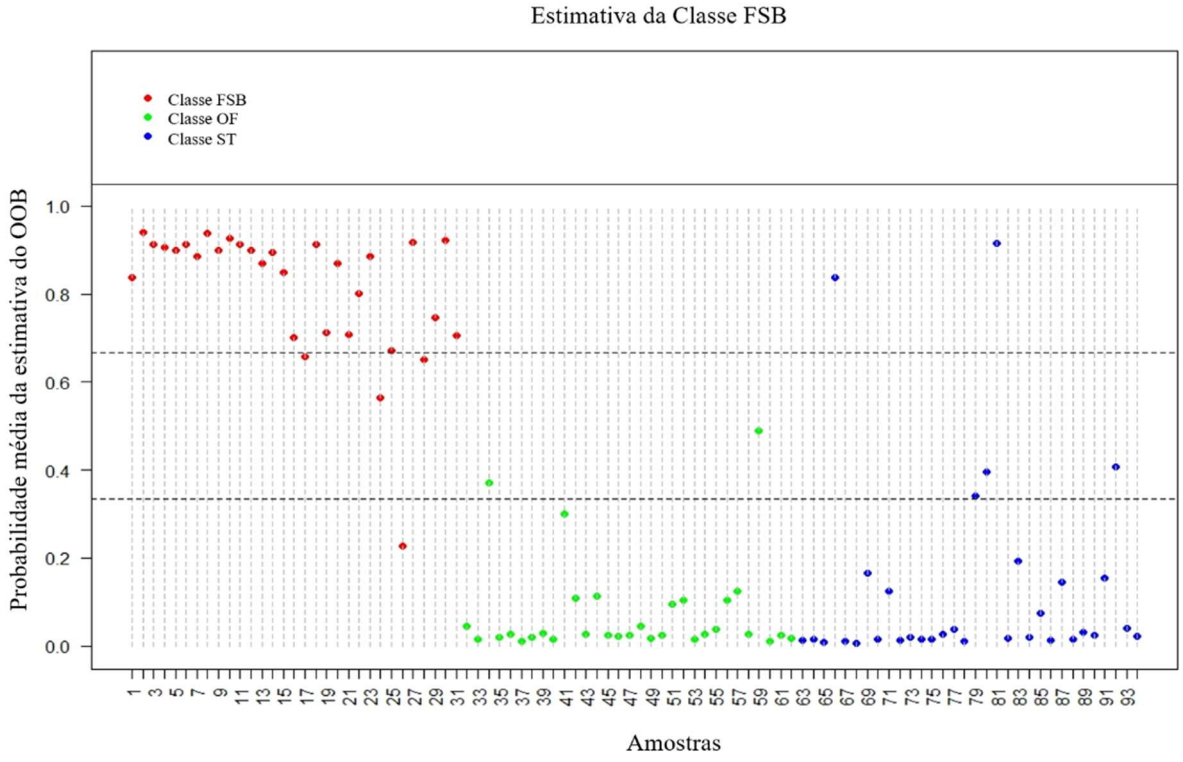
Sobreposição das florestas do Bootstrap com todos os dados:

