

Predição de Custo Operacional de Mineração utilizando Aprendizagem de Máquina

Luã Lázaro Jesus dos Santos



**CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DE PERNAMBUCO**

Recife, 2017

Luã Lázaro Jesus dos Santos

Predição de Custo Operacional de Mineração utilizando Aprendizagem de Máquina

Trabalho de Conclusão de Curso apresentado no
Centro de Informática, da Universidade Federal de Pernambuco,
como requisito para obtenção de grau Bacharel em Engenharia da Computação

Orientador: Ricardo Martins de Abreu e Silva

Dezembro de 2017



Centro de Informática
Universidade Federal de Pernambuco

Trabalho de Conclusão de Curso de Engenharia da Computação intitulado ***Predição de Custo Operacional de Mineração utilizando Aprendizagem de Máquina*** de autoria de Luã Lázaro Jesus dos Santos, aprovado pela banca examinadora constituída pelos seguintes professores:

Prof. Dr. Ricardo Martins de Abreu e Silva
Universidade Federal de Pernambuco

Prof. Dr. Silvio Melo
Universidade Federal de Pernambuco

Coordenador do curso de Engenharia da Computação do Centro de Informática
Renato Mariz de Moraes
CIN/UFPE

Recife 14 de dezembro de 2017

Centro de Informática, Universidade Federal de Pernambuco
Av. Jorn. Aníbal Fernandes, s/n - Cidade Universitária, Recife - PE, 50740-560
Fone: +55 (81) 2126-8430

RESUMO

Uma boa estimativa do custo operacional na atividade mineradora é de fundamental importância para o estudo de viabilidade econômica da mina e por conseguinte, para o sucesso do empreendimento. Nos moldes atuais, a estimativa do Custo Operacional (OPEX) é realizado através de séries temporais, porém a necessidade de dados dispostos temporalmente e similares à mina que se deseja prever é uma grande limitação. Os algoritmos de aprendizagem de máquina são capazes de serem treinados exatamente sob o contexto da mina alvo de previsão, além de não precisarem de dados dispostos temporalmente. Este trabalho se propõe a gerar uma plataforma automática para pré-processamento, experimentação, treinamento e previsão do OPEX, de modo a permitir sua utilização sem a necessidade de um especialista.

Para tal, emprega-se as bibliotecas de aprendizagem de máquina Weka e LibSVM, ambas sob a linguagem Java. O algoritmo utilizado para a seleção de atributos é o *Support Vector Machine* (SVM), enquanto que para a experimentação e previsão são utilizados adicionalmente os algoritmos *MultiLayer Perceptron* (MLP) e *Reduced Error Pruning Tree* (REPTree). A base de dados utilizada para teste e validação é sintética, gerada através de um simulador geológico voltado à mineração. Sob esta base foi possível verificar uma boa performance nos três algoritmos experimentados, com destaque para o SVM pela sua boa acurácia, verificada via teste de hipótese, e com ressalva para o REPTree por apresentar indícios de overfitting.

Palavras chave: OPEX; Mineração; Aprendizagem de Máquina;

LISTA DE FIGURAS

1	Ilustração da importância da mineração.....	12
2	Bancada para desmonte de rocha com os parâmetros do plano de fogo.....	14
3	Hiperplano de separação. SVM.....	16
4	Modelo Perceptron Simples.....	17
5	Modelo MultiLayer Perceptron. Retropropagação.....	17
6	Gráfico do tempo total de execução antes e após a implementação da otimização via Micro-Execução da fase Seleção de Atributos.....	28

LISTA DE TABELAS

1	[ID da configuração] (Número de atributos resultantes, Tempo de processamento em segundos). Média dos atributos igual a 4. Média do tempo igual a 12. Configurações removidas em cinza. Razão da remoção em negrito.....	23
2	Atributos preditivos financeiros, média e desvio padrão.....	25
3	Atributos preditivos relativos ao Plano de Fogo, média e desvio padrão.....	26
4	MAE e desvio padrão médio obtido por cada tipo de algoritmo de aprendizagem presente no grid.....	29
5	Ranking das 10 melhores configurações do grid e seus respectivos índices	29
6	MAE e desvio padrão das configurações melhores ranqueadas, por algoritmo.....	30

LISTA DE ABREVIATURAS

OPEX – *Operational Expenditure*

SVM – Support Vector Machine

SVR – Support Vector Regression

MLP – Multilayer Perceptron

REPTree – Reduced Error Pruning Tree

RF – Random Forest

MSE – Mean Square Error

MAE – Mean Absolute Error

WEKA – *Waikato Environment for Knowledge Analysis*

LibSVM – Support Vector Machine Library

GPL – *General Public Licence*

RBF – *Radial Basis Function*

Conteúdo

1	INTRODUÇÃO	10
1.1	Motivação.....	10
1.2	Objetivos.....	11
1.2.1	Objetivo.....	11
1.3	Estrutura do trabalho.....	11
2	FUNDAMENTAÇÃO TEÓRICA	12
2.1	A mineração.....	12
2.1.1	Lavra por bancadas.....	13
2.2	Aprendizagem de Máquina.....	14
2.2.1	Aprendizagem de máquina supervisionada.....	15
2.2.2	Support Vector Machines (SVM).....	15
2.2.3	Multilayer Perceptron (MLP).....	17
2.2.4	Reduced Error Pruning Tree (REPTree).....	18
2.3	Técnica de avaliação de algoritmos de aprendizagem de máquina.....	18
2.3.1	Validação Cruzada K-Fold.....	19
2.3.2	Paired T-Test.....	19
3	METODOLOGIA DE PREDIÇÃO DO CUSTO OPERACIONAL	20
3.1	Pré-processamento.....	20
3.2	Experimentação.....	21
3.3	Treinamento e Predição.....	22
3.4	Otimização.....	22

4	RESULTADOS E ANÁLISE	24
4.1	Java.....	24
4.2	Base de dados.....	25
4.3	Parâmetros do Experimento.....	26
4.4	Análise dos dados.....	27
4.4.1	Seleção de Atributos.....	27
4.4.2	Micro-Execução da Seleção de Atributos.....	28
4.4.3	Acurácia de predição.....	28
5	CONSIDERAÇÕES FINAIS	31
5.1	Trabalhos futuros.....	31
6	REFERÊNCIAS	32

1 INTRODUÇÃO

1.1 Motivação

O planeta Terra possui sua crosta mais superficial formada por uma grande quantidade de rochas e minerais. Alguns desses minerais possuem valor econômico, uma vez que servem como matéria prima para recursos fundamentais à sociedade moderna. A estes minerais que possuem valor econômico, dá-se o nome de minérios (UNIDIS Minerais e rochas, 2009). A mineração tornou-se uma atividade de fundamental importância para a evolução tecnológica, saindo dos primórdios dos povos do mediterrâneo, que exploravam o solo através de túneis e galerias, até a era moderna atual, onde os minérios são obtidos através de complexos procedimentos de extração, resultando em uma indústria que movimenta bilhões de dólares anualmente (IBRAM, 2003).

