UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO DE INFORMÁTICA

JELSON ANDRE CORDEIRO

MACHINE LEARNING APLICADO NO PROBLEMA DE PERDAS COM CRÉDITOS DE UMA DISTRIBUIDORA DE ENERGIA ELÉTRICA

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA
2021

JELSON ANDRE CORDEIRO

MACHINE LEARNING APLICADO NO PROBLEMA DE PERDAS COM CRÉDITOS DE UMA DISTRIBUIDORA DE ENERGIA ELÉTRICA

Machine Learning Applied to the Problem of Doubtful Accounts of a Electric Energy Utility Company

Trabalho de Conclusão de Curso apresentado(a) como requisito para obtenção do título(grau) de Especialista em Ciência de Dados e suas Aplicações, do Departamento de Informática, da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador(a): Prof(a). Dr(a). Marcelo de Oliveira Rosa

CURITIBA

2021

Ministério da Educação
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
UTFPR - CAMPUS CURITIBA
DIRETORIA-GERAL - CAMPUS CURITIBA
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO - CAMPUS CURITIBA
DEPARTAMENTO DE APOIO DAS ESPECIALIZAÇÕES LATO-SENSU DOS CURSOS DE INFORMÁTICA - CAMPUS CURITIBA
CURSO DE ESPECIALIZAÇÃO EM CIÊNCIA DE DADOS E SUAS APLICAÇÕES

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

TERMO DE APROVAÇÃO

MACHINE LEARNING APLICADO NO PROBLEMA DE PERDAS COM CRÉDITOS DE UMA DISTRIBUIDORA DE ENERGIA ELÉTRICA

por

Jelson Andre Cordeiro

Este Trabalho de Conclusão de Curso foi apresentado às 19h00 do dia 19 de Julho de 2021 por videoconferência como requisito parcial à obtenção do grau de Especialista em Ciência de Dados e suas Aplicações na Universidade Tecnológica Federal do Paraná - UTFPR - Campus Curitiba. O aluno foi arguido pela Banca de Avaliação abaixo assinados. Após deliberação, a Banca de Avaliação considerou o trabalho aprovado.

_____

Prof. Dr. Marcelo de Oliveira Rosa (Presidente/Orientador – DAELT-CT/ UTFPR-CT)

_____

Prof. Dr. Leandro Miranda Zatesk (Avaliador 1– DAINF-CT/ UTFPR-CT)

_____

Profa. Dra. Rita Cristina Galarraga Berardi (Avaliadora 2 – DAINF-CT/ UTFPR-CT)

O Termo de Aprovação assinado encontra-se no sistema SEI- Processo nº 23064.028109/2021-03

RESUMO

CORDEIRO, Jelson A.. **Machine Learning Aplicado no Problema de Perdas com Créditos de uma Distribuidora de Energia Elétrica**. 2021. 36 f. Trabalho de Conclusão de Curso (Especialização em Ciência de Dados e suas Aplicações) – Universidade Tecnológica Federal do Paraná. Curitiba, 2021.

As Perdas Estimadas em Créditos de Liquidação Duvidosa nas empresas é um campo atraente para investigação devido ao percentual dos lucros que representa. O objetivo deste trabalho é encontrar um modelo de aprendizagem de máquina para predizer em que dia o cliente irá pagar a fatura visando maximizar o lucro da empresa. Para avaliar a metodologia proposta foram realizados experimentos utilizando dados reais de faturas dos clientes. Os resultados dos modelos foram comparados entre si e realizado a análise estatística para verificar se existe diferença significativa entre eles. Os resultados alcançados indicam que é promissora a aplicação da modelagem proposta.

Palavras-chave: Aprendizagem de Máquina Supervisionado. Regressão Linear. Perdas Estimadas em Créditos.

ABSTRACT

CORDEIRO, Jelson A.. **Machine Learning Applied to the Problem of Doubtful Accounts of a Electric Energy Utility Company**. 2021. 36 p. Trabalho de Conclusão de Curso (Especialização em Ciência dos Dados e suas Aplicações) – Universidade Tecnológica Federal do Paraná. Curitiba, 2021.