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.5455  0.7004  0.7071  0.7348  0.8165  1.0000
```

Fonte - R CORE TEAM ,(2016).

Para averiguar a capacidade do modelo em estimar corretamente cada nova amostra, a estimativa foi feita para a probabilidade de classificar como FSB e isto está apresentado na Figura 56.

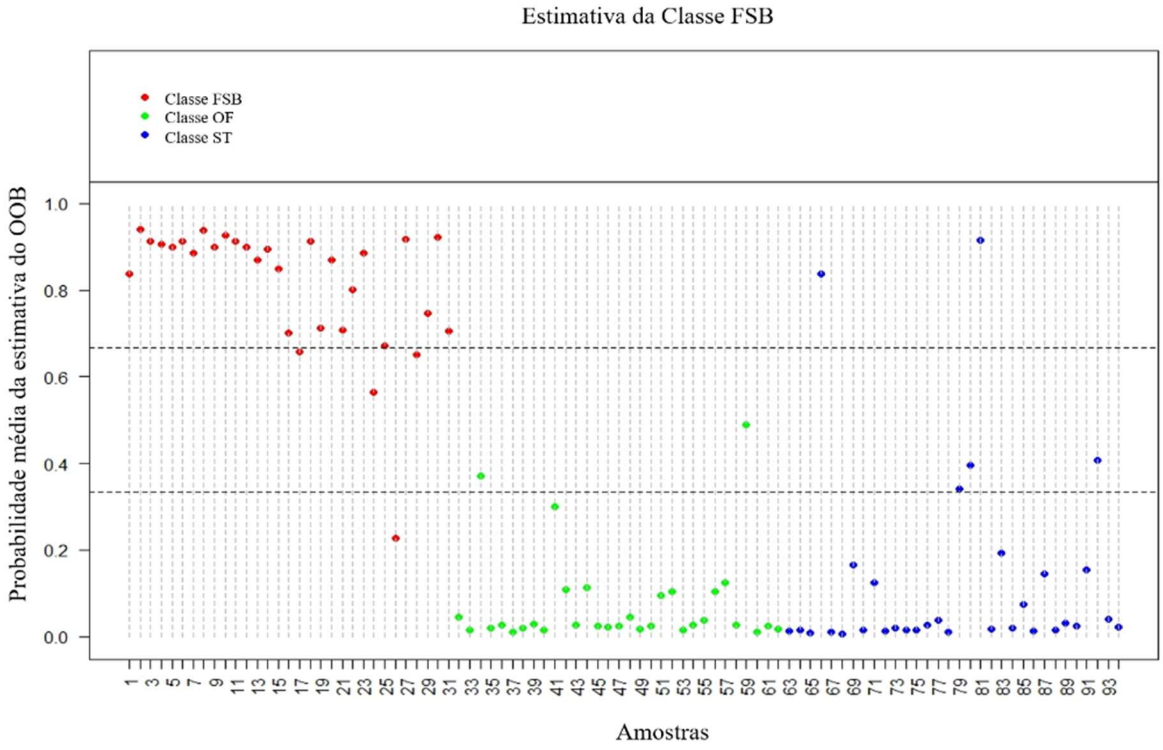
Figura 56 Gráfico da probabilidade de classificação das amostras do novo *bootstrap* para a classe FSB.



Fonte - R CORE TEAM ,(2016).

Novamente a classe FSB é visivelmente uma zona de transição entre as classes ST e OF tendo amostras das 3 classificações com probabilidades diversas entre eles. No entanto, isto ainda é menos acentuado se comparado com os resultados anteriores. A próxima classe é para a probabilidade de classificação como OF, o que pode ser visto na Figura 57.

Figura 57 - Gráfico da probabilidade de classificação das amostras do novo *bootstrap* para a classe OF.

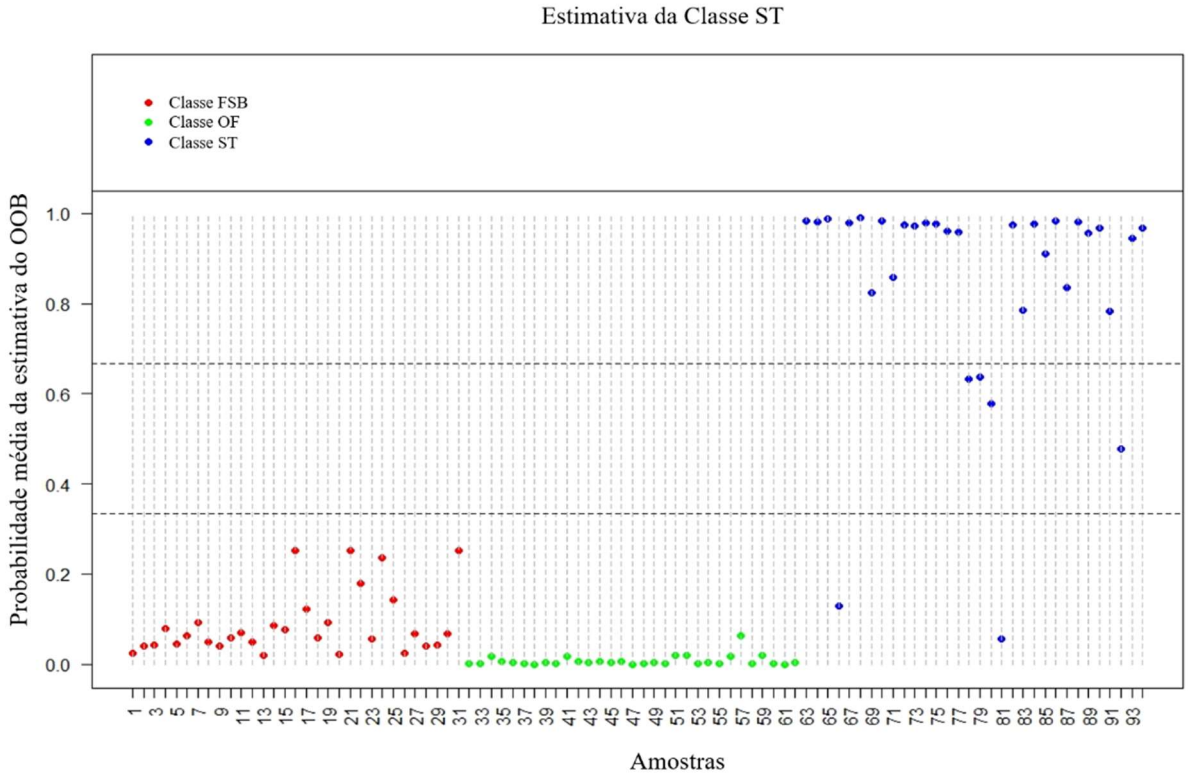


Fonte - R CORE TEAM ,(2016).

Novamente, como ocorreu no primeiro *bootstrap*, há dados na zona de transição de OF/FSB, isso significa que mesmo com a retirada de amostras, o banco de dados não foi descaracterizado, outro grande indício dos malefícios que estes dados estavam causando para o modelo final.

Por fim, a última classe foi testada e os resultados podem ser vistos na Figura 58. Mais uma vez é clara a presença da zona de transição dos dados entre FSB/ST, e como visto anteriormente, esta área é muito mais presente entre estas classes.

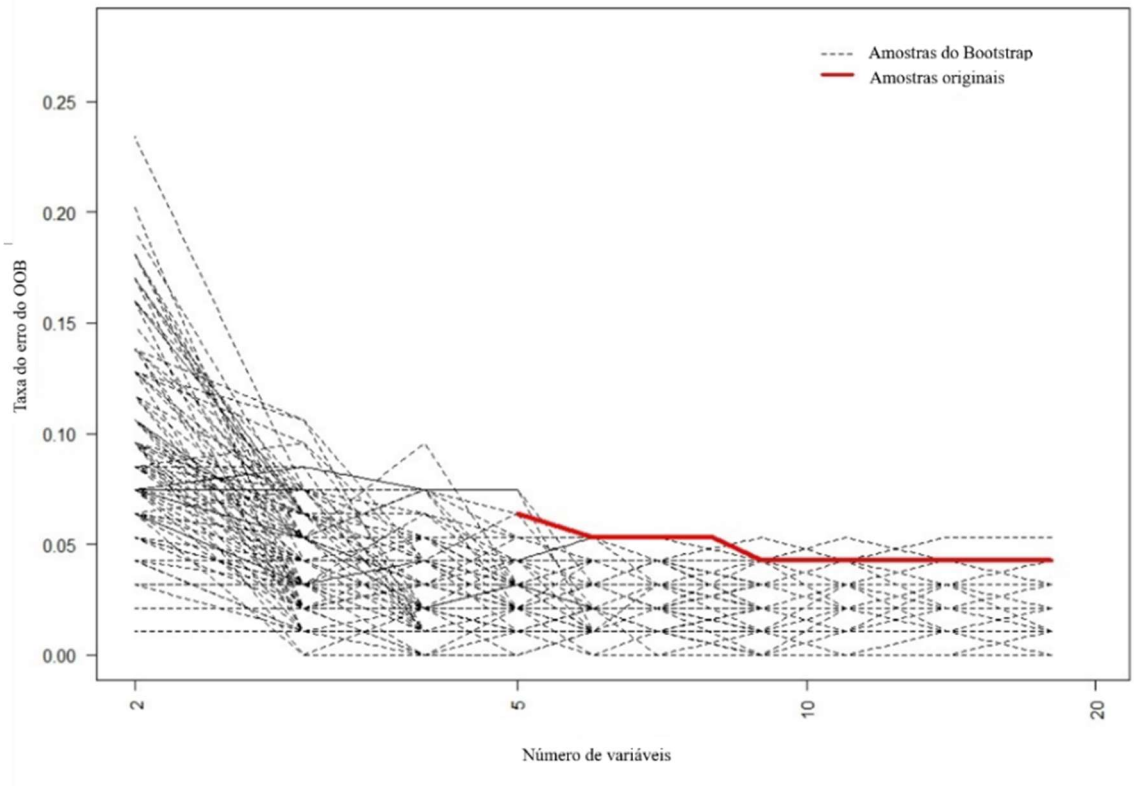
Figura 58 - Gráfico da probabilidade de classificação das amostras do novo *bootstrap* para a classe ST.



Fonte - R CORE TEAM ,(2016).

O erro relativo das 100 interações de *bootstrap* podem ser vistas na Figura 59 e comparadas com o erro relativo dos dados originais usados no modelo do RF. Desta vez, a estabilização dos erros das interações ficou abaixo do erro original, já que as linhas tracejadas ficaram abaixo da linha vermelha no gráfico.

Figura 59 - Erro relativo das interações do bootstrap comprado com as amostras originais em função do número de variáveis.



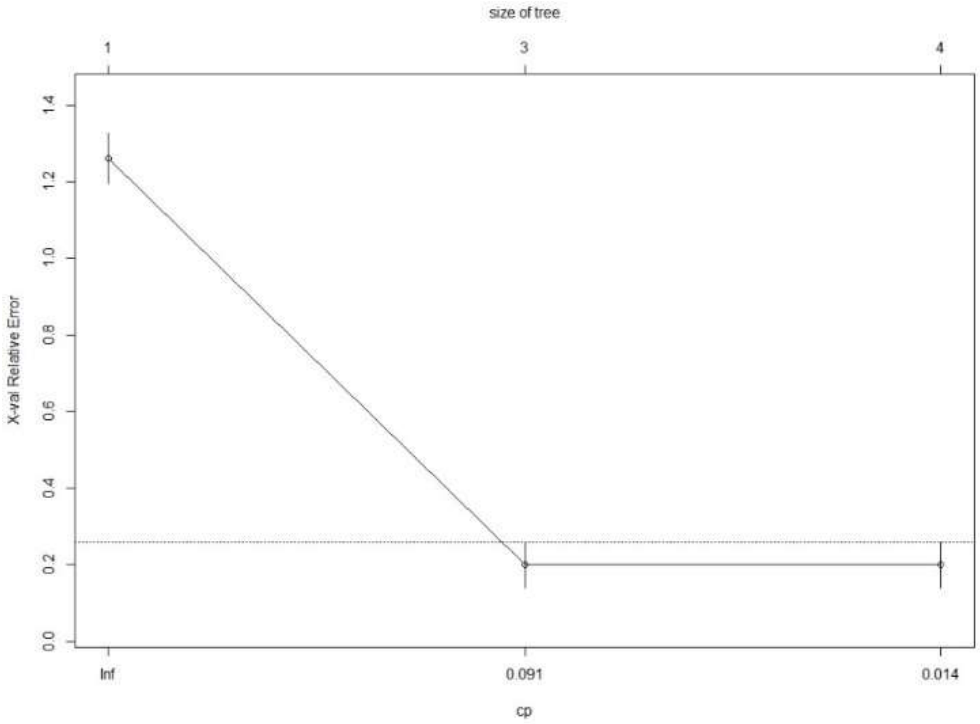
Fonte - R CORE TEAM ,(2016).

Após esta etapa, foi possível comprovar a eficácia da seleção das variáveis pelo RF e por isso foi possível desenvolver a nova árvore para o MMS.

4.7.3. TREINO DO MODELO MMS

O treino do modelo foi feito seguindo as mesmas regras de partição anteriores, porém desta vez foram usados os dados contendo apenas as variáveis selecionadas pelo novo *Random Forest* feito anteriormente. O CP foi determinado de acordo com os dados da Figura 60, o que determinou o tamanho ótimo da árvore nestas condições.

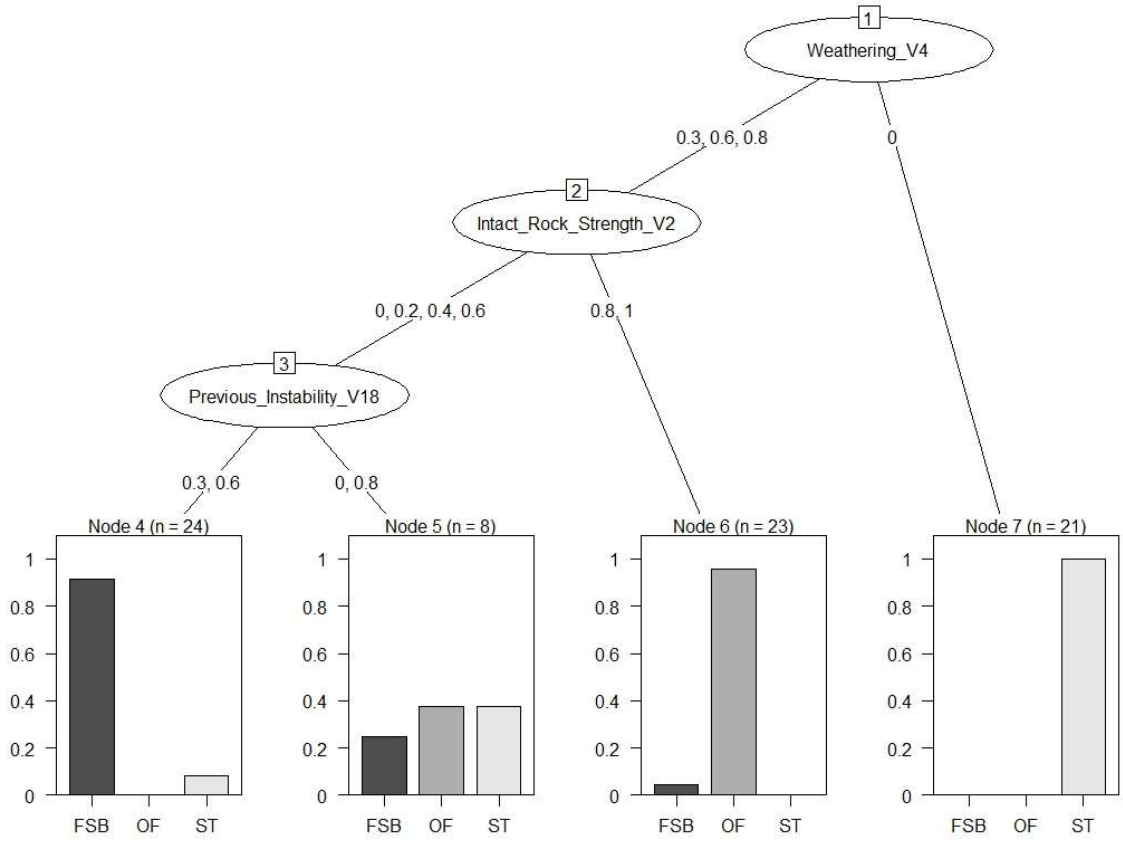
Figura 60 - Gráfico do CP para o MMS.



Fonte - R CORE TEAM ,(2016).

O MMS pode ser visto na Figura 61 e já é possível analisar nesta árvore que não há dados errados neste modelo que podem ser classificados como erros perigosos. Outra informação obtida nesta árvore é que apenas as variáveis V2, V4 e V18 foram utilizadas após a poda do modelo, isso está associado com a grande importância dada a estas variáveis, o que pode ser visto no Tópico 4.3.1 na seleção de variáveis do RF.

Figura 61 Árvore de decisão do MMS.



Fonte - R CORE TEAM ,(2016).

Uma característica que esse novo modelo apresentou foi seu conservadorismo. Como não houve amostras com uma estabilidade estimada maior que a real, ou seja, não houve erros perigosos, este até agora seria o melhor modelo obtido entre todos. Além disso, as variáveis selecionadas são facilmente obtidas com testes de campo, o que facilitaria sua utilização em operações de mina para validações de estabilidade prévias por qualquer pessoa.

4.7.4. TESTE DO MODELO MMS

Para testar o modelo, o conjunto de teste de 20% e o banco de dados original sem amostras erradas foram utilizados para criar as matrizes de confusão. Na Figura 62, na matriz com amostra de teste de 20%, foi notável o aumento da acurácia do MMS se comparado com os demais modelos. O mesmo acontece com a matriz de confusão para todos os dados que teve também a maior acurácia entres as demais árvores.

Figura 62 - Matriz de confusão e dados estatísticos do novo teste com 20% dos dados balanceados como teste.

```
Confusion Matrix and Statistics

          predict_unseen7
           FSB OF ST
  FSB    5  1  0
  OF     0  6  0
  ST     0  0  6

Overall Statistics

                Accuracy : 0.9444
                  95% CI : (0.7271, 0.9986)
     No Information Rate : 0.3889
     P-Value [Acc > NIR] : 1.212e-06

                   Kappa : 0.9167

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: FSB Class: OF Class: ST
Sensitivity              1.0000    0.8571    1.0000
Specificity              0.9231    1.0000    1.0000
Pos Pred Value           0.8333    1.0000    1.0000
Neg Pred Value           1.0000    0.9167    1.0000
Prevalence               0.2778    0.3889    0.3333
Detection Rate           0.2778    0.3333    0.3333
Detection Prevalence     0.3333    0.3333    0.3333
Balanced Accuracy        0.9615    0.9286    1.0000
```

Fonte - R CORE TEAM ,(2016).

O mesmo acontece com a matriz de confusão para todos os dados apresentados na Figura 63, que teve também a maior acurácia entres as demais árvores. Claramente a remoção das amostras diminui o erro do modelo, melhorando sua interpretação das amostras e principalmente reduzindo drasticamente os erros perigosos da estimativa. Dando continuidade, o Modelo SANTOS sem erros (MSS) foi desenvolvido para ver se o mesmo efeito causado pela remoção das amostras problemáticas contribui positivamente para a árvore final.

Figura 63 - Matriz de confusão e dados estatísticos do novo teste com 100% dos dados originais como teste.