Os procedimentos de extração de minérios exigem um elevado investimento. Mão de obra especializada e maquinários de grande porte são necessários para executar a extração em larga escala. Atualmente o controle dos gastos de operação é um dos maiores desafios desta indústria. O OPEX, abreviação para Custo de Operação, é um parâmetro indispensável para a análise da viabilidade econômica de uma mina, pois é através da subtração deste parâmetro da receita gerada que é obtido o lucro líquido. A estimativa desse custo é complexa e envolve variáveis associadas à muitas incertezas tendo em vista que seu cálculo é realizado com base em parâmetros de lavra.

Entende-se por lavra o conjunto de operações que tem por objetivo realizar de forma organizada, econômica, rápida e segura a extração do minério. Seus parâmetros, por sua vez, estão diretamente ligados a distribuição do minério, tipologia do solo e dificuldade de acesso à região de interesse (Lavra de Minas, 2017). O custo para amostragem do solo é elevado, o que dificulta o processo de mapeamento do minério e extração de características, gerando assim as incertezas que dificultam a estimativa do OPEX.

A aprendizagem de máquina é uma metodologia que permite gerar modelos analíticos de elevada precisão e representatividade através da análise automatizada de dados. Os algoritmos de aprendizagem funcionam através da extração de regularidades estatísticas, de forma interativa e sem intervenção humana direta, sendo capazes de descobrir padrões ocultos e/ou complexos nos dados.

Nos dias atuais, a estimativa do OPEX na mineração é realizada através de séries temporais. Esta técnica utiliza uma coleção de observações feitas sequencialmente ao longo do tempo, onde a ordem dos dados é fator fundamental (Jan Grandell, 2011). Desta forma é necessário ter, à priori, amostras de longos períodos de tempo, e estas, devem ser de minas semelhantes à que se deseja prever. Dados assim nem sempre estão disponíveis, sendo este um grande limitante. Outra característica restritiva deste modelo de predição é que os dados vizinhos são ditos sempre dependentes. A aprendizagem de máquina mostra-se uma alternativa muito interessante e robusta, sendo capaz de trabalhar com conjuntos de dados não ordenados e de mensurar a correlação destes dados. Além disso, as máquinas de aprendizagem não exigem necessariamente dados coletados durante longos períodos de

tempo, sendo capazes de lidar com amostras fornecidas durante o processo de extração da própria mina, aumentando assim a probabilidade de resultados mais precisos.

1.2 Objetivos

1.2.1 Objetivo

Este trabalho tem como objetivo o estudo de viabilidade e desenvolvimento de uma plataforma para predição do Custo Operacional de atividades mineradoras. Esta plataforma deve ser automática e robusta para que possa ser executada facilmente por um profissional da mineração e performe bem sobre diferentes bases de dados. A base de dados para validação é sintética, gerada a partir de um simulador geológico voltado à mineração.

1.3 Estrutura do trabalho

Para atingir os objetivos citados anteriormente, este trabalho será estruturado do seguinte modo: O Capítulo 2 traz a fundamentação teórica do projeto, detalhando os conceitos intrínsecos à mineração, além dos conceitos de aprendizagem de máquina e das máquinas de aprendizagem utilizadas. Ainda no Capítulo 2 são apresentadas as técnicas de avaliação dos regressores a serem utilizadas. No Capítulo 3 é apresentada a metodologia sob a qual a plataforma foi desenvolvida. No Capítulo 4 é possível verificar os resultados obtidos e por fim no Capítulo 5 temos as considerações finais e trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 A Mineração

O processo de expansão e evolução da civilização humana esteve sempre ligado a extração de recursos minerais. O primeiros indícios de atividade mineradora remetem à pré-história, onde pedras eram utilizadas para criação de utensílios de combate e exploração. A terminologia mineração, todavia, surgiu somente a partir do século XVI, quando os primeiros pesquisadores se reuniram com o objetivo de compreender composição química e importância dos minerais (Simineral, 2017). Este processo incitou o estudo sobre as vastas formas de utilização dos minerais e, por conseguinte, revelou sua representatividade econômica. Hoje, entende-se por minério todas as substâncias que compõem minerais e detêm valor econômico que viabilize sua extração.

Os minérios são a matéria prima da sociedade moderna estando presente nos mais diversas áreas. Desde edificações, eletrônicos e meios de transporte à utensílios domésticos, remédios e vestimentas. Tendo em vista o crescente aumento da população humana e sua busca por áreas urbanas a mineração constitui a base de seu desenvolvimento.

Nossa vida depende da mineração

A natureza é dividida em três reinos: animal, vegetal e mineral. Seria impossível haver o desenvolvimento da humanidade sem exploração mineral. O conhecimento do homem e as tecnologias criadas por ele fizeram com que o minério esteja presente no nosso dia a dia.

O minério é um recurso que só tem valor quando é retirado e processado. Debaxo da terra, não tem valor.



Figura 1: Ilustração da importância da mineração

Fonte: <http://www.vale.com/brasil>

A atividade mineradora possui uma série de peculiaridades que a torna diferente de outras indústrias. Segundo Girodo (2005) a mineração é uma atividade extrativa que trabalha com recursos naturais não renováveis, o que implica em buscá-los onde eles se encontram. Tendo em vista esse aspecto, nota-se que esta atividade é transitória, pois depende do volume de minério e de sua taxa de extração, ditada pelo plano de lavra. Uma vez esgotadas as

reservas do minério, encerram-se as atividades naquela mina. Cada reserva de minério é única, ou seja, cada jazida tem suas próprias características, e muitas vezes, o teor do minério pode estar distribuído irregularmente dentro da própria jazida. Assim, cada mina necessita de um projeto próprio e há grande dificuldade em processos de predição devido a sua natureza singular.

2.1.1 Lavra por bancadas

A lavra por bancadas é um dos principais métodos de lavra a céu aberto. O uso desta prática é ideal quando os minérios estão cobertos por uma camada densa de sedimentos. As bancadas são processadas de cima para baixo, até que seja alcançado todo o montante mineral. Este montante é então tratado e o estéril é dissociado, podendo muitas vezes ser devolvido ao local de extração, facilitando assim o processo de recuperação ambiental (Lavra de Minas, 2017). Entende-se por estéril todo o material rochoso sem valor econômico que vem agregado ao minério durante seu processo de extração.

Para um entendimento mais completo do procedimento se faz necessário o conhecimento de algumas terminologias (Britanite, 2010):

- Bancada: forma dada ao terreno rochoso pelos fogos sucessivos e constantes, composta de topo, praça e face;
- Altura da Bancada: é a altura vertical medida do topo a praça da bancada;
- Afastamento: distância entre a face da bancada e uma fileira de furos ou distância entre duas fileiras de furos;
- Profundidade do Furo: é o comprimento total perfurado que, devido a inclinação e da sub-furação, será maior que a altura bancada;
- Sub-Furação: é o comprimento perfurado abaixo da praça da bancada ou do “Greide” a ser atingido.
- Carga de Fundo: é uma carga reforçada, necessária no fundo do furo onde a rocha é mais presa;
- Carga de Coluna: é a carga acima a de fundo: não precisa ser tão concentrada quanto a de fundo já que a rocha desta região não é tão presa;
- Tampão: parte superior do furo que não é carregado com explosivos, mas sim com terra, pedrisco (mais aconselhável) ou outro material socado cuidadosamente que tem a finalidade de evitar que os gases provenientes da detonação escapem pela boca do furo, diminuindo a ação do explosivo;
- Razão de Carregamento: é a quantidade de explosivo usada para detonar um certo volume de rocha;
- Perfuração Específica: é a relação de metros perfurados por metros cúbicos de rocha detonada.