Allowance for Doubtful Accounts (AFDA) in companies is an attractive field for investigation due to the percentage of profits it represents. The objective of this work is to find a machine learning model to predict which day the customer will pay the invoice in order to maximize the company's profit. To evaluate the proposed methodology, experiments were carried out using real data from customer invoices. The results of the models were compared with each other and a statistical analysis was carried out to verify if there was a significant difference between them. The results indicate that it is promising to apply the proposed model to the problem of Estimated Losses on Loan Losses.

Keywords: Supervised Machine Learning. Linear Regression. Estimated Losses on Loan Losses.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS, SIGLAS E ACRÔNIMOS

SIGLAS

CD	*Critical Distance*
LGPD	Lei Geral de Proteção de Dados
LIME	*Local Interpretable Model-agnostic Explanations*
MAE	*Mean Absolute Error*
ML	*Machine Learning*
PCA	*Principal Components Analysis*
PECLD	Perdas Estimadas em Créditos de Liquidação Duvidosa
QII	*Quantitative Input Influence*
RMSE	*Root Mean Square Error*
RNA	*Rede Neural Artificial (RNA)*
SHAP	*SHapley Additive exPlanations*
SMOTE	*Synthetic Minority Over-sampling Technique*
SVM	*Support Vector Machine*
UC	*Unidade Consumidora*

SUMÁRIO

1 INTRODUÇÃO

Todo planejamento financeiro de uma empresa é baseado nos créditos e débitos. Desse cenário, surge a necessidade de se realizar a chamada Perdas Estimadas em Créditos de Liquidação Duvidosa (PECLD) (PADOVEZE, 1996). Medidas como a PECLD são fundamentais para a empresa não se endividar por falta de precaução. A PECLD se refere a uma reserva de dinheiro feita pela empresa com foco em casos de inadimplência.

Dessa forma, quanto maior for o risco de o cliente não pagar o que deve, maior deve ser o montante guardado pela empresa através da PECLD.

Nas empresas de Distribuição de Energia Elétrica no Brasil e no mundo o problema de PECLD representa um valor significativo no lucro. Em 2019, o PECLD de umas das maiores empresas deste setor foi de R$ 137,75 milhões que representou 20,87% do lucro daquele período (http://sistemas.cvm.gov.br/).

Também quando uma fatura de energia é gerada, com ela também são gerados impostos e tributos, que são repassados para os entres tributantes, seja ela arrecadada ou não. Se o consumidor não quita a fatura, a empresa precisa pagar os impostos do seu próprio bolso.

Outro custo que a empresa possui quando o consumidor não quita a fatura é o custo de procedimento de desligamento da unidade consumidora. Após a geração da fatura e caso o consumidor não tenha quitado dentro do prazo limite, é iniciada uma série de procedimentos. Estes procedimentos vão desde envio de e-mails/SMS, contato telefônico, inclusão no Serviço de Proteção ao Crédito (SPC), desligamento, cobrança personalizada e cobrança terceirizada. Além destes custos, a escolha de qual cliente cortar primeiro é importante porque os recursos que a empresa tem (equipe em campo disponível, por exemplo) são limitados.

Portanto, para resolver este tipo de problema das Distribuidoras de Energia Elétrica o objetivo geral é buscar um modelo de *Machine Learning* (ML) capaz de predizer o dia em que o cliente pagará a fatura visando maximizar o lucro da empresa.

1.1 OBJETIVOS

O objetivo geral deste trabalho é encontrar um modelo de ML supervisionado de regressão capaz de predizer o dia em que o cliente pagará a fatura visando maximizar o lucro da empresa.

Os objetivos específicos são:

1. Definir quais algoritmos utilizar para encontrar o modelo;
2. Avaliar o desempenho dos modelos com experimentos utilizando dados históricos reais de uma Distribuidora de Energia Elétrica;
3. Fazer uma análise estatística apropriada dos resultados comparativos dos modelos para verificar se existe diferença significativa.

Este trabalho está organizado da seguinte forma: no Capítulo 2 é apresentada a fundamentação teórica, ou seja, uma pesquisa dos temas envolvidos e trabalhos realizados. No Capítulo 3 é apresentada a aplicação dos algoritmos. No Capítulo 4 são apresentados e analisados os resultados obtidos. No Capítulo 6 é apresentada a conclusão e sugestões de trabalhos futuros.

2 REVISÃO DA LITERATURA

Existe um grande número de algoritmos específicos de aprendizado de máquinas desenvolvidos ao longo dos anos para tratar alguns tipos de tarefas fundamentais.