```
Confusion Matrix and Statistics

          predict_unseen7
           FSB OF ST
  FSB  17  2  0
  OF    0 15  0
  ST    2  3 27

Overall Statistics

               Accuracy : 0.8939
                 95% CI : (0.7936, 0.9563)
    No Information Rate : 0.4091
    P-Value [Acc > NIR] : 2.664e-16

                  Kappa : 0.8368

 Mcnemar's Test P-Value : 0.0719

Statistics by Class:

                     Class: FSB Class: OF Class: ST
Sensitivity              0.8947    0.7500    1.0000
Specificity              0.9574    1.0000    0.8718
Pos Pred Value           0.8947    1.0000    0.8438
Neg Pred Value           0.9574    0.9020    1.0000
Prevalence               0.2879    0.3030    0.4091
Detection Rate           0.2576    0.2273    0.4091
Detection Prevalence     0.2879    0.2273    0.4848
Balanced Accuracy        0.9261    0.8750    0.9359
```

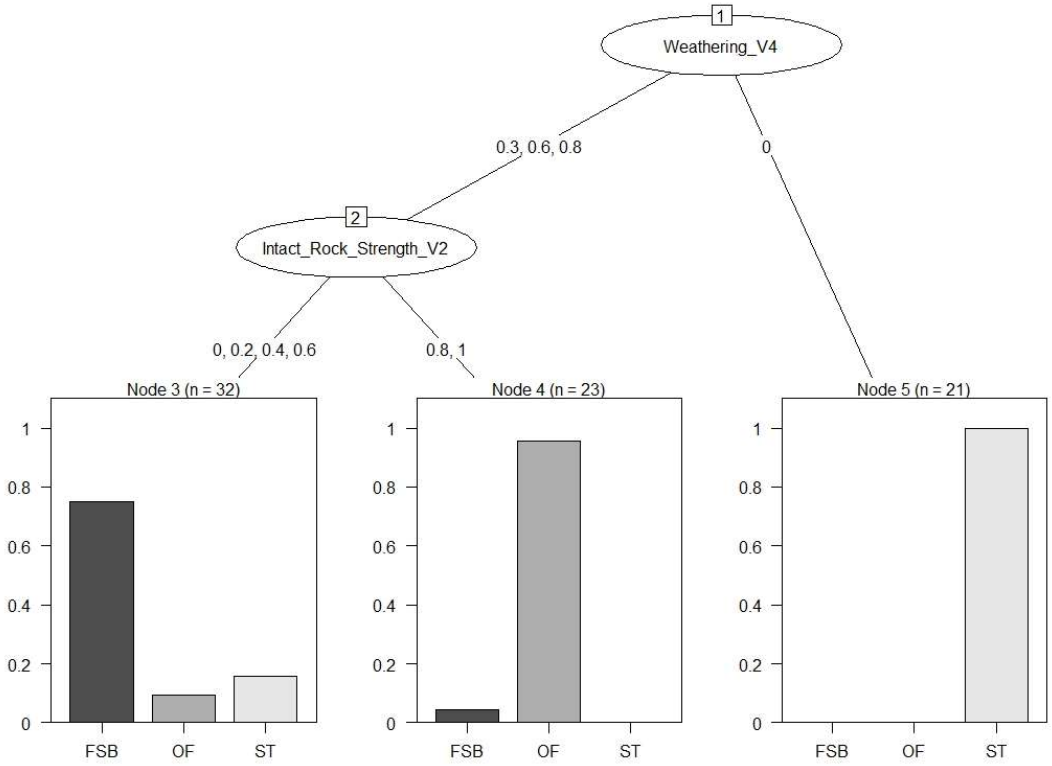
Fonte - R CORE TEAM ,(2016).

4.8.Modelos SANTOS sem erros (MSS)

4.8.1. TREINO DO MODELO MSS

A seleção das variáveis para este modelo permaneceu igual, pois como estas foram selecionadas por Santos *et.al* (2020) usando métodos diferentes, não havia motivos para reinterpretar os seus resultados. Sendo assim, seguindo a mesma partição dos dados, o conjunto de treino foi criado e o modelo desenvolvido. O resultado pode ser visto na Figura 64.

Figura 64 Árvore para o MSS.



Fonte - R CORE TEAM ,(2016).

Desta vez, apenas as variáveis V2 e V4 foram utilizadas após a poda da árvore. Muito provavelmente a variável V18 teria entrado caso Santos *et.al* (2020) a tivessem selecionado em seu trabalho por causa da importância que esta tem no modelo como visto anteriormente, resultando em duas árvores iguais para MMS e MSS.

4.8.2. TESTE DO MODELO MSS

A falta da variável V18 nos dados utilizados para o desenvolvimento do modelo fez com que surgisse erros perigosos neste modelo como pode ser visto nas Figura 65. A acurácia de MSS e MMS foram iguais, porém neste caso o erro foi do tipo perigoso já que uma amostra OF foi classificada como FSB.

Figura 65 - Matriz de confusão e dados estatísticos do novo teste com 20% dos dados balanceados como teste.

```
Confusion Matrix and Statistics

     predict_unseen8
      FSB OF ST
  FSB   6  0  0
  OF    1  5  0
  ST    0  0  6

Overall Statistics

               Accuracy : 0.9444
                 95% CI : (0.7271, 0.9986)
    No Information Rate : 0.3889
    P-Value [Acc > NIR] : 1.212e-06

                  Kappa : 0.9167

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: FSB Class: OF Class: ST
Sensitivity              0.8571    1.0000    1.0000
Specificity              1.0000    0.9231    1.0000
Pos Pred Value           1.0000    0.8333    1.0000
Neg Pred Value           0.9167    1.0000    1.0000
Prevalence               0.3889    0.2778    0.3333
Detection Rate           0.3333    0.2778    0.3333
Detection Prevalence     0.3333    0.3333    0.3333
Balanced Accuracy        0.9286    0.9615    1.0000
```

Fonte - R CORE TEAM ,(2016).

Já para o teste com 100% do banco de dados original apresentado na Figura 66, a acurácia foi de 87,88%, inferior se comparado com o MMS, porém maior que os demais modelos, e além disso houve 2 erros perigosos neste conjunto de dados testado.

Figura 66 - Matriz de confusão e dados estatísticos do novo teste com 100% dos dados originais como teste.

```
Confusion Matrix and Statistics

          predict_unseen8
           FSB OF ST
      FSB  18  1  0
      OF    2 13  0
      ST    5  0 27

Overall Statistics

                    Accuracy : 0.8788
                      95% CI : (0.7751, 0.9462)
        No Information Rate : 0.4091
        P-Value [Acc > NIR] : 2.878e-15