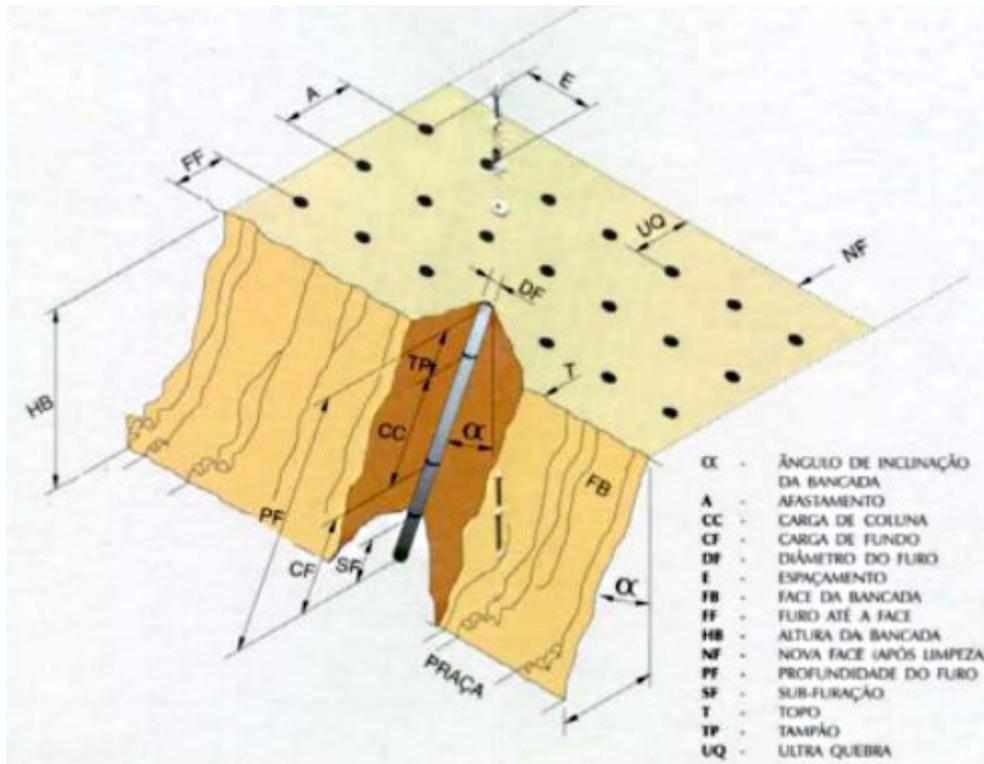


Figura 2: Bancada para desmonte de rocha com os parâmetros do plano de fogo.

Fonte: Manual Britanite, 2012.

2.2 Aprendizagem de Máquina

A aprendizagem de máquina pode ser vista como a área de pesquisa que tem como objetivo modelar e reproduzir de forma artificial a habilidade de aprendizagem de sistemas biológicos. Para tanto, esta área agrega conceitos de outros dois campos de pesquisa, a Inteligência Artificial e a Estatística. Em 1959, Arthur Samuel definiu aprendizagem de máquina como o "campo de estudo que dá aos computadores a habilidade de aprender sem serem explicitamente programados".

Nos primórdios da inteligência artificial a aprendizagem de máquina era bastante limitada de modo a não trazer benefícios evidentes à sociedade. Grande parte destas limitações deviam-se ao acesso restringido a dados em formato digital e ao baixo poder computacional disponível. Os modelos de aprendizado surgiram na era das mídias físicas, isto é, na época em que o principal meio de distribuição de conteúdo era através de VHS\CDs\DVDs, sendo difícil obter entradas realmente representativas para os algoritmos. Com o advento da internet tudo mudou. Uma quantidade massiva de informação trafega pela rede a cada minuto e com o auxílio da computação paralela e distribuída é possível treinar máquinas de aprendizagem para gerar modelos de boa representatividade.

Os algoritmos de aprendizagem são capazes hoje de resolver uma gama imensa de problemas, sejam eles de classificação, clusterização ou regressão. Estes algoritmos trabalham de forma diferente dos algoritmos tradicionais. Enquanto os algoritmos tradicionais são

inflexíveis por seguir puramente instruções estáticas o aprendizado automático permite extrair regularidades estatísticas dos dados, gerando um modelo que os represente. Assim sendo, os algoritmos de aprendizagem prezam por explorar o raciocínio indutivo, adaptando seus modelos de representação iterativamente de modo a minimizar o erro em relação a base de dados.

2.2.1 Aprendizagem de máquina supervisionada

Usualmente, algoritmos de aprendizado supervisionado tem por objetivo encontrar uma função preditora, isto é, uma função que represente a relação existente entre os atributos preditores e seus rótulos associados, a partir de um conjunto de dados representativos.

De modo formal, considere C um conjunto multidimensional com N exemplos de treinamento, os quais tem o seguinte formato: $C = \{(\rightarrow x_1, y_1), \dots, (\rightarrow x_N, y_N)\}$. Seja x_i um exemplo de entrada com valores de atributos $x_{ij} \in X$ e $y_i \in Y$ um valor associado. A aprendizagem de máquina supervisionada busca produzir uma função $g : X \rightarrow Y$, onde X é o espaço de entrada e Y é o espaço de saída. O espaço X é geralmente multidimensional e é também conhecido como vetor de características. Seus elementos podem ter as mais diversas naturezas, indo desde imagens e áudio, até texto propriamente dito. O espaço Y , de forma similar, apresenta esta mesma estrutura e flexibilidade, mas na maioria das vezes é unidimensional. Apesar de ser possível trabalhar com os mais diversos elementos de entrada e saída, estes devem ser convertidos de modo a serem corretamente interpretados pelos algoritmos de aprendizagem. Quando convertidos em dados categóricos, espaço discreto, tem-se a subárea da classificação ou reconhecimento de padrões. Por outro lado, quando são convertidos em escalares de valor real, espaço contínuo, tem-se a subárea chamada regressão.

Nesta seção serão apresentados os algoritmos de regressão alvos deste trabalho. São eles: *Support Vector Machine*, *Multilayer Perceptron* e *Reduced Error Pruning Tree*.

2.2.2 Support Vector Machine (SVM)

Support Vector Machine (SVM) compreende uma nova classe de algoritmos de aprendizado motivados pelos resultados da teoria de aprendizado estatístico (Vapnik, 1995). Estes algoritmos têm por objetivo encontrar uma fronteira de decisão que minimize a entropia entre as classes alvo. Formalmente, esta fronteira é conhecida como hiperplano de separação e sua estimativa é feita a partir de um subconjunto diminuto dos dados de treinamento, os vetores de suporte. Assim sendo, o hiperplano de separação é escolhido de modo a maximizar a distância entre suas margens de separação e os vetores de suporte.