P e Srinivasan (2011) utilizaram o aprendizado supervisionado para estimar se o cliente irá pagar ou não a fatura em uma base de 150.000 registros. *Support Vector Machine* (SVM) e *Random Forest* foram utilizados na qual esse teve o melhor desempenho com 86% de *score*. Para encontrar os melhores *hyperparameters* foi utilizado a validação cruzada. Bastos (2010) utilizou *Random Forest* para prever perda de crédito no setor bancário. Neste trabalho a avaliação de desempenho foi divido em horizontes de 12, 24, 36 e 48 meses. (YAZDI *et al.*, 2020) utilizou SVM para identificação de *Fake news*. (PAHWA; AGARWAL, 2019) utilizou SVM para análise do mercado de ações. (LUNDBERG *et al.*, 2018) utilizou o XGBoost, Lasso e SVM para prevenir hipoxemia durante uma cirurgia.

Além dos algoritmos citados, entender como um modelo estatístico fez uma previsão específica é um desafio importante no aprendizado de máquina. No entanto, muitos modelos complexos com excelente precisão, fazem previsões que até mesmo os especialistas lutam para interpretar. Isso força uma compensação entre precisão e interpretação. Entender como o modelo de chegou a determinado resultado traz diversos benefícios: Primeiro a necessidade de transparência dos modelos nas organizações cresceu na medida em que usam grandes volumes de informações pessoais e complexas (Datta *et al.*, 2016). É essencial para empresa permitir a identificação de danos, como discriminação, introduzido pela tomada de decisão do modelo de ML (por que o modelo chegou a conclusão que determinado cliente não vai pagar a conta e o outro vai). Em segundo lugar, a transparência pode ajudar detectar erros nos dados entrada que resultaram em uma decisão errada pelo modelo.

Uma dos algoritmos é o *SHapley Additive exPlanations* (SHAP) (LUNDBERG; LEE, 2017; LIPOVETSKY; CONKLIN, 2001), no qual o impacto de cada atributo no modelo é representado usando valores de Shapley. É uma abordagem para explicar a saída de qualquer modelo de aprendizado de máquina que conecta a alocação de crédito ideal com explicações locais usando os valores clássicos de Shapley da teoria dos jogos e suas extensões relacionadas. Este algoritmo utiliza a média das contribuições marginais em todas as permutações. Lundberg *et al.* (2018) utilizou para explicar para médicos o resultado do modelo encontrado para prever hipoxemia durante a cirurgia. Outra abordagem também encontrada na literatura é o *Quantitative*

Input Influence (QII) (Datta *et al.*, 2016), que quebra as correlações entre as entradas para permitir o raciocínio causal e calcula a influência marginal de entradas em situações na qual as entradas não podem afetam apenas os resultados. *DeepLIFT* (SHRIKUMAR *et al.*, 2017) é um algoritmo de interpretação para modelos de *Deep Leraning* que utiliza um método para decompor a previsão de saída de uma rede neural propagando as contribuições de todos os neurônios da rede para cada recurso da entrada. Shrikumar *et al.* (2017) utilizou o *DeepLIFT* em simulação de genoma. E por fim, o *Local Interpretable Model-agnostic Explanations* (LIME) (RIBEIRO *et al.*, 2016), que também é um algoritmo para explicar as previsões de qualquer modelo em uma forma interpretável.

3 MATERIAL E MÉTODOS

Para a implementação da metodologia proposta, alguns aspectos devem ser definidos. Os dados utilizados (ou dados de origem) são apresentados na seção 3.1. Na seção 3.2 são definidos quais algoritmos serão utilizados e seus respectivos *hyperparameters*. A metodologia de avaliação será apresentada na seção 3.3 e na seção subseção 3.3.1 como será realizada a comparação entre os modelos para verificar se existe diferença significativa entre eles.

O trabalho seguiu o processo de mineração de dados CRISP-DM (PROVOST *et al.*, 2018), em que foi necessário voltar diversas vezes para o processo de preparação dos dados antes de inserir os dados no modelo novamente.