                       Kappa : 0.8119

     Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: FSB Class: OF Class: ST
Sensitivity              0.7200    0.9286    1.0000
Specificity              0.9756    0.9615    0.8718
Pos Pred Value           0.9474    0.8667    0.8438
Neg Pred Value           0.8511    0.9804    1.0000
Prevalence               0.3788    0.2121    0.4091
Detection Rate           0.2727    0.1970    0.4091
Detection Prevalence     0.2879    0.2273    0.4848
Balanced Accuracy        0.8478    0.9451    0.9359
```

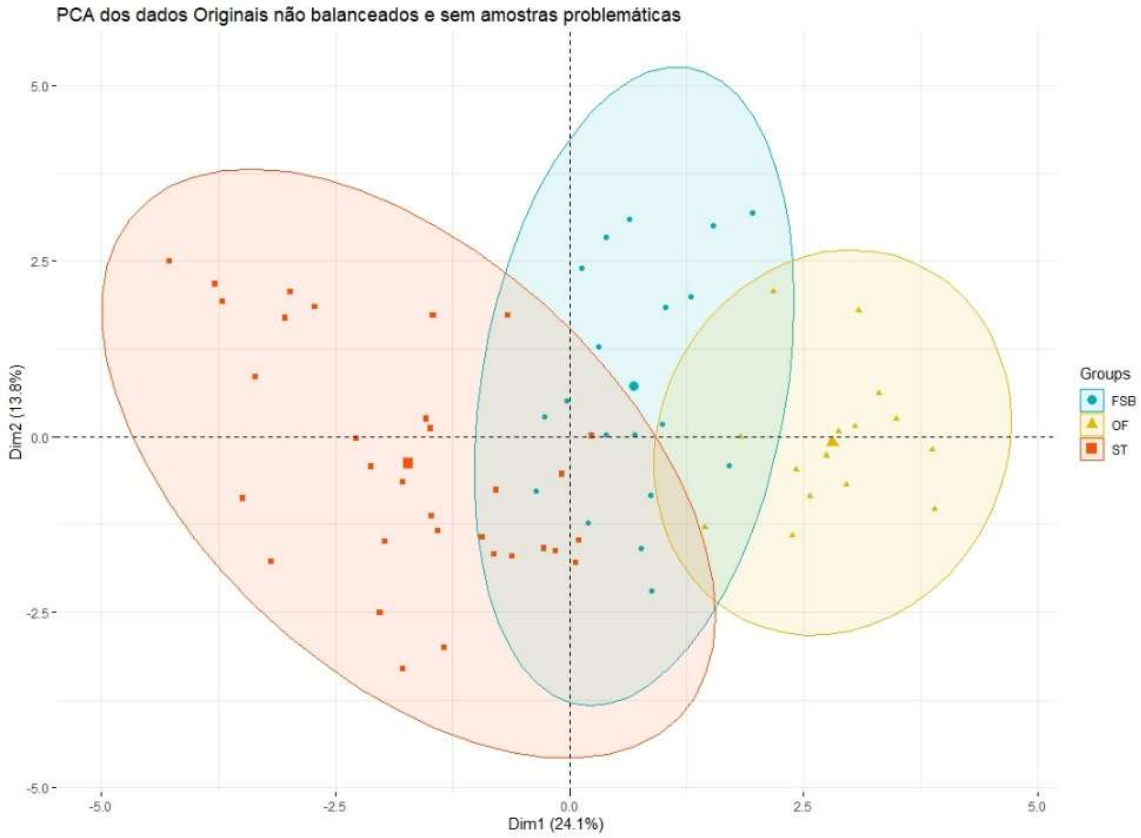
Fonte - R CORE TEAM ,(2016).

Mesmo tendo um resultado abaixo do Modelo Matemático sem erros, o MSS teve uma grande melhora na redução de erros gerais do modelo ao mesmo tempo que manteve uma baixa taxa de erros perigosos. Mais uma vez comprovando o efeito negativo das amostras retiradas destes dois novos modelos. Para visualizar espacialmente as amostras, o PCA deste novo banco de dados foi feito e seus resultados dispostos a seguir.

4.9. Análise dos Componentes Principais para as amostras sem erros

Primeiramente o PCA dos dados foi feito e demonstrado na Figura 67 e como já era esperado, a zona de transição não foi comprometida com a remoção das amostras que estavam comprometendo a eficácia da modelagem das árvores.

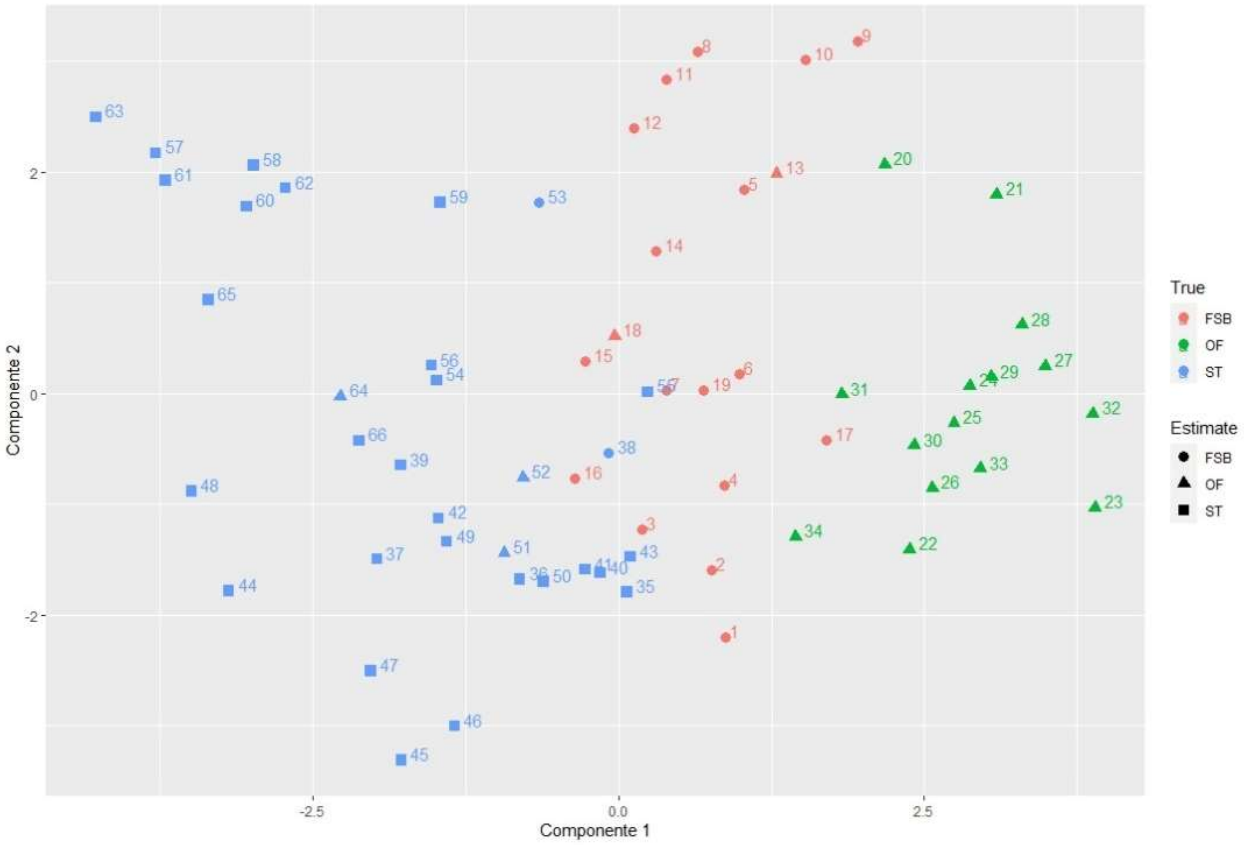
Figura 67 - Gráfico PCA para os dados não balanceados sem amostras problemáticas.



Fonte - R CORE TEAM ,(2016).

Para novamente identificar quais amostras foram estimadas erradas, o PCA comparativo dos dois modelos foi criado. O primeiro para o MMS está representado na Figura 68.

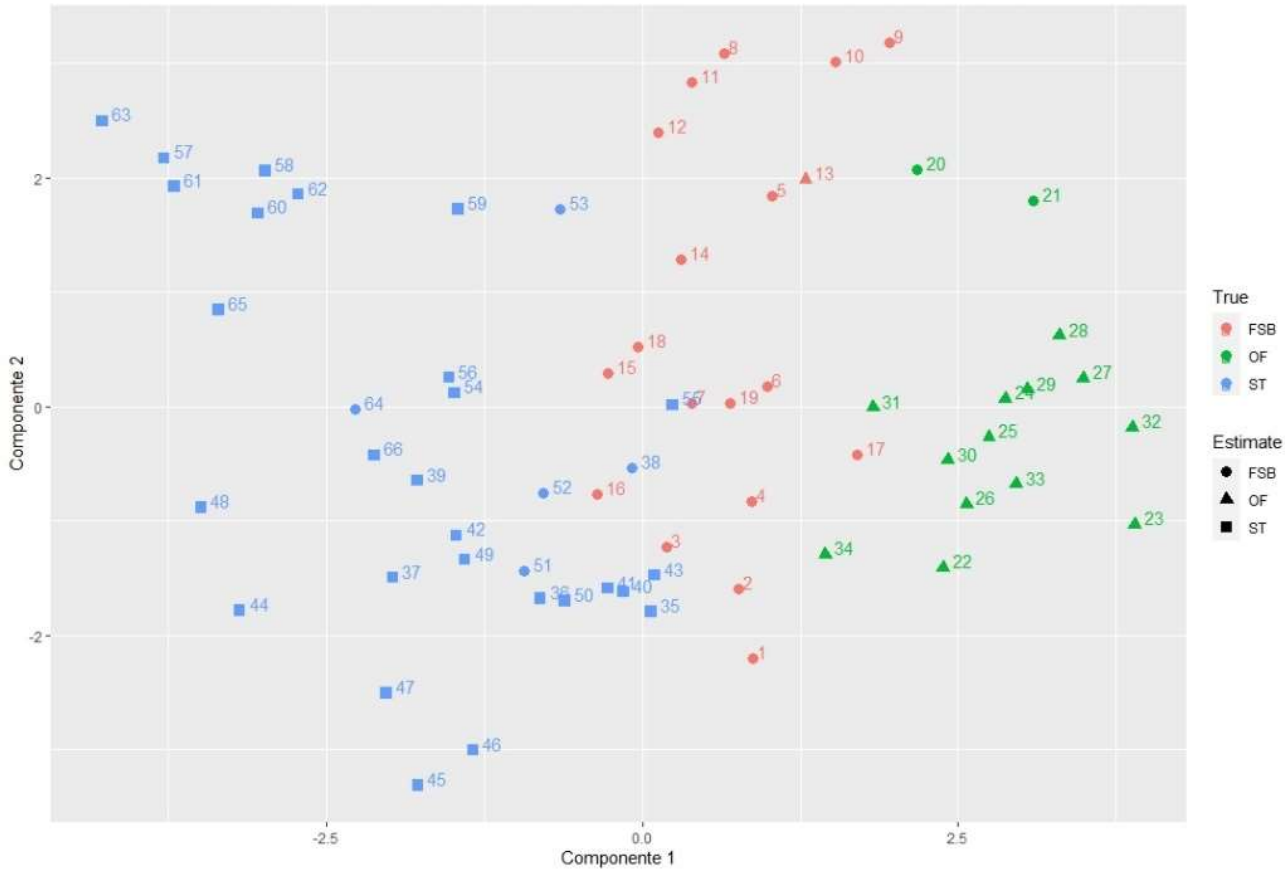
Figura 68 - PCA comparativo entre a estimativa do MMS e a classificações reais sem as amostras problemáticas.



Fonte - R CORE TEAM ,(2016).

O mesmo pode ser visto para o MSS na Figura 69, no entanto é importante comentar que as identificações dos PCAs apresentados não correspondem à mesma identificação do banco de dados original com todas as amostras, por isso que a relação da Tabela 16 foi criada para identificar corretamente as amostras.

Figura 69 - PCA comparativo entre a estimativa do MSS e a classificações reais sem as amostras problemáticas.



Fonte - R CORE TEAM ,(2016).

Tabela 16 - Relação das identificações das amostras originais com as identificações do banco de dados sema amostras problemáticas.

ID novo banco de dados	ID Original	ID novo banco de dados	ID Original	ID novo banco de dados	ID Original	ID novo banco de dados	ID Original	ID novo banco de dados	ID Original
1	1	15	18	29	36	43	52	57	69
2	2	16	19	30	37	44	53	58	70
3	3	17	20	31	38	45	54	59	72
4	4	18	23	32	39	46	55	60	73
5	7	19	24	33	40	47	56	61	74
6	8	20	27	34	41	48	57	62	75
7	9	21	28	35	43	49	58	63	76
8	11	22	29	36	44	50	59	64	80
9	12	23	30	37	45	51	60	65	81
10	13	24	31	38	46	52	61	66	83
11	14	25	32	39	48	53	64		
12	15	26	33	40	49	54	66		
13	16	27	34	41	50	55	67		
14	17	28	35	42	51	56	68		

Com essa relação em mãos, foi possível determinar as amostras estimadas de forma errada e organizá-las na Tabela 17. Uma grande redução no número de erros ocorreu com a retirada das amostras problemáticas e como os dois modelos finais obtidos são muito parecidos não se esperava menos que a presença de muitas amostras preditas erradas repetidas nos dois modelos.

Tabela 17 - Relação das amostras classificadas erradas, na cor vermelha os erros perigosos e na cor preta os erros não perigosos.

Banco de dados	Modelo	Amostras classificadas erradas	Total de erros
Sem erros problemáticos	**Matemático**	16 23 46 60 61 64 80	7
	SANTOS	16 **27** **28** 46 60 61 64 80	8
	Repetidos	16 46 60 61 64 80	6

Com estas informações é possível comparar todos os modelos criados para determinar qual deles teve a melhor resposta para diminuir principalmente os erros perigosos. Esta análise pode ser vista na Tabela 18.

Tabela 18 - Comparação dos Modelos desenvolvidos.

Modelo	SEM ERROS REPETIDOS				
	Nº de erros	Erros aceitáveis	Erros perigosos	Porcentagem de erros perigosos	Porcentagem geral dos erros
Matemático	7.00	7.00	0.00	0.00%	10.61%
SANTOS	8.00	6.00	2.00	25.00%	12.12%

Os modelos Matemático e SANTOS sem a presença das amostras problemáticas tiveram o melhor desempenho preditivo dentre os 6 modelos desenvolvidos, sendo que o MMS teve um desempenho excepcional pois reduziu ao máximo o número de erros perigosos em sua estimação.

5. CONCLUSÕES

O desenvolvimento de novos métodos preditivos de estabilidade de talude utilizando modelos de aprendizado de máquina tem sido amplamente utilizados para criar modelos mais confiáveis e fáceis de utilizar em empreendimentos de mineração a céu aberto. Ao utilizar modelos como as árvores de decisão é possível criar uma forma direta e simples de interpretar a estabilidade de um talude.

Esta pesquisa apresenta como objetivo principal a proposta de criação de modelos de fácil implementação e com uma eficiência satisfatória em campo para determinação da estabilidade de talude. Esta pesquisa, a partir de um banco de dados com 84 amostras coletadas em taludes no mundo todo, criou 6 modelos de árvore de decisão utilizando diferentes variáveis a partir de interpretações matemáticas e da literatura de diferentes parâmetros geotécnicos e espaciais de taludes.

Após a realização de todas as etapas do desenvolvimento foi possível determinar que os melhores modelos foram o Modelo Matemático e o Modelo SANTOS utilizando o banco de dados sem as amostras interpretadas como problemáticas. Além de terem uma elevada acurácia principalmente para um modelo simples como as árvores de decisão, estes também obtiveram uma baixa incidência de erros perigosos o que eleva mais ainda o potencial de utilização deles para estimativas de estabilidade de taludes.

Com o uso destes modelos desenvolvidos, é possível determinar as condições de estabilidade de taludes de minas a céu aberto em escala industrial, podendo variar os diferentes parâmetros geotécnicos para avaliar o resultado da interação das variáveis. Principalmente no controle da altura e ângulo geral dos taludes, fundamentais para determinar o andamento das atividades minerais em qualquer empreendimento mineral a céu aberto. Desta forma, pode-se assim otimizar a explotação e aproveitamento da reserva, mantendo a operacionalidade da cava e maximizando a segurança das operações.

Além disso, por utilizarem variáveis facilmente obtidas em campo, estes modelos podem ser utilizados por usuários em geral. Os modelos aqui apresentados desconstroem as “Black-boxes” presentes nos modelos de inteligência artificial que são limitadores da utilização de um público em geral, facilitando as tomadas de decisão em empreendimentos que envolvam estes tipos de problemas. Por fim, como proposta para trabalhos futuros, têm-se a possibilidade de adição de novos conjuntos de dados com o objetivo de refinar a predição dos modelos.

APÉNDICE