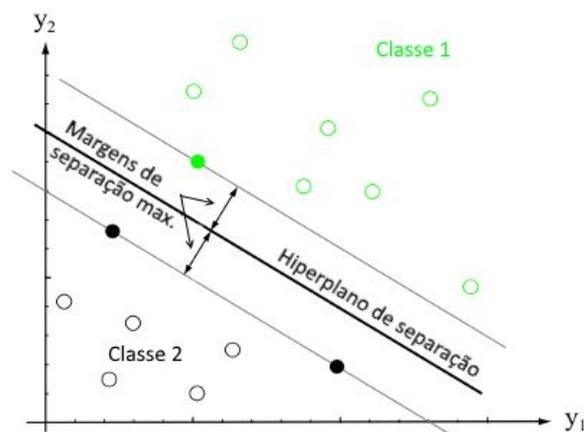


Figura 3: Hiperplano de separação. SVM.

Embora concebidas para fornecer uma separação linear dos dados, devido a sua natureza vetorial, as SVM podem se aproveitar de funções de kernel para extrapolar o espaço de dimensões dos dados. Esta propriedade conhecida como *Reproducing Kernel Hilbert Spaces* (RKHS) (Aronszajn, 1950; Girosi, 1997; Heckman, 1997; Wahba, 1990), diz que um produto interno no espaço de características possui um kernel equivalente no espaço de entrada,

$$K(x, x') = \langle \phi(x), \phi(x') \rangle, \quad (1)$$

desde que sejam obedecidas algumas restrições,

$$K(x, x') = \sum_m^{\infty} a_m \phi_m(x) \phi_m(x'), \quad a_m \geq 0, \quad (2)$$

$$\iint K(x, x') g(x) g(x') dx dx' > 0, \quad g \in L_2, \quad (3)$$

assim, o kernel representa de fato um produto interno no espaço de características. Deste modo é possível representar os dados em uma dimensionalidade que permita a adequada escolha de um hiperplano de separação.

As SVM foram propostas idealmente para problemas em espaços discretos, isto é, problemas de classificação, porém podem ser utilizadas para regressão através de um método chamado *Platt Scaling*. Este método consiste em um passo adicional ao processo de treinamento, onde os valores da regressão são estimados a partir da probabilidade de pertinência à classes. (John C. Platt, 1999).

Adicionalmente, devido à natureza do mecanismo de mensuração dos vetores de suporte, esta classe de algoritmos se mostra indiferente a presença de outliers porém, muito sensível a presença de ruído nos dados. Além disso, tende a ter seu custo computacional muito elevado quando da utilização em conjuntos de dados muito grandes, uma vez que exige operações de inversão de matrizes em seu processo de aprendizagem.

2.2.3 Multilayer Perceptron (MLP)

Este algoritmo de aprendizagem pertence à família dos algoritmos baseados em Redes Neurais Artificiais e surgiu como uma forma de superar as limitações do modelo de Perceptron Simples proposto Rosenblatt (1958).

O Perceptron simples, é um classificador linear binário. Ele recebe um conjunto de dados de entrada ($x_1...x_n$), os quais são multiplicados por um peso associado ($w_1...w_n$), que indica o quanto aquela entrada é significativa. A partir daí, o somatório destes produtos é submetido a uma função de ativação, de limiar (Θ), a qual produz o resultado da classificação (y).

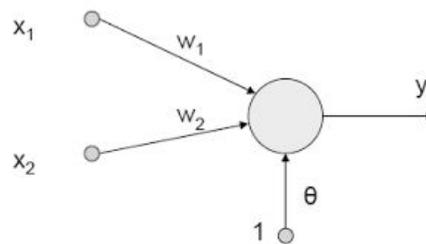


Figura 4: Modelo Perceptron Simples

Este tipo de técnica gera um hiperplano de separação, o que permite classificar os dados. Minsky e Papert (1969), porém, provaram que este modelo de Perceptron era incapaz de resolver problemas não linearmente separáveis, o que reduzia em muito a aplicabilidade deste algoritmo.

Rumelhart, Hinton e Williams (1986) encontraram a solução para este problema ao desenvolverem o algoritmo de treinamento *back-propagation*, capaz de treinar eficientemente redes neurais com mais de uma camada intermediária. Este algoritmo baseia-se na aprendizagem por retificação de erro via propagação e retropropagação.

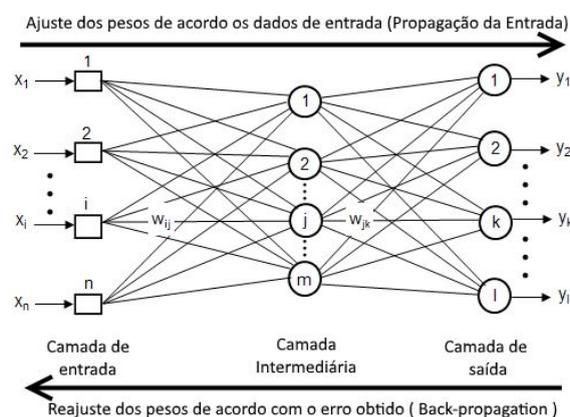


Figura 5: Modelo Multilayer Perceptron. Retropropagação.

De acordo com Negnevitsky (2005), durante a aprendizagem, um conjunto de formação de padrões de entrada é apresentado a MLP, que calcula o padrão de saída; caso haja diferença entre a saída da rede e a desejada, os pesos são ajustados para reduzir este erro.

A adição de mais camadas intermediárias permitiu a coexistência de múltiplos hiperplanos de separação, trabalhando em conjunto, tornando possível a resolução de problemas não linearmente separáveis. Sob a ótica de um problema de regressão uma rede neural multicamadas é capaz de aproximar quaisquer funções, desde que haja um número suficiente de neurônios na camada intermediária.

2.2.4 Reduced Error Pruning Tree (REPTree)

Reduced Error Pruning Tree é um rápido algoritmo de aprendizagem baseado do grupo das Árvores de Decisão. Sua construção baseia-se no ganho de informação ou na redução da variância. Segundo Mitchell, a árvore de decisão é um dos métodos de aprendizado simbólico mais amplamente utilizados e práticos para inferência indutiva. É um método para aproximar funções discretas robustas a dados com ruído e que permite o aprendizado de expressões disjuntas.

Similar ao algoritmo *Random Forest* (RF), o algoritmo REPTree utiliza regressão logística para criar múltiplas árvores em diferentes iterações. Porém apenas a árvore mais representativa é usada para a predição, diferenciando-se do RF, que utiliza todas as árvores para predição, em uma prática de comitê.

O principal aspecto deste algoritmo é a realização da poda da árvore como parte do processo de aprendizagem através do *Reduced Error Pruning*. Esta técnica reduz o tamanho da árvore de decisão através da remoção de sessões que promovem baixo ganho de predição (Kalmegh, 2015). Este processo de poda reduz a complexidade da árvore final, melhorando sua acurácia de predição devido a redução do overfitting.