3.1 DADOS DE ORIGEM

Serão utilizados dados históricos reais de faturas de clientes de uma Distribuidora de Energia Elétrica. Para satisfazer a Lei Geral de Proteção de Dados (LGPD) atributos que pudessem identificar o cliente foram retirados do *dataset* inicial. O *dataset* inicial contêm 220.218 faturas com 9418 clientes distintos e cada um deles com 23 faturas em média. Este *dataset* possui 40 atributos preditores e um atributo alvo (“dias em atraso”). O atributo alvo está balanceado, na qual 59% são faturas com atraso. Caso não estivesse balanceado poderia ser utilizado alguma técnica de balanceamento como o *Synthetic Minority Over-sampling Technique* (SMOTE) (CHAWLA *et al.*, 2002) ou *Oversampling*.

Dentre os atributos preditores podem-se destacar alguns de maior relevância como o valor da fatura, se o cliente esta negativado no Serasa, valor total de débito que o cliente possui e quantidade de faturas em aberto para o mesmo cliente. Idade, sexo, classe, tipo da pessoa (PF/PJ), indicativo de cliente VIP, indicativo de cliente baixa renda, dias que o cliente retornou contato após SMS e cortes gerados também merecem destaques. Outros atributos menos relevantes utilizados foram: tipo da ligação (monofásica, bifásica ou trifásica), tensão e localidade.

Os atributos valor total de débito e quantidade de faturas em aberto foram criados na etapa de *Feature Engineering*. Os atributos tensão, localidade, sexo, tipo da pessoa e classe sofreram o processo de *OneHotEncoder* (LAKSHMANAN *et al.*, 2020) no qual, por exemplo, o atributo classe que possui três valores possíveis (Residencial, Comercial e Industrial) se transformaram em três atributos binários classeRE, classeCO e classeIN. Segundo Lakshmanan

et al. (2020) realizando este processo melhora o desempenho dos algoritmos, pois caso tivesse valores numéricos 1 (Industrial), 2 (comercial) e 3 (residencial) para as classes o algoritmo pode achar que residencial é maior ou melhor que os outros o que não faz sentido no mundo real.

Neste trabalho, 176.174 faturas (80% dos dados) foram reservadas para treinamento para garantir assertividade do modelo e garantir que esta sendo coberta a maioria dos casos da população de 44.044 faturas (20% dos dados) para os testes.

3.2 ALGORITMOS

Existe um grande número de algoritmos específicos de *machine learning* desenvolvidos ao longo dos anos para tratar alguns tipos de tarefas fundamentais. Este trabalho utiliza a tarefa de regressão (RUSSELL; NORVIG, 2004) que tenta predizer o valor numérico de alguma variável alvo. No caso presente trabalho esta variável é em quantos dias o consumidor irá pagar a fatura.

Além da regressão linear simples, que é o método de aprendizagem de máquina para descrever e quantificar a relação entre a variável alvo e as variáveis preditoras (PIERSON; MACHADO, 2019), foram utilizados o Lasso e Ridge que foram criados para melhorar a previsão em relação a regressão linear simples (TAN *et al.*, 2009). O Lasso utiliza o *hyperparameters* L1 para penalizar os coeficientes, reduzindo o coeficiente para zero, e a regressão Ridge utiliza o *hyperparameters* L2 para amortecer o coeficiente mas não força para zero. A diferença básica entre os dois algoritmos é que o Lasso pode retirar do modelo atributos preditores insignificantes enquanto o Ridge mantém todos os atributos por que o coeficiente nunca chega a zero.

O algoritmo de *Random Forest* também foi utilizado, que é um método de aprendizagem supervisionado que constrói uma grande coleções de árvores descorrelacionadas ao contrário das Árvores de Decisão (BREIMAN *et al.*, 1984; QUINLAN, 1986) que cria apenas uma árvore. Esta coleção de árvores criadas pelo *Random Forest* permite uma generalização maior do modelo evitando inclusive o *overfitting*. No *Random Forest* é buscado o valor que cada uma das árvores de decisão indicou. Esta técnica é chamada de aprendizagem em conjunto. Fazendo uma analogia com o mundo real, seria buscar a resposta para o problema consultando diversas pessoas ao invés de uma pessoa. Para problema de regressão é utilizado à média e para problema de classificação é utilizado os votos da maioria.