```
##################################RANDOM FOREST#################################


library(magrittr)
library(dplyr)
library(tidyr)
library(readxl)
library(ecodist)
library(summarytools)
library(party)
library(ggplot2)


nag <- read_excel("C:/Users/SAMSUNG/Desktop/r/tcc/nag.xlsx",
                  sheet = "semerrobalan")
nag[sapply(nag, is.numeric)] <- lapply(nag[sapply(nag, is.numeric)],
                               as.factor)
nag$Stability_status <- factor(nag$Stability_status)

dados <- nag[,-19]

#############################################################################################

{set.seed(2356547)
partitionrule <- createDataPartition(nag$Stability_status,p=0.8, list=FALSE)

trainingset <- nag[partitionrule,]
testingset <- nag[-partitionrule,]
}

#############################################################################################
############### RANDOM FOREST & OOB COM O RFSRC ##############################
#############################################################################################

library(varSelRF)
set.seed(9999)
rf.vs2 <- varSelRF(dados,nag$Stability_status, ntree = 3000, ntreeIterat = 2000,
                   vars.drop.frac = 0.01, whole.range = FALSE,
                   keep.forest = TRUE)

predict(rf.vs2$rf.model,
      newdata = subset(testingset, select = rf.vs2$selected.vars))
predict(rf.vs2$rf.model,
      newdata = subset(testingset, select = rf.vs2$selected.vars),
      type = "prob")

rpartpred2<- predict(rf.vs2$rf.model, newdata=testingset)
confusionMatrix(data=rpartpred2,testingset$Stability_status)
```

```
{
 layout(matrix(c(1,2),nrow=1),width=c(8,1))
 par(mar=c(5,5,4,0))
 plot(rf.vs2$rf.model, xlog=TRUE, ylog= TRUE)
 par(mar=c(5,0,4,2))
 plot(c(0,1),type="n", axes=F, xlab="", ylab="")
 legend("top", colnames(rf.vs2$rf.model$err.rate),col=1:4,cex=0.8,fill=1:4)
}

rf.vs2$selected.model
rf.vs2$selected.vars
rf.vs2$initialImportances
rf.vs2$selec.history

################################################################################

NAGA.rf2<-randomForest(dados,nag$Stability_status,
                       ntree = 3000,
                       mtry=7 ,
                       ntreeIterat = 2000,
                       importance=TRUE,
                       subset = train,
                       vars.drop.frac = 0.01)
NAGA.rf2

rf.rvi <- randomVarImpsRF(dados,nag$Stability_status,
                          NAGA.rf2,
                          numrandom = 20,
                          usingCluster = FALSE)



op <- par(las = 2)
randomVarImpsRFplot(rf.rvi,NAGA.rf2, show.var.names = TRUE)
par(op)

NAGA.rf2$confusion
NAGA.rf2$importance
NAGA.rf2$predicted
```

```
##############################################################################
################################  BOOTSTRAP  #################################
##############################################################################

        rf.vsb <- varSelRFBoot(dados,nag$Stability_status,
                        ntree = 3000,
                        ntreeIterat = 2000,
                        bootnumber = 100,
                        usingCluster = FALSE,
                        Errornum = TRUE,
                        oobProb = TRUE,
                        srf = rf.vs2)
        rf.vsb
        summary(rf.vsb)
        plot(rf.vsb)


        library(rpart)
        library(rpart.plot)
        library(caret)
        library(partykit)
        library(readxl)
        library(factoextra)
        library(FactoMineR)

##############################################################################
############################### DATA for PCA  ################################
##############################################################################

        nag3 <- read_excel("C:/Users/SAMSUNG/Desktop/r/tcc/nag.xlsx",
                        sheet = "Original")

        nag3$Stability_status <- factor(nag3$Stability_status)

##############################################################################

        nag10 <- read_excel("C:/Users/SAMSUNG/Desktop/r/tcc/nag.xlsx",
                        sheet = "semerrosID")

        nag10$Stability_status <- factor(nag10$Stability_status)
```

```
##############################################################################
############################  DATA for MODELS  ###############################
##############################################################################

        nag1 <- read_excel("C:/Users/SAMSUNG/Desktop/r/tcc/nag.xlsx",
                           sheet = "Model_NAIM")
        nag1[sapply(nag1, is.numeric)] <- lapply(nag1[sapply(nag1, is.numeric)],
                                    as.factor)
        nag1$Stability_status <- factor(nag1$Stability_status)

##############################################################################

        nag2 <- read  excel("C:/Users/SAMSUNG/Desktop/r/tcc/nag.xlsx",
                           sheet = "BALANCEADO")

        nag2[sapply(nag2, is.numeric)] <- lapply(nag2[sapply(nag2, is.numeric)],
                                    as.factor)
        nag2$Stability_status <- factor(nag2$Stability_status)

##############################################################################

        nag5 <- read_excel("C:/Users/SAMSUNG/Desktop/r/tcc/nag.xlsx",
                           sheet = "qslope")

        nag5[sapply(nag5, is.numeric)] <- lapply(nag5[sapply(nag5, is.numeric)],
                                    as.factor)
        nag5$Stability_status <- factor(nag5$Stability_status)

##############################################################################

        nag4 <- read_excel("C:/Users/SAMSUNG/Desktop/r/tcc/nag.xlsx",
                           sheet = "Original")

        nag4[sapply(nag4, is.numeric)] <- lapply(nag4[sapply(nag4, is.numeric)],
                                    as.factor)
        nag4$Stability_status <- factor(nag4$Stability_status)

##############################################################################

        nag9 <- read_excel("C:/Users/SAMSUNG/Desktop/r/tcc/nag.xlsx",
                           sheet = "semerrosID")

        nag9[sapply(nag9, is.numeric)] <- lapply(nag9[sapply(nag9, is.numeric)],
                                    as.factor)
        nag9$Stability_status <- factor(nag9$Stability_status)

##############################################################################
        nag7 <- read_excel("C:/Users/SAMSUNG/Desktop/r/tcc/nag.xlsx",
                           sheet = "Model_NAIM_semerros")
```

```
nag7[sapply(nag7, is.numeric)] <- lapply(nag7[sapply(nag7, is.numeric)],
                                  as.factor)
nag7$Stability_status <- factor(nag7$Stability_status)

################################################################################

nag8 <- read_excel("C:/Users/SAMSUNG/Desktop/r/tcc/nag.xlsx",
                   sheet = "Model_SILVA_semerros")

nag8[sapply(nag8, is.numeric)] <- lapply(nag8[sapply(nag8, is.numeric)],
                                  as.factor)
nag8$Stability_status <- factor(nag8$Stability_status)

################################################################################

nag6 <- read_excel("C:/Users/SAMSUNG/Desktop/r/tcc/nag.xlsx",
                   sheet = "Model_allan")
nag6[sapply(nag6, is.numeric)] <- lapply(nag6[sapply(nag6, is.numeric)],
                                  as.factor)
nag6$Stability_status <- factor(nag6$Stability_status)

################################################################################
##############################                                  PCA      GRAFIC
##############################
################################################################################

nag.pca <- PCA(nag3[,-19], graph = TRUE)

fviz_pca_ind(nag.pca,
             label = "all",
             habillage = nag3$Stability_status,
             palette = c("#00AFBB", "#E7B800", "#FC4E07"),
             addEllipses = TRUE,
             repel = TRUE,
             title = "PCA dos dados Originais n?o balanceados"
             )

nag.pca <- PCA(nag10[,-19], graph = TRUE)

fviz_pca_ind(nag.pca,
             label = "none",
             habillage = nag10$Stability_status,
             palette = c("#00AFBB", "#E7B800", "#FC4E07"),
             addEllipses = TRUE,
             repel = TRUE,
             title = "PCA dos dados Originais n?o balanceados e sem amostras
problem?ticas"
)
```

```
##############################################################################
####################  CLASSIFICATION TREE MODEL ##########################
#########################          Math Model          ######################
########################## PARTITION RULE ###################################

        {set.seed(2356547)
        partitionrule <- createDataPartition(nag1$Stability_status,p=0.8, list=FALSE)
        trainingset1 <- nag1[partitionrule,]
        testingset1 <- nag1[-partitionrule,]
        prop.table(table(testingset1$Stability_status))}


################################ TREE MODEL###################################
        set.seed(9999)
        fit1 <- rpart(Stability_status~., data = trainingset1, method = 'class')
        rpart.plot(fit1, extra = 104)

        rparty.tree1 <- as.party(fit1)
        plot(rparty.tree1)

        printcp(fit1)
        plotcp(fit1)
        summary(fit1)


############################### TESTING TREE MODEL############################

        predict_unseen1 <-predict(fit1, testingset1, type = 'class')

        table_mat <- table(testingset1$Stability_status, predict_unseen1)
        table_mat
        confusionMatrix(table_mat)

##############################################################################
####################  CLASSIFICATION TREE MODEL ##########################
##############################################################################
#############################          GERAL Model          ##################
############################PARTITION RULE ###################################

        {set.seed(235997)
        partitionrule <- createDataPartition(nag2$Stability_status,p=0.8, list=FALSE)
        trainingset2 <- nag2[partitionrule,]
        testingset2 <- nag2[-partitionrule,]
        prop.table(table(testingset2$Stability_status))}


################################### TREE MODEL###############################
        set.seed(32523462)
        fit2 <- rpart(Stability  status~., data = trainingset2, method = 'class')
        rpart.plot(fit2, extra = 104)
```

```
rparty.tree2 <- as.party(fit2)
plot(rparty.tree2)

printcp(fit2)
plotcp(fit2)
summary(fit2)

############################## TESTING TREE MODEL#############################

predict_unseen2 <-predict(fit2, testingset2, type = 'class')

table_mat <- table(testingset2$Stability_status, predict_unseen2)
table_mat

confusionMatrix(table_mat)

#############################################################################
######################## CLASSIFICATION TREE MODEL###########################
#############################################################################
############################## QSLOPE Model ################################
################################## PARTITION RULE#############################

{set.seed(2356547)
partitionrule <- createDataPartition(nag5$Stability_status,p=0.8, list=FALSE)

trainingset5 <- nag5[partitionrule,]
testingset5 <- nag5[-partitionrule,]
prop.table(table(testingset5$Stability_status))}

################################### TREE
MODEL#################################
set.seed(9999)
fit5 <- rpart(Stability_status~., data = trainingset5, method = 'class')
rpart.plot(fit5, extra = 104)

rparty.tree5 <- as.party(fit5)
plot(rparty.tree5)

printcp(fit5)
plotcp(fit5)
summary(fit5)

############################## TESTING TREE MODEL#############################

predict_unseen3 <-predict(fit5, nag4, type = 'class')

table_mat <- table(nag4$Stability_status, predict_unseen3)
table_mat
confusionMatrix(table_mat)
```

```
##############################################################################
#################### CLASSIFICATION TREE MODEL#########################
##############################################################################
############################       SANTOS Model       ############################
############################    PARTITION RULE    ############################