2.3 Técnicas de avaliação de algoritmos de aprendizagem de máquina

Como é possível verificar a acurácia de um regressor? Dado dois regressores com desempenho de predição similar, qual deles possui maior generalização? A avaliação de regressores é, sobre tudo, uma atividade experimental. A abordagem mais tradicional, consiste em dividir o conjunto total de dados em 2 subconjuntos: treinamento e teste. Deste modo, a máquina de aprendizagem modela-se sobre o conjunto de treinamento e tem sua predição avaliada sob o conjunto de teste. Dentro deste contexto, algumas métricas comuns são o (MSE) Erro Médio Quadrático e o (MAE) Erro Médio Absoluto. Outra importante metodologia de avaliação é o teste de hipótese, o qual tem por objetivo verificar a superioridade estatística de um algoritmo sob outro.

2.3.1 Validação Cruzada K-Fold

Esta técnica consiste em dividir o conjunto total de dados em K subconjuntos disjuntos representativos. A partir deste ponto, o regressor é treinado com os dados de $K - 1$ conjuntos, e avaliado sob conjunto restante. Esta técnica é repetida K vezes, de modo que cada subconjunto tenha sido utilizado como subconjunto de teste uma vez. Afim de maximizar ainda mais o resultado desta validação, o procedimento acima descrito pode ser repetido N vezes, onde para cada repetição N , é feita uma subdivisão diferente dos dados dentre os K subconjuntos.

2.3.2 Paired T-Test

O teste estatístico T foi desenvolvido em 1908 pelo químico em cervejaria William Sealy Gosset. Embora desenvolvido com o intuito de monitoramento da qualidade de cervejas Stout, esta abordagem ganhou grande notoriedade após ser publicada na revista acadêmica *Biometrika*, em 1908.

O *T-Student* ou simplesmente *T-Test* é um teste de hipótese que utiliza uma distribuição *t-Student* para validação da recusa ou aceitação de uma hipótese nula. Esta distribuição é especialmente útil quando assume-se uma distribuição normal mas não se conhece a variância σ^2 , neste caso é utilizada uma variância amostral S^2 , e a estatística de teste passa a seguir uma distribuição *t-Student*. O *Paired T-Test*, no contexto de aprendizagem de máquina, é uma técnica para comparação de regressores par a par, que possibilita validar a superioridade estatística de um algoritmo, e posteriormente, a geração de um ranking.

3 METODOLOGIA DE PREDIÇÃO DO CUSTO OPERACIONAL

O sucesso no uso do aprendizado de máquina depende fortemente da seleção adequada do modelo de predição (Giraud-Carrier et al, 2004). Tendo em vista que na indústria de mineração não é comum a presença de um expert em aprendizagem de máquina na atividade de predição do OPEX, este trabalho buscou a implementação de uma plataforma user-friendly que possibilitasse o automático pré-processamento, experimentação, treinamento e predição do OPEX.

3.1 Pré-processamento

Bases brutas provindas da mineração tem grandes chances de conter dados inconsistentes ou matematicamente dispersos. O pré-processamento tem como objetivo tratar a base de forma a maximizar a acurácia das máquinas de aprendizagem. São empregadas nesta etapa as técnicas de Uniformização e Normalização, substituição de dados faltantes, além da Seleção de Atributos

Uniformização

A uniformização dos dados é o processo de redimensionar atributos numéricos de modo que eles venham a ter média zero e desvio padrão um. Este processo assume que o conjunto de dados tem uma distribuição Gaussiana. Este fato não precisa ser estritamente verdade, porém, em caso positivo, o resultado final será mais efetivo.

Para este problema, a uniformização foi aplicada ao atributo alvo, o OPEX.

Normalização

A normalização dos dados é o processo de redimensionar atributos numéricos proporcionalmente, entre zero e um. Desta forma, tem-se que o maior valor do atributo será representado por um, e o menor valor por zero. De modo geral, a normalização é uma boa abordagem quando a distribuição dos dados não é conhecida, ou quando sabe-se a priori que a distribuição de dados não é Gaussiana.

Para este problema, a normalização foi aplicada aos atributos preditores.

Substituição de dados faltantes

A substituição de dados faltantes é o processo de tratar dados faltantes em cada atributo do conjunto de dados. Os dados faltantes em um atributo são substituídos por sua média geral.

Embora esta técnica tenha sido redundante a priori, tendo em vista que não há dados faltantes na base sintética utilizada, ela é de fundamental importância quando da aplicação desta plataforma em uma base real.

Seleção de Atributos

Atributos sem representatividade estatística ou redundantes acabam por dificultar o processo de aprendizagem. A seleção de atributos é capaz de verificar a relevância dos atributos dispostos na base, eliminando aqueles que não contribuirão positivamente para o treinamento. Para tanto, a técnica empregada neste estudo utiliza *Backward-Elimination*.

O *Backward-Elimination* inicializa um subconjunto com todos os atributos preditores da base. Progressivamente, os atributos são removidos um a um, e o desempenho de predição resultante é avaliado. A menor combinação de atributos resultante, que mantiver o desempenho de predição é escolhida como resultado final.

A avaliação dos subconjuntos de atributos é feita a partir de um Grid de N configurações de *Support Vector Machines*. Deste modo a seleção de atributos é rodada N vezes. Após as N iterações a base gerada com menor número de atributos é a retornada para a execução das próximas etapas. O procedimento de seleção de atributos traz além de benefícios para a acurácia da predição, benefícios para o tempo de processamento, uma vez que reduz a dimensionalidade do conjunto de dados.

3.2 Experimentação

Na indústria de mineração não é comum a presença de um especialista em aprendizagem de máquina na atividade de predição do OPEX uma vez que esta predição é tradicionalmente feita através de séries temporais. Tendo em vista este cenário, se faz necessário uma etapa de experimentação capaz de selecionar o algoritmo e parâmetros mais adequados ao conjunto de dados.

Esta etapa faz o uso de um grid composto por configurações de máquinas de aprendizagem para regressão. Entenda por configuração uma tupla composta por um modelo e um conjunto de parâmetros válidos. Os modelos presentes são: *Support Vector Machine*, *Multilayer Perceptron* e *Reduced Error Pruning Tree*. Estes modelos são treinadas sob o conjunto de dados e submetidas a uma validação estatística através do *Paired T-Test*, onde a métrica alvo é o Erro Médio Absoluto. Este teste compara todas as configurações presentes dois a dois, e atribui a cada configuração o número de vitórias e derrotas válidas estatisticamente.

Com os resultados de cada configuração em mãos, é possível a geração de um Ranking, ordenado por suas pontuações. Estas pontuações são obtidas através da subtração entre o número de vitórias e o número de derrotas de cada configuração. Deste modo é possível definir qual configuração se comportou melhor diante das demais. Aquela que ficou melhor rankeada é a escolhida para a predição do OPEX.

3.3 Treinamento e Predição

Descoberta a configuração que melhor performa sob a base de dados é o momento de seu uso para a predição do OPEX. A melhor máquina de aprendizagem é treinada sob a base de dados, permitindo a predição a partir de dados não presentes no conjunto de treinamento.

Para a predição do valor do OPEX optou-se por utilizar máquinas de aprendizagem supervisionada, conforme descritas no capítulo anterior. Embora sejam utilizadas máquinas de aprendizagem distintas, o padrão dos dados de entrada e saída é o mesmo para todas.

input = [atributos numéricos preditores: $1, \dots, N$]

output = [atributo numérico alvo: OPEX]

No qual input é representado por um vetor de números reais, referentes aos parâmetros de lavra e o output por um número real. Além disso, cada atributo contém uma label descritiva associada.