Outro algoritmo utilizado foi o SVM (RAY, 2019) que é muito utilizado em problema de classificação mas que também podem ser utilizados em problemas de regressão. Uma característica em destaque neste algoritmo é que ele pode lidar com dados não lineares utilizando a

técnica que *kernel trick* que aumenta a dimensão do problema para poder separar o alvo.

Deep Learning, outro algoritmo utilizado neste trabalho e que sua utilização aumentou muito nos últimos anos devido principalmente a capacidade das GPUs. A base deste algoritmo é a *Rede Neural Artificial (RNA)* (RNA) com mais camadas. Embora a RNA possa fazer predições aproximadas, camadas ocultas adicionais podem ajudar a otimizar e aumentar a precisão. Estes algoritmos tendem a imitar o cérebro humano por meio de uma combinação de entradas de dados, pesos e tendências.

O último algoritmo que será utilizado é o *XBBoost* (CHEN; GUESTRIN, 2016). Um algoritmo recente que utiliza técnicas de *boosting* para tentar ajustar os dados de forma adaptativa. O *XBBoost* pode ser usado tanto para problemas de regressão quanto para problemas de classificação. Ele representa uma categoria de algoritmo baseada em árvores de decisão com aumento de gradiente. Aumento de gradiente significa que o algoritmo usa o algoritmo *Gradient Descent* para minimizar a perda. O *XBBoost* vai repetidamente criando novos modelos e os combina um modelo único. A cada ciclo ele constrói o modelo, adiciona no modelo agrupado e calcula o erro.

Os algoritmos de ML possui dois tipos de parâmetros, os parâmetros de modelos e *hyperparameters*. A principal diferença é os parâmetros de modelo podem ser apreendidos direto do treinamento enquanto *hyperparameters* não, por isso precisam ser otimizados. Para ajudar na otimização a técnica de validação cruzada foi utilizada. A validação cruzada é um processo de duas partes. A primeira consiste em realizar a otimização no período de treinamento. A segunda parte consiste em selecionar o melhor modelo encontrado na parte anterior e executar no período de teste. Kirkpatrick e Dahlquist (2011) sugerem utilizar de 20% a 30% do período de treinamento para o período teste. Após as duas partes, a janela de tempo é deslocada e se repetem as duas partes anteriores para as 5 etapas (*folds)*. No final é possível obter os melhores *hyperparameters*. A Tabela 1 mostra o range de valores que cada *hyperparameters* de cada algoritmo foi variado.

Serão realizadas modificações nos dados de entrada para dividir o problema em cortes de 90, 180, 365 e sem modificação. Para o corte de 90 dias quando a variável alvo estiver com mais de 90 dias ela será alterada para 90, para o corte de 180 dias quando a variável alvo estiver com mais de 180 dias ela será alterada para 180, para o corte de 365 dias, quando a variável alvo estiver com mais de 365 dias ela será alterada para 365 e o último corte irá manter os dados originais sem alteração. Isto será realizado para verificar o desempenho dos algoritmos em

Tabela 1 – *Hyperparameters* que serão otimizados de cada algoritmo.

Algoritmos	*Hyperparameters*	Valores testados
XGBoost	booster	gbtree, gblinear e dart
	eta	0, 0.3, 0.6 e 1
	$nEstimators$	10, 100, 400
Random Forest	$nEstimators$	10, 100, 200
	$maxFeatures$	auto, sqrt e log2
	$minSamplesLeaf$	1, 5, 10, 100
Ridge	L2	0.0001, 0.001, 0.01, 0.1, 1, 5, 10
Lasso	L1	0.0001, 0.001, 0.01, 0.1, 1, 5, 10.
SVM	tol	$1e-5$
	$loss$	epsilonInsensitiv e e squaredEpsilonInsensitive
Deep Learning	$optimizer$	Adam e RMSprop
	$activation$	ReLU e Softmax
	Camadas	1 camada oculta com 8 neurônios
Linear	-	-

Fonte: Autoria própria.

grupos de dados e sem a influência dos *outliers*.

Outra modificação nos dados de entrada que serão avaliadas é a aplicação da técnica de redução de dimensionalidade (*Principal Components Analysis* (PCA)) que irá reduzir os 40 atributos de entrada para poucos atributos mas mantendo a cobertura do espaço amostral.