        {set.seed(2356547)
         partitionrule <- createDataPartition(nag6$Stability_status,p=0.8, list=FALSE)

         trainingset6 <- nag6[partitionrule,]
         testingset6 <- nag6[-partitionrule,]
         prop.table(table(testingset6$Stability_status))}

################################ TREE MODEL ###################################

        fit6 <- rpart(Stability_status~., data = trainingset6, method = 'class', control =
rpart.control(cp = 0.010))
        rpart.plot(fit6, extra = 104)

        rparty.tree6 <- as.party(fit6)
        plot(rparty.tree6)

        printcp(fit6)
        plotcp(fit6)
        summary(fit6)

############################ TESTING TREE MODEL#################################

        predict_unseen6 <-predict(fit6,nag4, type = 'class')

        table_mat <- table(nag4$Stability_status, predict_unseen6)
        table_mat
        confusionMatrix(table_mat)



##############################################################################
#######################  CLASSIFICATION TREE MODEL#########################
##############################################################################
############################Math no erros Model       ############################
############################       PARTITION RULE    ############################

        {set.seed(2356547)
         partitionrule <- createDataPartition(nag7$Stability_status,p=0.8, list=FALSE)
         trainingset7 <- nag7[partitionrule,]
         testingset7 <- nag7[-partitionrule,]
         prop.table(table(testingset7$Stability_status))}
```

```
################################## TREE MODEL#################################
        set.seed(9999)
        fit7 <- rpart(Stability_status~., data = trainingset7, method = 'class')
        rpart.plot(fit7, extra = 104)

        rparty.tree7 <- as.party(fit7)
        plot(rparty.tree7)

        printcp(fit7)
        plotcp(fit7)
        summary(fit7)

########################### TESTING TREE MODEL#################################

        predict_unseen7 <-predict(fit7, nag9, type = 'class')

        table_mat <- table(nag9$Stability_status, predict_unseen7)
        table_mat
        confusionMatrix(table_mat)

##############################################################################
########################## CLASSIFICATION TREE MODEL#########################
##############################################################################
##########################SANTOS  no erros Model      ########################
############################## PARTITION RULE#################################

        {set.seed(2356547)
          partitionrule <- createDataPartition(nag8$Stability_status,p=0.8, list=FALSE)
          trainingset8 <- nag8[partitionrule,]
          testingset8 <- nag8[-partitionrule,]
          prop.table(table(testingset8$Stability_status))}


############################## TREE MODEL#####################################
        set.seed(9999)
        fit8 <- rpart(Stability_status~., data = trainingset8, method = 'class')
        rpart.plot(fit8, extra = 104)

        rparty.tree8 <- as.party(fit8)
        plot(rparty.tree8)

        printcp(fit8)
        plotcp(fit8)
        summary(fit8)
```

```
######################## TESTING TREE MODEL#################################

        predict_unseen8 <-predict(fit8, nag9, type = 'class')

        table_mat <- table(nag9$Stability_status, predict_unseen8)
        table_mat
        confusionMatrix(table_mat)

#############################################################################
##############################        PCA output       ##############################
#############################################################################
##############################       Math Model        ##############################

        nag.pca1 <- PCA(nag3[,-19], graph = FALSE)

        dt1 <- fviz_pca_ind(nag.pca1,
                label = "all",
                habillage = predict_unseen1,
                palette = c("#00AFBB", "#E7B800", "#FC4E07"),
                addEllipses = FALSE,
                repel = TRUE,
                title = "PCA para o modelo de ?rvore matem?tico com predi??o dos dados
n?o balanceados"
        )

        print(dt1)

#############################################################################
##############################        PCA output       ##############################
#############################################################################
##############################       GERAL Model       ##############################

        nag.pca2 <- PCA(nag3[,-19], graph = FALSE)

        dt2 <- fviz_pca_ind(nag.pca2,
                label = "all",
                habillage = predict_unseen2,
                palette = c("#00AFBB", "#E7B800", "#FC4E07"),
                addEllipses = FALSE,
                repel = TRUE,
                title = "PCA para o modelo de ?rvore geral com predi??o dos dados n?o
balanceados"
        )

        print(dt2)
```

```
################################################################################
################################  PCA output  ##################################
################################################################################
################################  QSLOPE Model  ################################

        nag.pca3 <- PCA(nag3[,-19], graph = FALSE)

        dt3 <- fviz_pca_ind(nag.pca3,
                        label = "all",
                        habillage = predict_unseen3,
                        palette = c("#00AFBB", "#E7B800", "#FC4E07"),
                        addEllipses = FALSE,
                        repel = TRUE,
                        title = "PCA para o modelo de ?rvore Q-Slope com predi??o dos dados
n?o balanceados"
        )

        print(dt3)

################################################################################
################################  PCA output  ##################################
################################################################################
################################                                SANTOS  Model
##############################

        nag.pca6 <- PCA(nag3[,-19], graph = FALSE)

        dt6 <- fviz_pca_ind(nag.pca6,
                        label = "all",
                        habillage = predict_unseen6,
                        palette = c("#00AFBB", "#E7B800", "#FC4E07"),
                        addEllipses = FALSE,
                        repel = TRUE,
                        title = "PCA para o modelo de ?rvore SILVA com predi??o dos dados
n?o balanceados"
        )

        print(dt6)
```

```
################################################################################
############################        PCA output        ##########################
################################################################################
############################    MATH no erros Model    ##########################

        nag.pca7 <- PCA(nag10[,-19], graph = FALSE)

        dt7<- fviz_pca_ind(nag.pca7,
                        label = "all",
                        habillage = predict_unseen7,
                        palette = c("#00AFBB", "#E7B800", "#FC4E07"),
                        addEllipses = FALSE,
                        repel = TRUE,
                        title = "PCA para o modelo de ?rvore matem?tica sem erros com
predi??o dos dados n?o balanceados"
        )

        print(dt7)

################################################################################
############################        PCA output        ##########################
################################################################################
############################    SILVA no erros Model   ##########################

        nag.pca8 <- PCA(nag10[,-19], graph = FALSE)

        dt8<- fviz_pca_ind(nag.pca8,
                        label = "all",
                        habillage = predict_unseen8,
                        palette = c("#00AFBB", "#E7B800", "#FC4E07"),
                        addEllipses = FALSE,
                        repel = TRUE,
                        title = "PCA para o modelo de ?rvore SILVA sem erros com predi??o dos
dados n?o balanceados"
        )

        print(dt8)
```

```
##############################################################################
##########################PCA COMPARATION    ##################################
##############################################################################
################################      ALL Models       ###########################

        {
        True<-nag3$Stability_status
        Estimate<-dt2$data$Groups

        ggplot(cbind(dt2$data,nag3$Stability_status),
        aes(x=dt2$data$x,y=dt2$data$y,col=True,  shape=Estimate))+theme(plot.title =
element_text(face = "bold", size = (20),hjust = 0.5))+labs(y= "Componente 2", x =
"Componente 1")+ggtitle("Gr?fico dos componentes principais comparativo entre a
estabilidade \nverdadeira e estimada pelo modelo geral")
+labs(fill="Original")+geom_point(size=3)+geom_text(aes(label =dt2$data$name,hjust=-
0.5, vjust=0))
        }

        {
          True<-nag3$Stability_status
          Estimate<-dt1$data$Groups

          ggplot(cbind(dt1$data,nag3$Stability_status),
                aes(x=dt1$data$x,y=dt1$data$y,col=True, shape=Estimate))+theme(plot.title =
element_text(face = "bold", size = (20),hjust = 0.5))+labs(y= "Componente 2", x =
"Componente 1")+ggtitle("Gr?fico dos componentes principais comparativo entre a
estabilidade \nverdadeira e estimada pelo modelo matem?tico")
+labs(fill="Original")+geom_point(size=3)+geom_text(aes(label =dt1$data$name,hjust=-
0.5, vjust=0))
        }

        {
          True<-nag3$Stability_status
          Estimate<-dt3$data$Groups

          ggplot(cbind(dt3$data,nag3$Stability_status),
                aes(x=dt3$data$x,y=dt3$data$y,col=True, shape=Estimate))+theme(plot.title =
element_text(face = "bold", size = (20),hjust = 0.5))+labs(y= "Componente 2", x =
"Componente 1")+ggtitle("Gr?fico dos componentes principais comparativo entre a
estabilidade \nverdadeira e estimada pelo modelo Q-Slope")
+labs(fill="Original")+geom_point(size=3)+geom_text(aes(label =dt3$data$name,hjust=-
0.5, vjust=0))
        }

        {
          True<-nag3$Stability_status
          Estimate<-dt6$data$Groups

          ggplot(cbind(dt6$data,nag3$Stability_status),
```

```
        aes(x=dt6$data$x,y=dt6$data$y,col=True, shape=Estimate))+theme(plot.title =
element_text(face = "bold", size = (20),hjust = 0.5))+labs(y= "Componente 2", x =
"Componente 1")+ggtitle("Gr?fico dos componentes principais comparativo entre a
estabilidade \nverdadeira e estimada pelo modelo SILVA")
+labs(fill="Original")+geom_point(size=3)+geom_text(aes(label =dt6$data$name,hjust=-
0.5, vjust=0))
    }

    {
     True<-nag10$Stability_status
     Estimate<-dt7$data$Groups

     ggplot(cbind(dt7$data,nag10$Stability  status),
        aes(x=dt7$data$x,y=dt7$data$y,col=True, shape=Estimate))+theme(plot.title =
element_text(face = "bold", size = (20),hjust = 0.5))+labs(y= "Componente 2", x =
"Componente 1")+ggtitle("Gr?fico dos componentes principais comparativo entre a
estabilidade \nverdadeira e estimada pelo modelo Matematico sem amostras problem?ticas")
+labs(fill="Original")+geom_point(size=3)+geom_text(aes(label =dt7$data$name,hjust=-
0.5, vjust=0))
    }

    {
     True<-nag10$Stability_status
     Estimate<-dt8$data$Groups

     ggplot(cbind(dt8$data,nag10$Stability_status),
        aes(x=dt8$data$x,y=dt8$data$y,col=True, shape=Estimate))+theme(plot.title =
element_text(face = "bold", size = (20),hjust = 0.5))+labs(y= "Componente 2", x =
"Componente 1")+ggtitle("Gr?fico dos componentes principais comparativo entre a
estabilidade \nverdadeira e estimada pelo modelo SILVA sem amostras problem?ticas")
+labs(fill="Original")+geom_point(size=3)+geom_text(aes(label =dt8$data$name,hjust=-
0.5, vjust=0))
    }
```

```
################################### Balanced data  ################################
        library(ROSE)
        library(rpart)
        library(caret)
        library(readxl)

        nag <- read_excel("C:/Users/SAMSUNG/Desktop/r/tcc/nag.xlsx",
                    sheet = "semerrosID")

        nag$Stability_status <- factor(nag$Stability_status)

        table(nag$Stability_status)
#################################################################################
############ SMOTE Algorithm For Unbalanced Classification Problems #############
#################################################################################

        data_balanced_over <- ovun.sample(Stability_status ~ ., data = nag, method =
"over",N = 84, seed = 12)$data

        table(data_balanced_over$Stability_status)

        fit.over <- rpart(Stability_status~., data=data_balanced_over)
        pred.over <- predict(fit.over, data=nag, type="class")
        roc.curve(data_balanced_over$Stability_status, pred.over,
              main="ROC curve \n (Half circle depleted data balanced by SMOTE
oversampling)")
        confusionMatrix(data=pred.over,data_balanced_over$Stability_status)