3.4 Otimização

O tempo de computação assim como o uso eficiente dos recursos computacionais são aspectos primordiais quando do processamento de grandes bases de dados. Para atingir tais princípios foram realizadas as implementações seguir:

Paralelismo

A API de aprendizagem de máquina utilizada, o WEKA, não fornece suporte nativo à computação paralela, porém a implementação de paralelismo é ainda possível devido a natureza de execução da fase de Seleção de Atributos. Esta fase utiliza um grid de configurações, as quais são experimentadas de forma independente e deste modo, permitem ser executadas paralelamente.

O mecanismo de *Pool* foi escolhido para o gerenciamento das threads. Este mecanismo garante que um número N de threads estejam ativas por vez. Além disso, este valor N é mantido, ou seja, sempre que uma thread finaliza sua execução outra é inicializada, garantindo o número N de threads ativas simultaneamente. Este procedimento é realizado até que todas as threads endereçadas no *Pool* sejam executadas. Como o valor de N é configurado a partir da quantidade de threads disponíveis no hardware de execução, todo sua capacidade de processamento é utilizada.

Micro-Execução da Seleção de Atributos

Um grid com configurações diversificadas é fundamental para que a experimentação encontre um modelo que performe bem sob a base de dados. De todo modo, a execução de um grid muito extenso é custosa. Com o objetivo de reduzir o tamanho do grid da Seleção de

Atributos, é realizada uma micro-execução desta etapa. A micro-execução é feita sob um subconjunto reduzido do conjunto total de dados, porém, representativo. Após esta execução todas configurações que resultaram em um número de atributos maior que a média, ou que obtiveram um tempo de execução maior que a média são removidos do grid. Observe na tabela 1 a remoção das configurações.

Tabela 1: [ID da configuração] (Número de atributos resultantes, Tempo de processamento em segundos). Média dos atributos igual a 4. Média do tempo igual a 12. Configurações removidas em cinza. Razão da remoção em negrito.

[1] (9 , 6)	[2] (6 , 6)	[3] (4, 5)	[4] (4, 4)	[5] (4, 3)
[6] (8 , 4)	[7] (6 , 4)	[8] (4, 5)	[9] (4, 4)	[10] (4, 3)
[11] (6 , 4)	[12] (4, 6)	[13] (4, 5)	[14] (4, 4)	[15] (4, 3)
[16] (6 , 6)	[17] (4, 22)	[18] (4, 9)	[19] (4, 5)	[20] (4, 3)
[21] (4, 37)	[22] (4, 158)	[23] (4, 31)	[24] (4, 11)	[25] (4, 9)
[26] (7 , 2)	[27] (6 , 1)	[28] (5 , 3)	[29] (6 , 3)	[30] (6 , 4)

Um dos problemas do uso do *Pool* de threads é que este pode ficar abaixo de sua capacidade máxima de execução simultânea quando estiver executando as últimas threads. Mais formalmente, seja T o número total de threads a serem processadas e N o número máximo de threads simultâneas no pool. $T \bmod N$ threads serão inevitavelmente processadas subutilizando o número máximo de threads simultâneas. Esta situação torna-se grave quando estas threads tem longa duração, tendo em vista que o pool só é finalizado quando todas as threads endereçadas são processadas.

Para minimizar este problema, as configurações do grid são escalonadas de modo que aquelas com tempo de execução mais longo sejam executadas primeiro pelo pool, minimizando seu tempo total de execução.

4 RESULTADOS E ANÁLISE

Nesta sessão os resultados obtidos sob a base sintética utilizada são apresentados. A obtenção destes resultados seguiu a metodologia descrita previamente no Capítulo 3. Para a experimentação, a abordagem utilizada conste em uma validação cruzada de 10 *folds*, onde para cada fold foi verificado o Erro Médio Absoluto, além de seu desvio padrão. O teste de hipótese *Paired T-Test* foi também empregado para validação estatística dos resultados. O resultado final exposto consiste na média dos resultados individuais de cada fold.

Um explanação a cerca das tecnologias empregadas será feita a fim de levantar as características que culminaram em sua escolha, bem como um detalhamento da base de dados utilizada.

4.1 Java

Java é uma linguagem de programação interpretada e orientada a objetos. Seu desenvolvimento começou em meados da década de 90, quando uma equipe de desenvolvedores se reuniram para propor uma linguagem portátil, capaz de funcionar independente de plataforma. Esta equipe era composta por uma série de programadores e chefiada por James Gosling, na empresa Sun Microsystems.

Para obter este desacoplamento em relação à plataforma, Java não é compilada diretamente para código nativo, em vez disso, ela é compilada para um código intermediário, o bytecode, que é interpretado por uma máquina virtual (Java Virtual Machine - JVM) específica para cada plataforma. As principais características desta linguagem são:

- Portabilidade: *Write once, run anywhere*. Escreva uma vez, execute em qualquer lugar.
- Sintaxe similar às linguagens C/C++/C#.
- Desalocação automática de memória através do uso de um *Garbage Collector*.
- É distribuída por padrão com um vasto conjunto de bibliotecas.

Java foi totalmente liberada como software livre, através da licença GPL (General Public Licence) em maio de 2007. No ano seguinte a Oracle fez a aquisição da Sun, proponto expandir a linguagem para outros produtos, para benefício geral do consumidor.

Weka

O Weka (*Waikato Environment for Knowledge Analysis*), é uma famosa biblioteca escrita sob a linguagem java, voltada para o estudo de aprendizagem de máquina. Seu desenvolvimento teve início em 1993 na Universidade de Waikato na Nova Zelândia. Atualmente o Weka, assim como Java, possui a licença GPL (*General Public Licence*), o que permite o estudo e modificação de seu código fonte.

De modo geral, o Weka tem por objetivo reunir em um ambiente seguro e coeso os algoritmos resultantes da área de aprendizagem de máquina, permitindo suas execuções, análises e validações. Seu portfólio é bastante completo, contendo os principais algoritmos da área em estado da arte, assim como abordagens de avaliação e validação estatísticas. Desta biblioteca foram utilizados os algoritmos de aprendizagem *MultiLayer Perceptron* e *Reduced Error Pruning Tree*, além do teste de hipótese *Paired-T Test*.

LibSVM

O LibSVM é uma famosa biblioteca de algoritmos *Support Vector Machine*. Seu desenvolvimento teve início em 2000 na Universidade Nacional de Taiwan. O objetivo principal era melhorar a eficiência de predição e facilidade de uso de SVMs quando de sua aplicação para classificação e regressão. Esta biblioteca está disponível para as linguagens Java, Matlab e C, sendo aplicada sob Java neste projeto.

4.2 Base de Dados

A base de dados utilizada neste trabalho é sintética. Seus dados foram gerados a partir de um simulador geológico voltado para mineração, e representam uma atividade de lavra por bancadas. Sua composição compreende 2159 instâncias, onde cada uma possui 17 atributos preditores e 1 atributo alvo. Estes atributos preditores podem ser divididos em 2 subgrupos: atributos do Plano de Fogo e atributos Financeiros.