Os algoritmos foram incluídos em um *pipeline* (ver Algoritmo 1) para executar de forma automática a validação cruzada e executar todas as combinações possíveis dos experimentos. O ciclo principal representa os tipos de cortes (linha 1). Foi utilizado dado da variável alvo com corte de 90 ($corte = 0$), 180 ($corte = 1$) e 365 ($corte = 2$) dias além dos experimentos sem cortes ($corte = 3$). Na linha 2 foi repetido os experimento utilizando o PCA ($pca = 0$) e sem ($pca = 1$). O PCA foi executado para poder comparar no final dos experimentos se vale a pena executar todos os 40 atributos ou se com apenas 18 atributos encontrados pelo PCA que cobrem 98% do espaço já é suficiente para ter um desempenho aceitável. Na linha 3, para cada algoritmo a validação cruzada com *fold* 5 foi executada para encontrar os melhores *hyperparameters* e realizado o treinamento. E finalmente na linha 5 o modelo treinado de cada tipo de combinação foi testado.

Algoritmo 1 – Pipeline de execução.

```
1: para corte = 0 até 3 faça
2:     para pca = 0 até 1 faça
3:         para algoritmo = 0 até 6 faça
4:             Treinar modelo com validação cruzada
5:             Testar modelo e preparar resultados
6:         finaliza para
7:     finaliza para
8: finaliza para
```

Fonte: Autoria própria.

Para executar os experimentos foi utilizado um computador I9 G10 20 cores e 32GB RAM. Para a implementação dos algoritmos foi utilizada a biblioteca *open-source* scikit-learn (http://scikit-learn.org) com exceção do algoritmo de *Deep Learning*, oriundo do pacote Keras https://keras.io/ em conjunto com um *wrapper* para poder acoplar no *pipeline* do *GridSearchCV* para poder encontrar os melhores *hyperparameters*. O trabalho foi realizado utilizando a linguagem python em conjunto com outras bibliotecas de suporte como o pandas (CHEN, 2018; MCKINNEY, 2018; KAZIL; JARMUL, 2016; GRUS, 2016).

3.3 METODOLOGIA DE AVALIAÇÃO

Para medir o desempenho, além do *Root Mean Square Error* (RMSE) que penaliza erros maiores e é mais eficiente computacionalmente, outras métricas foram utilizadas como o *Mean Absolute Error* (MAE) que realiza um penalização linear e o R^2 que é uma métrica padrão para tarefas de regressão e por fim o R^2 ajustado que penaliza o modelo de acordo com o número de atributos usados (definido pela Equação (1)).

$$AdjR^2 = 1 - \frac{(1 - R^2) * (n - 1)}{n - p - 1} \tag{1}$$

na qual p é a quantidade de atributos e n é a quantidade de observações.

Valores negativos podem ser encontrados para a variável alvo dias em atraso pelos algoritmos. Quando isto ocorre, eles foram ajustados para zero porque no mundo real o valor negativo seria como se o cliente tivesse pagado antes do vencimento.

3.3.1 Análise Estatística

Para determinar se existe diferença estatística entre os modelos será realizado os testes estatísticos de Friedman e Nemenyi (GRANATYR, 2018).

Primeiramente os experimentos serão executados 30 vezes para cada algoritmo com sementes aleatórias diferentes. Depois os cálculos estatísticos serão executados no software R (https://www.r-project.org/) utilizando a biblioteca *open-source* TStools (https://github.com/trnnick/TStools).

4 RESULTADOS

A Tabela 2 mostra o resultados da otimização dos *hyperparameters* para os algoritmos. Os modelos executaram a etapa de teste que serão mostrados na sequência utilizando estes valores.

Tabela 2 – *Hyperparameters* otimizados de cada algoritmo.

Algoritmo	*Hyperparameters*
XGBoost	booster="gbtree", eta=0.3 e nEstimators=400
Random Forest	maxFeatures="auto", minSamplesLeaf=5 e nEstimators=200
Ridge	L2=10
Lasso	L1=0.01
SVM	loss="epsilonInsensitive"e tol=$1e-5$
Deep Learning	optimizer="Adam"e activation="ReLU"
Linear	-

Fonte: Autoria própria.

As Tabela 3 e Tabela 4 mostram os resultados dos modelos utilizando os dados de testes com corte de 90 dias sem o PCA e com o PCA respectivamente. Os melhores resultados estão destacados em negrito.