#################################################################################
######################### Randomly Over Sampling Examples########################
#################################################################################

        nag1 <- read_excel("C:/Users/SAMSUNG/Desktop/r/tcc/nag.xlsx",
                    sheet = "semerrosFSB_ST")

        nag1[sapply(nag1, is.numeric)] <- lapply(nag1[sapply(nag1, is.numeric)],
                                  as.factor)
        nag1$Stability_status <- factor(nag1$Stability_status)
        str(nag1)

        Rosedata2 <- ROSE(Stability_status ~ ., data = nag1 , seed = 12, N=64)$data

        table(Rosedata2$Stability_status)

        fit.rose2 <- rpart(Stability_status~., data=Rosedata2)
        pred.rose2 <- predict(fit.rose2, data=nag1, type="class")
        roc.curve(Rosedata2$Stability_status, pred.rose2,
              main="ROC curve \n (Half circle depleted data balanced by ROSE)")
        confusionMatrix(data=pred.rose2,Rosedata2$Stability_status)
```

```
#################################################################################
#################################################################################
#################################################################################

        nag2 <- read_excel("C:/Users/SAMSUNG/Desktop/r/tcc/nag.xlsx",
                           sheet = "semerrosOF_ST")

        nag2[sapply(nag2, is.numeric)] <- lapply(nag2[sapply(nag2, is.numeric)],
                                         as.factor)
        nag2$Stability_status <- factor(nag2$Stability_status)
        str(nag2)

        Rosedata3 <- ROSE(Stability_status ~ ., data = nag2 , seed = 12, N=64)$data

        table(Rosedata3$Stability_status)

        fit.rose3 <- rpart(Stability_status~., data=Rosedata3)
        pred.rose3 <- predict(fit.rose3, data=nag2, type="class")
        roc.curve(Rosedata3$Stability_status, pred.rose3,
                main="ROC curve \n (Half circle depleted data balanced by ROSE)")
        confusionMatrix(data=pred.rose3,Rosedata3$Stability_status)