Os atributos do Plano de Fogo são relativos ao processo de detonação para extração dos minérios, descritos em mais detalhes na seção 2.1.1. Estes atributos estão dispostos na tabela 3. Os demais atributos são financeiros e estão dispostos na tabela 2. O atributo alvo, a saber o OPEX, possui média 445418279, desvio padrão 405425612 e é expresso em dólares.

Tabela 2: Atributos preditivos financeiros, média e desvio padrão.

Preço do explosivo	4,10 (0,04)	R\$/kg
Preço do iniciador curto	20,4 (0,00)	R\$/ud
Preço do iniciador longo	23,15 (0,00)	R\$/ud
VODe	4402,91 (99,98)	n/a
VODn	6184,66 (159,96)	n/a

Tabela 3: Atributos preditivos relativos ao Plano de Fogo, média e desvio padrão.

Altura da Bancada	13,50 (3,20)	m
Subfuração	1,01 (0,40)	m
Diâmetro do Furo	0,23 (0,07)	m
Inclinação do Furo	84,92 (5,00)	Graus
Afastamento	6,99 (2,23)	m
Espaçamento	5,98 (1,64)	m
Comprimento de carga Inferior	7,76 (1,93)	m
Comprimento de carga superior	0,48 (0,41)	m
Desvio padrão da precisão de perfuração	0,50 (0,00)	m
Força relativa dos explosivos	124,11 (3,99)	%
Densidade dos explosivos	1125,72 (24,99)	km/m ³
Comprimento total da carga	0,23 (0,07)	m

4.3 Parâmetros do Experimento

O ambiente de aprendizagem desenvolvido é alimentado por um Grid de configurações em duas de suas fases: Seleção de Atributos e Experimentação. Na fase de seleção de atributos o grid é composto por configurações do algoritmo SVM, enquanto que na fase de experimentação são presentes adicionalmente os algoritmos MLP e REPTree.

Support Vector Machine

A biblioteca LibSVM oferece uma grande variedade de parâmetros de configuração. Escolhemos o kernel RBF (*Radial Basis Function*), os parâmetros variados dentro do grid estão dispostos abaixo, os demais parâmetros foram utilizados com seus valores padrão.

- Tipo da SVM: epsilon-SVR e nu-SVR (SVMs para regressão)
- Gamma da função de kernel: 0.001, 0.01, 0.1, 0.5 (padrão), 1.0
- Parâmetro nu: 0.1, 0.25, 0.5 (padrão), 0.75, 1.0
- Parâmetro C: 1.0 (padrão), 10.0, 100.0, 1000.0, 10000.0

Totalizando de 31 configurações.

Multilayer Perceptron

O algoritmo utilizado para implementação de redes neurais foi o *Multilayer Perceptron*, disponível na biblioteca de aprendizado Weka. Os parâmetros variados dentro do grid estão dispostos abaixo, os demais parâmetros foram utilizados com seus valores padrão.

- Taxa de aprendizado do algoritmo *Backpropagation*: 0.001, 0.01, 0.1, 0.3 (padrão), 0.5
- Número de camadas escondidas criadas para a rede: 2, 10, 25, 50, 100.

Totalizando 25 configurações.

Reduced Error Pruning Tree

O *Reduced Error Pruning Tree* é um algoritmo baseado em árvore de decisão e pertencente à biblioteca de aprendizado Weka. Os parâmetros variados dentro do grid estão dispostos abaixo, os demais parâmetros foram utilizados com seus valores padrão.

- Número mínimo de instâncias por folha: 2 (padrão), 4, 8, 16, 32, 64, 256.
- Variância mínima da classe para divisão em ramos: 0.001 (padrão), 0.01, 0.1

Totalizando 21 configurações.

4.4 Análise dos dados

4.4.1 Seleção de Atributos

A fase de seleção de atributos remove atributos preditores pouco representativos a fim de diminuir a complexidade da base de dados, melhorando a acurácia de predição e diminuindo o tempo de treinamento. Alguns dos atributos claramente não apresentam ganho de informação por serem atributos constantes, são eles Desvio padrão da precisão de perfuração, Preço do iniciador curto e Preço do iniciador longo. Além destes 3, a seleção de atributos removeu mais 11 atributos, selecionando apenas 3 atributos preditores para representar toda a base. Estes atributos foram Altura da Bancada, Afastamento e Espaçamento.

Conforme descrito na seção 2.1.1 estes atributos são relativos ao tamanho e arranjo da bancada. Quanto maior a altura da bancada e menor o afastamento e espaçamento dos furos maior é o custo de detonação, tendo em vista que serão necessários mais explosivos. Deste modo não é de grande surpresa estes atributos serem os mais representativos tendo em vista que o atributo alvo é o Custo Operacional. Revise a figura 7 para uma melhor consolidação deste cenário.

4.4.2 Micro-Execução da Seleção de Atributos

A fase de seleção de atributos padrão gasta em média 80% do tempo total de computação da plataforma. Com o objetivo de minimizar o tempo de execução desta fase foi implementada uma Micro-Execução da mesma, conforme detalhado na seção 3.3. Esta implementação trouxe uma economia de 89% do tempo total de computação tornando a atividade de predizer o OPEX 9,3 vezes mais rápida em relação ao mecanismo padrão.

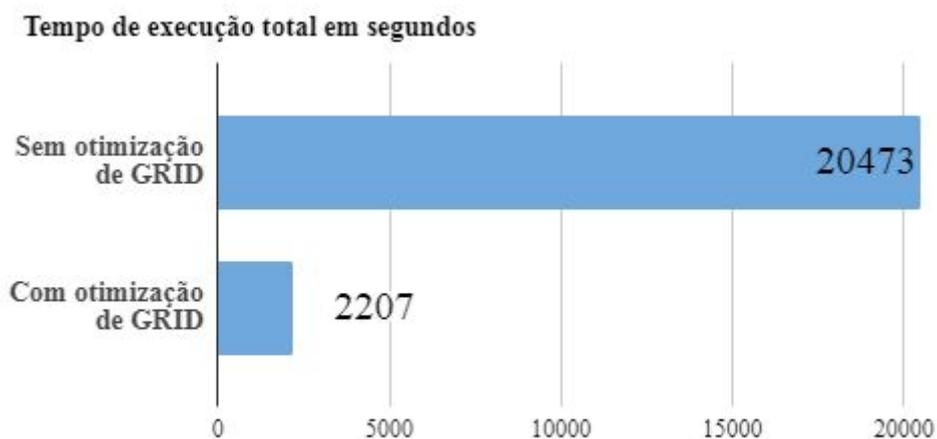


Figura 6: Gráfico do tempo total de execução antes e após a implementação da otimização via Micro-Execução da fase Seleção de Atributos.

Seguem as configurações da máquina de testes: Intel Core I3 4160 @3,60GHz (2 núcleos físicos/4 threads), 8GB RAM DDR3 @1333MHz, Windows 10 x64 release 1607.