Tabela 3 – Resultado dos modelos com corte de 90 dias sem PCA.

Algoritmo	R^2	Adj R^2	MAE	MSE	RMSE
Linear	0,5585	0,5581	12,6092	311,6358	17,6532
SVM	0,5069	0,5065	10,9590	348,0058	18,6549
Deep	0,6433	0,6430	10,5894	251,7373	15,8662
Lasso	0,5583	0,5579	12,6144	311,7425	17,6562
Ridge	0,5585	0,5581	12,6092	311,6400	17,6533
RF	**0,7275**	**0,7273**	**8,3200**	**192,3256**	**13,8681**
XGBoost	0,7099	0,7096	9,0525	204,7822	14,3102

Fonte: Autoria própria.

Tabela 4 – Resultado dos modelos com corte de 90 dias com PCA.

Algoritmo	R^2	Adj R^2	MAE	MSE	RMSE
Linear	0,5479	0,5477	12,8889	319,0927	17,8632
SVM	0,4976	0,4974	11,1531	354,5997	18,8308
Deep	0,6428	0,6427	10,2184	252,1155	15,8781
Lasso	0,5479	0,5477	12,8913	319,1214	17,8640
Ridge	0,5479	0,5477	12,8891	319,0972	17,8633
RF	**0,7138**	**0,7137**	**8,4869**	**202,0176**	**14,2133**
XGBoost	0,6921	0,6920	9,2877	217,3315	14,7422

Fonte: Autoria própria.

As Tabela 5 e Tabela 6 mostram os resultados dos modelos utilizando os dados de testes com corte de 180 dias sem o PCA e com o PCA respectivamente.

Tabela 5 – Resultado dos modelos com corte de 180 dias sem PCA.

Algoritmo	R^2	Adj R^2	MAE	MSE	RMSE
Linear	0,5976	0,5972	15,7999	727,8781	26,9792
SVM	0,5481	0,5477	14,0234	817,3559	28,5894
Deep	0,6865	0,6862	13,5836	567,0313	23,8124
Lasso	0,5975	0,5972	15,8041	727,9815	26,9811
Ridge	0,5976	0,5972	15,8003	727,8853	26,9793
RF	**0,7680**	**0,7678**	**10,3877**	**419,6897**	**20,4863**
XGBoost	0,7615	0,7613	11,2040	431,3633	20,7693

Fonte: Autoria própria.

Tabela 6 – Resultado dos modelos com corte de 180 dias com PCA.

Algoritmo	R^2	Adj R^2	MAE	MSE	RMSE
Linear	0,5884	0,5882	16,1946	744,5063	27,2856
SVM	0,5424	0,5422	14,4624	827,6665	28,7692
Deep	0,6773	0,6772	13,2377	583,7094	24,1601
Lasso	0,5884	0,5882	16,1967	744,5206	27,2859
Ridge	0,5884	0,5882	16,1950	744,5300	27,2861
RF	**0,7524**	**0,7523**	**10,6793**	**447,9092**	**21,1639**
XGBoost	0,7357	0,7356	11,6360	478,0869	21,8652

Fonte: Autoria própria.

As Tabela 7 e Tabela 8 mostram os resultados dos modelos utilizando os dados de testes com corte de 365 dias sem o PCA e com o PCA respectivamente.

Tabela 7 – Resultado dos modelos com corte de 365 dias sem PCA.

Algoritmo	R^2	Adj R^2	MAE	MSE	RMSE
Linear	0,6362	0,6358	21,4240	2032,4190	45,0824
SVM	0,5459	0,5455	19,3643	2536,8419	50,3671
Deep	0,7433	0,7431	16,8233	1433,9832	37,8680
Lasso	0,6362	0,6358	21,4254	2032,4920	45,0832
Ridge	0,6362	0,6358	21,4241	2032,4287	45,0825
RF	0,8258	0,8256	**12,8994**	973,1054	31,1946
XGBoost	**0,8285**	**0,8284**	13,7030	**957,9523**	**30,9508**

Fonte: Autoria própria.

Tabela 8 – Resultado dos modelos com corte de 365 dias com PCA.