#################################################################################
```

					GERAL	GERAL	DESMONTE	ANUAL	E PREVIA	ADE
1.00	0.60	0.80	0.30	0.60	0.30	0.60	1.00	0.00	0.30	FSB
1.00	0.80	0.80	0.30	0.60	0.30	0.80	1.00	0.00	0.60	FSB
0.80	0.60	0.80	0.60	1.00	0.30	1.00	0.80	0.00	0.30	FSB
0.80	0.80	0.80	0.60	0.80	0.30	1.00	0.80	0.00	0.60	FSB
1.00	0.60	0.80	0.30	0.80	0.30	1.00	1.00	0.60	0.00	FSB
1.00	0.80	0.80	0.30	0.80	0.60	0.60	0.00	0.60	0.80	FSB
1.00	0.80	0.60	0.60	0.60	0.80	0.60	0.00	0.60	0.30	FSB
1.00	0.60	0.80	0.60	0.60	0.80	1.00	0.80	0.30	0.30	FSB
1.00	0.60	0.60	0.60	0.60	1.00	0.80	0.80	0.30	0.30	FSB
1.00	1.00	0.80	0.30	0.80	0.60	1.00	0.00	0.00	0.60	FSB
1.00	1.00	0.80	0.60	0.60	0.80	1.00	0.00	0.80	0.30	FSB
1.00	1.00	0.80	0.60	0.80	0.80	1.00	0.00	0.80	0.60	FSB
1.00	0.80	0.80	0.80	0.80	0.80	1.00	0.00	0.80	0.60	FSB
1.00	0.80	0.80	0.60	0.80	0.80	1.00	0.00	0.80	0.30	FSB
1.00	0.60	0.80	0.30	0.60	0.80	1.00	0.00	0.80	0.30	FSB
1.00	0.80	0.80	0.60	0.80	0.80	1.00	0.00	1.00	0.30	FSB
1.00	0.80	0.80	0.60	0.60	0.60	0.80	0.60	1.00	0.30	FSB
0.80	0.80	0.80	0.30	0.80	0.80	0.60	0.60	1.00	0.30	FSB
1.00	0.60	0.60	0.30	0.60	0.30	1.00	0.80	0.00	0.30	FSB
1.00	1.00	0.60	0.80	0.80	0.30	1.00	0.80	0.00	0.30	FSB
1.00	0.60	0.60	0.60	0.60	0.30	0.80	0.80	1.00	0.00	FSB
1.00	0.80	0.80	0.30	0.80	0.60	0.60	0.80	0.00	0.30	FSB
1.00	0.60	1.00	0.30	0.60	0.60	1.00	0.80	0.60	0.00	FSB
1.00	0.30	0.80	0.60	0.60	0.00	1.00	0.60	1.00	0.60	FSB
1.00	0.80	0.80	0.80	1.00	0.30	0.60	1.00	0.80	0.80	OF
1.00	1.00	1.00	0.60	0.60	0.30	0.60	1.00	0.00	0.60	OF
1.00	1.00	0.80	0.60	0.60	0.80	0.30	0.00	0.60	0.80	OF
1.00	1.00	0.80	0.60	0.60	0.60	0.60	0.00	0.60	0.80	OF
1.00	0.80	0.80	0.60	0.60	0.30	0.80	0.80	0.30	0.60	OF
1.00	0.80	0.80	0.80	0.80	0.60	0.80	0.80	0.30	0.80	OF
1.00	1.00	0.80	0.60	0.80	0.30	1.00	0.00	0.00	1.00	OF
1.00	1.00	0.80	0.80	0.60	0.60	1.00	0.80	0.00	0.60	OF
1.00	1.00	0.80	0.60	0.80	0.30	1.00	0.80	0.00	0.60	OF
1.00	0.80	0.80	0.60	0.80	0.30	0.80	0.80	1.00	0.80	OF
1.00	1.00	0.80	0.60	0.60	0.30	0.80	0.80	1.00	0.80	OF
1.00	0.80	0.80	0.80	0.80	0.30	0.80	0.80	1.00	0.60	OF
1.00	0.60	0.80	0.30	0.60	0.30	0.80	0.80	0.30	0.60	OF
1.00	0.60	0.80	0.60	0.30	0.00	1.00	0.80	0.30	0.60	OF
1.00	1.00	1.00	0.60	0.30	0.30	0.80	0.80	0.30	0.60	OF

ÁGUA SUBTERRÂNEA	NÚMERO DE FAMÍLIAS DE DESCONTINUIDADE	PERSISTÊNCIA	ESPAÇAMENTO	ORIENTAÇÃO	ABERTURA	RUGOSIDADE	PREENCHIMENTO
0.60	0.80	0.80	1.00	0.80	0.80	0.60	0.80
0.60	0.80	0.60	1.00	0.60	0.60	0.60	0.80
0.30	0.80	0.30	1.00	0.60	0.60	0.30	0.80
0.00	0.80	0.80	1.00	0.80	0.80	0.60	0.80
0.60	0.80	0.60	1.00	0.00	0.80	0.30	0.30
0.00	0.80	0.30	1.00	0.80	0.60	0.60	0.60
0.30	0.80	0.80	1.00	0.80	0.80	0.60	0.80
0.60	0.80	0.30	1.00	0.80	0.80	0.60	0.80
0.30	1.00	0.80	1.00	0.60	0.60	0.30	0.60
0.60	0.80	0.30	1.00	0.80	0.80	0.60	0.60
0.30	0.60	0.00	1.00	0.30	0.80	0.60	0.60
0.60	1.00	0.00	1.00	0.30	0.60	0.30	0.80
0.60	0.80	0.00	1.00	0.60	0.80	0.30	0.80
0.30	0.60	0.00	1.00	0.60	0.80	0.60	0.60
0.30	0.80	0.30	0.60	0.30	0.80	0.30	0.60
0.30	0.80	0.60	0.80	0.80	0.80	0.30	0.60
0.60	0.80	0.60	1.00	0.30	0.80	0.60	0.80
0.30	0.80	0.30	0.80	0.60	1.00	0.30	0.80
0.60	0.80	0.30	1.00	0.60	0.80	0.30	0.30
0.60	0.80	0.00	1.00	0.30	0.80	0.30	0.30
0.60	0.80	0.00	1.00	0.60	0.60	0.30	0.60
0.30	0.80	0.00	1.00	0.60	0.80	0.60	0.30
0.60	0.80	0.30	1.00	0.60	0.80	0.30	0.30
0.30	1.00	0.30	0.80	0.60	0.60	0.60	0.60
0.60	0.80	0.30	1.00	0.80	0.60	0.80	0.60
0.60	1.00	0.00	1.00	0.60	0.60	0.30	0.60
0.30	0.60	0.00	0.80	0.30	0.60	0.30	0.30
0.30	0.80	0.30	0.80	0.60	0.60	0.30	0.80
0.30	1.00	0.00	1.00	0.30	0.80	0.00	0.80
0.30	0.80	0.30	1.00	0.80	0.60	0.30	0.80
0.30	0.60	0.00	0.80	0.60	0.80	0.30	0.80
0.00	1.00	0.00	0.80	0.60	0.60	0.30	0.80
0.30	1.00	0.00	1.00	0.60	0.60	0.30	0.80
0.30	1.00	0.00	0.80	0.30	0.30	0.30	0.00
0.30	1.00	0.00	0.80	0.30	0.60	0.60	0.60
0.30	0.80	0.00	1.00	0.30	0.60	0.30	0.60
0.60	0.80	0.00	0.80	0.30	0.60	0.30	0.60
0.30	0.80	0.00	0.80	0.60	0.60	0.60	0.30
0.30	0.80	0.00	0.80	0.30	0.30	0.30	0.60
0.30	0.60	0.00	1.00	0.30	0.60	0.30	0.60
0.30	0.80	0.00	1.00	0.60	0.80	0.30	0.80
0.30	1.00	0.30	1.00	0.60	0.60	0.30	0.00

REFERÊNCIAS

ÁVILA, C. R.. **Determinação das propriedades mecânicas de maciços rochosos e/ou descontinuidades utilizando classificações geomecânicas** [manuscrito]: uma comparação entre os diversos métodos de classificação / Cássio Ricardo de Ávila. - 2012. 232f.: il., color.; grafs.; tabs

HOEK B.. Hoek-Brown criterion – 2002 edition.5th North American Rock Mechanics Symposium. Toronto, Canada. Vol. 1,(2002), p.267-273.

BAR, N. & BARTON, N. **The Q-Slope Method for Rock Slope Engineering**. Rock Mechanics And Rock Engineering, [S.L.], v. 50, n. 12, p. 3307-3322, 31 ago. 2017. Springer Science and Business Media LLC. http://dx.doi.org/10.1007/s00603-017-1305-0.

BARTON, N.; LIEN, R.; LUNDE, J. **Engineering classification of rock masses for the design of tunnel support**. Rock Mechanics 6, 189–236 (1974). https://doi.org/10.1007/BF01239496

BARTON, N.; LIEN, R.; LUNDE, J. **Engineering classification of rock masses for the design of tunnel support.** Rock Mechanics Felsmechanik Mecanique Des Roches, [S.L.], v. 6, n. 4, p. 189-236, dez. 1974. Springer Science and Business Media LLC. http://dx.doi.org/10.1007/bf01239496.

BERETTA, F.; RODRIGUES, A. L.; PERONI, R. L.; COSTA, J. F. C. L.. **Automated lithological classification using UAV and machine learning on an open cast mine**. Applied Earth Science, [S.L.], v. 128, n. 3, p. 79-88, 20 fev. 2019. Informa UK Limited. http://dx.doi.org/10.1080/25726838.2019.1578031.

BIENAYMÉ, I.-J. **Considérations àl'appui de la découverte de Laplace**, Comptes Rendus de l'Académie des Sciences. 37: 309–324. (1853)

BIENIAWSKI, Z. T**. Engineering rock mass classifications** : a complete manual for engineers and geologists in mining, civil, and petroleum engineering. Wiley-Interscience.(1989), pp. 40–47. ISBN 0-471-60172-1.

BREIMAN, L. **Bagging predictors. Machine Learning** 26(2), (1996a), 123–140.

BREIMAN, L. **Bagging Predictors. Machine Learning**, [S.L.], v. 24, n. 2, p. 123-140, 1996b. Springer Science and Business Media LLC. http://dx.doi.org/10.1023/a:1018054314350.

BREIMAN, L. **Classification and Regression by randomForest**. Machine Learning, [S.L.], v. 45, n. 1, p. 5-32, 2001a. Springer Science and Business Media LLC. http://dx.doi.org/10.1023/a:1010933404324.

BREIMAN, L. **Random Forest. Machine Learning**, [S.L.], v. 45, n. 1, p. 5-32, 2001b. Springer Science and Business Media LLC. http://dx.doi.org/10.1023/a:1010933404324.

BREIMAN, L; FRIEDMAN, J. H.; OLSHEN, R. A.; STONE, C. J. **Classification and Regression Trees. Classification and Regression Trees**, [S.L.], v. 1, n. 1, p. 1-368, 19 out. 2017. Routledge. http://dx.doi.org/10.1201/9781315139470.

DEERE DU, Hendron AJ, Patton F, Cording EJ (1967) **Design of surface and near surface excavations in rock. In: Fairhust C** (ed) Proceedings of the 8th U.S. symposium on rock mechanics: failure and breakage of rock, AIME, New York, pp 237–302

DEERE, D U (1964). **Technical description of rock cores**, Rock Mechanics Engineering Geology, 1 (16-22).

DEERE, D U (1989). **Rock quality designation (RQD) after twenty years**, U.S. Army Corps of Engineers Contract Report GL-89-1, Waterways Experiment Station, Vicksburg, MS (67).

DE TOLEDO, P. E., De Freitas, M. H., & CGcol. Laboratory testing and parameters controlling the shear strength of filled rock joints. Géotechnique, 43(1), (1993), 1–19. doi:10.1680/geot.1993.43.1.1

DIAZ-Uriarte, R. and Alvarez de Andres, S. (2005) **Variable selection from random forests: application to gene expression data**. Tech. report. http://ligarto.org/rdiaz/Papers/rfVS/randomForestVarSel.html

DIETTERICH, T. (1998). **An experimental comparison of three methods for constructing ensembles of decision trees**: Bagging, boosting and randomization, Machine Learning, 1–22.

VLADISLAVLEVA E., Smits G. and den Hertog D., **On the Importance of Data Balancing for Symbolic Regression**, in IEEE Transactions on Evolutionary Computation, vol. 14, no. 2, pp. 252-277, April 2010, doi: 10.1109/TEVC.2009.2029697.

EINSTEIN HH, Veneziano D, Baecher GB, O'Reilly KJ. **The effect of discontinuity persistence on rock sloe stability**. Int J Rock Mech Sci Geomech Abstr 1983;20:227–36.

FOOKES, P. G., Gourley, C. S., & Ohikere, C. (1988). **Rock weathering in engineering time**. Quarterly Journal of Engineering Geology and Hydrogeology, 21(1), 33–57. doi:10.1144/gsl.qjeg.1988.021.01.03

HO, T. K. (1998). **The random subspace method for constructing decision forests**. IEEE Trans. on Pattern Analysis and Machine Intelligence, 20(8), 832–844.

HOTHORN, Torsten, and Achim Zeileis. 2021. Partykit: A Toolkit for Recursive Partytioning. http://partykit.r-forge.r-project.org/partykit/.

HUDSON JA. **Rock engineering systems, theory and practice**. Chichester: Ellis Horwood; 1992.

PEARSON K., **On lines and planes of closest fit to systems of points in space**, Philosophical Magazine, (6) 2 (1901) 559-572.

LANTZ, Brett. Machine Learning with R. Birmingham: Packt Publishing Ltd., 2013. 396 p.

LEAL, Filipe Lívio Nunes. **Desenvolvimento de um método de classificação para desmonte a explosivo numa mina de minério de ferro**. Belo Horizonte: UFMG, 1997. 252 p. Dissertação de Mestrado em tecnologia mineral – Escola de Engenharia, Universidade Federal de Minas Gerais, 1994.

LUNARDON N., Menardi G., and Torelli N.. R package ROSE: Random Over-Sampling Examples (version 0.0-3). Università di Trieste and Università di Padova, Italia, 2013. URL http://cran.r-project.org/web/packages/ROSE/index.html. [p79]

KASSAMBARA A., Extract and Visualize the Results of Multivariate Data Analyses, Cran Package, 2020. URL: https://cran.r-project.org/web/packages/factoextra/factoextra.pdf

KLEN AM, Lana MS , Fuzzy algorithm of discontinuity sets. REM: Rev Esc Minas 67:(2014),439–445

MANOJ, Khandelwal; M., Monjezi. **Prediction of flyrock in open pit blasting operation using machine learning method**. International Journal Of Mining Science And Technology, [S.L.], v. 23, n. 3, p. 313-316, maio 2013. Elsevier BV. http://dx.doi.org/10.1016/j.ijmst.2013.05.005.

MENARDI, Giovanna; TORELLI, Nicola. **Training and assessing classification rules with imbalanced data**. Data Mining And Knowledge Discovery, [S.L.], v. 28, n. 1, p. 92-122, 30 out. 2012. Springer Science and Business Media LLC. http://dx.doi.org/10.1007/s10618-012-0295-5.

MYLES, Anthony J.; FEUDALE, Robert N.; LIU, Yang; WOODY, Nathaniel A.; BROWN, Steven D.. **An introduction to decision tree modeling.** Journal Of Chemometrics, [S.L.], v. 18, n. 6, p. 275-285, jun. 2004. Wiley. http://dx.doi.org/10.1002/cem.873.

NGUYEN, Hoang; BUI, Xuan-Nam; TRAN, Quang-Hieu; VAN HOA, Pham; NGUYEN, Dinh-An; HOA, Le Thi Thu; LE, Qui-Thao; DO, Ngoc-Hoan; BAO, Tran Dinh; BUI, Hoang-Bac. **Correction to: a comparative study of empirical and ensemble machine learning algorithms in predicting air over-pressure in open-pit coal mine**. Acta Geophysica, [S.L.], v. 1, n. 68, p. 325-356, 30 mar. 2021. Springer Science and Business Media LLC. http://dx.doi.org/10.1007/s11600-021-00576-8.

PALMSTRÖM, A. **RMi – a rock mass characterization system for rock engineering purposes**. Ph.D. thesis, Oslo University, Norway, 1995, 400 p

PINHEIRO, Marisa; VALLEJOS, Javier; MIRANDA, Tiago; EMERY, Xavier. **Geostatistical simulation to map the spatial heterogeneity of geomechanical parameters**: a case study with rock mass rating. Engineering Geology, [S.L.], v. 205, p. 93-103, abr. 2016. Elsevier BV. http://dx.doi.org/10.1016/j.enggeo.2016.03.003.

R CORE TEAM (2016). R: **A language and environment for statistical computing**. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

ROMANA, M. **New adjustment ratings for application of Bieniawski classification to slopes**. Int. Symp. on the role of rock mechanics ISRM. Zacatecas, 1985, p. 49-53.

ROMANA, Manuel R. (1993). **A Geomechanical Classification for Slopes: Slope Mass Rating**. Rock Testing and Site Characterization. pp. 575–600. doi:10.1016/B978-0-08-042066-0.50029-X. ISBN 978-0-08-042066-0.

SANTOS, A.E.M., Lana, M.S. & Pereira, T.M. Rock Mass **Classification by Multivariate Statistical Techniques and Artificial Intelligence**. Geotech Geol Eng 39, 2409–2430 (2021). https://doi.org/10.1007/s10706-020-01635-5

SANTOS, Tatiana Barreto dos; LANA, Milene Sabino; PEREIRA, Tiago Martins; CANBULAT, Ismet. **Quantitative hazard assessment system (Has-Q) for open pit mine slopes**. International Journal Of Mining Science And Technology, [S.L.], v. 29, n. 3, p. 419-427, maio 2019. Elsevier BV. http://dx.doi.org/10.1016/j.ijmst.2018.11.005.

SEBASTIEN Le, Julie Josse, Francois Husson (2008). FactoMineR: An R Package for Multivariate Analysis. Journal of Statistical Software, 25(1), 1-18. 10.18637/jss.v025.i01

STEAD, D., & Wolter, A. (2015). **A critical review of rock slope failure mechanisms: The importance of structural geology. Journal of Structural Geology**, 74, 1–23. doi:10.1016/j.jsg.2015.02.002

TAO, Zhigang; ZHU, Chun; ZHENG, Xiaohui; HE, Manchao. **Slope stability evaluation and monitoring of Tonglushan ancient copper mine relics**. Advances In Mechanical Engineering, [S.L.], v. 10, n. 8, p. 1-16, ago. 2018. SAGE Publications. http://dx.doi.org/10.1177/1687814018791707.

THERNEAU, T., & Atkinson, B. (2019). rpart: Recursive Partitioning and Regression Trees.Retrieved from https://cran.r-project.org/package=rpart

TIBSHIRANI, R. (1996). **Bias, variance, and prediction error for classification rules**. Technical Report, Statistics Department, University of Toronto.

VLADISLAVLEVA E., Smits G. and den Hertog D., **On the Importance of Data Balancing for Symbolic Regression**, in IEEE Transactions on Evolutionary Computation, vol. 14, no. 2, pp. 252-277, April 2010, doi: 10.1109/TEVC.2009.2029697.

WANG, Qun; ZHANG, Ruixin; WANG, Yangting; LV, Shuaikang. **Machine Learning-Based Driving Style Identification of Truck Drivers in Open-Pit Mines.** Electronics, [S.L.], v. 9, n. 1, p. 19, 24 dez. 2019. MDPI AG. http://dx.doi.org/10.3390/electronics9010019.

WARD, R.C., ROBSON, M.. **Principles of Hydrology** 3 ed. Londres: Mc Graw Hill Book Book Company, 1990. 365p.

WOLD, S., Esbensen, K., & Geladi, P. (1987). **Principal component analysis. Chemometrics and Intelligent Laboratory Systems**, 2(1-3), 37–52. doi:10.1016/0169-7439(87)80084-9

WSM-World Stress Map. Release 2008 of the world stress map; 2008 (Available online at /www.world-stress-map.orgS).

ZARE NAGHADEHI, M., Jimenez, R., Khalokakaie, R., Esmaeil Jalali, S., 2013. A new open pit mine slope instability index defined using the improved rock engineering systems approach. Int. J. Rock Mech. Min. Sci. 61, 1–14.

ZHANG, Xian-Da. **Machine Learning. A Matrix Algebra Approach To Artificial Intelligence**, [S.L.], p. 223-440, 2020. Springer Singapore. http://dx.doi.org/10.1007/978-981-15-2770-8_6.