4.4.3 Acurácia de predição

Afim de maximizar a acurácia de predição e tornar a plataforma robusta quando da predição sob diferente bases de dados foi desenvolvida a etapa de Experimentação. Conforme descrito na seção 3.2, esta etapa compara estatisticamente acurácia de 71 configurações dos algoritmos SVM, MLP e REPTree levando em consideração o MAE (erro médio absoluto).

A tabela 4 mostra o MAE médio para cada tipo de algoritmo presente no Grid da fase de experimentação. Neste grid estão presentes 31 configurações de SVM, 25 configurações de MLP e 21 configurações de REPTree. Nesta abordagem avaliativa o algoritmo que obteve melhor acurácia de predição foi o *Reduced Error Pruning Tree*, apresentando o menor MAE e menor desvio padrão associado.

Este resultado embora possa parecer surpreendente é justificado quando tem-se em mente a natureza da base de dados: sintética. É natural algoritmos baseados em Árvores de Decisão performarem bem sob base de dados sintéticas, pois seu mecanismo de predição

consiste em definir sub-funções preditivas para os intervalos de decisão da árvore. Desta mesma forma trabalha um gerador de base de dados.

Em contrapartida, diante do cenário descrito, este baixo erro de predição tem grande chance de ser um forte indicador de overfitting. Em outras palavras, é grande a probabilidade do algoritmo ter decorado a base de dados, prejudicando sua generalização.

Tabela 4: MAE e desvio padrão médio obtido por cada tipo de algoritmo de aprendizagem presente no grid.

Support Vector Machine	0,0944 (0,0035)
MultiLayer Perceptron	0,1136 (0,0477)
Reduced Error Pruning Tree	0,0611 (0,0038)

Afim de validar estatisticamente a superioridade de predição dentre as configurações presentes no grid o teste de hipótese Teste-T Pareado foi utilizado (seção 2.3.2). Esta abordagem gera um ranking que possui os seguintes indicadores: Pontuação, Número de Vitórias estatísticas, Número de derrotas estatísticas, onde a Pontuação é o resultado da subtração entre os dois últimos indicadores citados. As 10 melhores configurações estão dispostas na tabela 5.

Tabela 5: Ranking das 10 melhores configurações do grid e seus respectivos índices

>-<	>	<	Configuração
70	70	0	SVM (26)
65	65	0	MultilayerPerceptron (36)
65	65	0	MultilayerPerceptron (35)
63	64	1	REPTree (77)
63	64	1	REPTree (76)
63	64	1	REPTree (75)
62	62	0	MultilayerPerceptron (41)
56	62	6	SVM (1)
53	60	7	SVM (6)
50	52	2	MultilayerPerceptron (45)

A configuração que obteve melhor predição em relação a todas as outras foi a de índice 26, uma SVM. Esta configuração foi capaz de vencer estatisticamente 70 das 77 configurações do GRID, e não perdeu para nenhuma delas. A tabela 6 contém o MAE e seu desvio padrão para a configuração melhor ranqueada de cada tipo de algoritmo.

Tabela 6: MAE e desvio padrão das configurações melhores ranqueadas, por algoritmo.

Support Vector Machine	0,2403 (0,0050)
MultiLayer Perceptron	0,2135 (0,0640)
Reduced Error Pruning Tree	0,1513 (0,0087)

Por fim, a suspeita de overfitting por parte das configurações baseadas no algoritmo REPTree pôde ser confirmada. Embora possua MAE e desvio padrão médios bem menores que a configuração SVM (26), esta obteve estatisticamente uma melhor generalização sob o conjunto de dados, sendo a escolhida para predição do OPEX.

5 CONSIDERAÇÕES FINAIS

O objetivo principal deste trabalho foi fornecer uma plataforma para a predição do Custo Operacional (OPEX) voltada à mineração utilizando aprendizagem de máquina. Para tanto esta plataforma deve ser robusta e automática, de modo a não exigir um especialista para seu uso e ainda assim performar bem sobre bases de dados diversas.

Para tal, foram implementadas etapas de pré-processamento da base de dados, seleção de atributos, experimentação, treinamento e predição. Adicionalmente foi também implementada uma micro-execução da seleção de atributos com o objetivo de reduzir o tempo de computação da plataforma.

As abordagens de tratamento dos dados e experimentação foram eficientes em lidar com a base de dados utilizada. Os três algoritmos experimentados, a saber SVM, MLP e REPTree, foram capazes de performar bem sobre a base de dados, com a ressalva para este último, por apresentar overfitting. Adicionalmente, a micro-execução da seleção de atributos foi capaz de economizar tempo substancial de computação, sem com isso prejudicar a acurácia de predição da plataforma.

5.1 Trabalhos futuros

Alguns possíveis trabalhos futuros são:

- Usar bases de dados reais para validação.
- Usar meta-aprendizado em substituição a etapa de experimentação.
- Usar multi-objetivo para maximização da acurácia e minimização do tempo de execução.
- Utilizar técnicas como Deep Learning para a predição do OPEX.

6 REFERÊNCIAS

NUNES, E., ORGIVEL, J.: Programa UNIDIS, Geografia Física 1, **Minerais e rochas** (2009)

LIMA, R.: **Mineração no Brasil**, Instituto Brasileiro De Mineração (2003)

CURI, A.: **Lavra de Minas**. Oficina de Textos (2017)

GRANDELL, J.: **Time series analysis**. Lecture notes (2011)

SIMINERAL: História da Mineração.

Disponível em: <http://simineral.org.br/mineracao/historia>

GIRODO, A. C.: **Mineração: Projeto Apa Sul RMBH – Estudos do Meio Físico**. v. 2. Belo Horizonte, 168 p (2005)

BRITANITE: **Manual - Uso Explosivos Britanite** (2012)

VAPNIK, V., CORTES C.: **Support-Vector Networks**, Machine Learning, 20, 273-297 (1995)

GUNN, R.: **Support Vector Machines for Classification and Regression**. Technical Report (1998)

PLATT, J.: **Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods**. Microsoft Research (1999)

ROSENBLATT, F.: **The Perceptron: A Probabilistic Model For Information Storage And Organization In The Brain**. Psychological Review Vol. 65, No. 6 (1958)

MINSKY, M., PAPERT, S.: **Perceptrons: an introduction to computational geometry**. (1969)

RUMELHART, D., HINTON, G., WILLIAMS, R.: **Learning representations by back-propagating errors** (1986)

NEGNEVITSKY, M.: **Artificial Intelligence A Guide to Intelligent Systems**. Second Edition (2005)

MITCHELL, T.: **Machine Learning**, McGraw-Hill (1997)

KALMEGH, S.: Analysis of WEKA Data Mining Algorithm REPTree, Simple Cart and RandomTree for Classification of Indian News (2015)

GIRAUD-CARRIER, C., VILALTA, R., BRAZDIL, P.: Introduction to the special issue on meta-learning. Machine Learning 54(3), 187–193 (2004)

Java Brasil. Disponível em: <https://www.java.com/>

LIBSVM: A Library for Support Vector Machines.

Disponível em: <https://www.csie.ntu.edu.tw/~cjlin/libsvm/>

WEKA: Waikato Environment for Knowledge Analysis.

Disponível em: <https://www.cs.waikato.ac.nz/ml/weka/>