Algoritmo	R^2	Adj R^2	MAE	MSE	RMSE
Linear	0,6101	0,6099	22,8335	2178,2554	46,6718
SVM	0,5083	0,5081	21,0150	2746,8730	52,4106
Deep	0,7161	0,7160	17,6775	1586,1072	39,8260
Lasso	0,6100	0,6099	22,8343	2178,4326	46,6737
Ridge	0,6101	0,6099	22,8339	2178,2920	46,6722
RF	**0,8079**	**0,8078**	**13,4573**	**1073,2314**	**32,7602**
XGBoost	0,8009	0,8008	14,4450	1112,2124	33,3498

Fonte: Autoria própria.

As Tabela 9 e Tabela 10 mostram os resultados dos modelos utilizando os dados de testes sem corte dias e sem o PCA e com o PCA respectivamente.

Tabela 9 – Resultado dos modelos sem corte de dias e sem PCA.

Algoritmo	R^2	Adj R^2	MAE	MSE	RMSE
Linear	0,6216	0,6212	47,4573	16473,0705	128,3475
SVM	0,4477	0,4472	39,4125	24041,9965	155,0548
Deep	0,7560	0,7558	30,0676	10621,6022	103,0612
Lasso	0,6215	0,6212	47,4516	16473,5797	128,3494
Ridge	0,6216	0,6212	47,4569	16473,1943	128,3479
RF	0,8947	0,8947	**20,2342**	4581,4393	67,6863
XGBoost	**0,9159**	**0,9158**	20,2878	**3661,7880**	**60,5127**

Fonte: Autoria própria.

Tabela 10 – Resultado dos modelos sem corte de dias e com PCA.

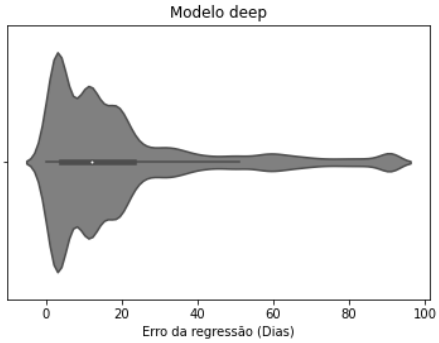
Algoritmo	R^2	Adj R^2	MAE	MSE	RMSE
Linear	0,5720	0,5718	53,8608	18630,9757	136,4953
SVM	0,2553	0,2550	45,3192	32414,6915	180,0408
Deep	0,7169	0,7168	33,7871	12320,9580	110,9998
Lasso	0,5720	0,5718	53,8581	18630,8975	136,4950
Ridge	0,5720	0,5718	53,8606	18631,1560	136,4960
RF	0,8639	0,8638	**22,1330**	5924,4302	76,9703
XGBoost	**0,8662**	**0,8661**	23,3439	**5825,5281**	**76,3251**

Fonte: Autoria própria.

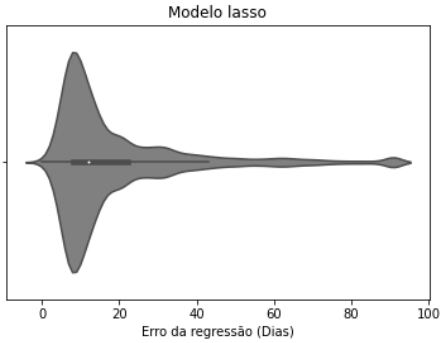
A Figura 1 mostra a densidade probabilística para cada modelo. Com o gráfico de densidade probabilística, além das informações dos quantils existe a exibição da densidade dos dados.

A Figura 2 mostra o *boxplot* para cada modelo. Com o *boxplot* é possível ver a alteração dos dados da variável alvo "dias em atraso"por meio de quantis. Os pontos desgarrados são os *Outliers*. Outras estatísticas são: o mínimo, o primeiro quantil, a mediana (o traço no meio do retângulo), o terceiro quantil e o máximo.

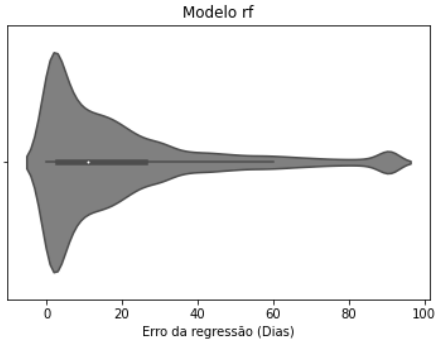
Figura 1 – Densidade probabilística de cada modelo



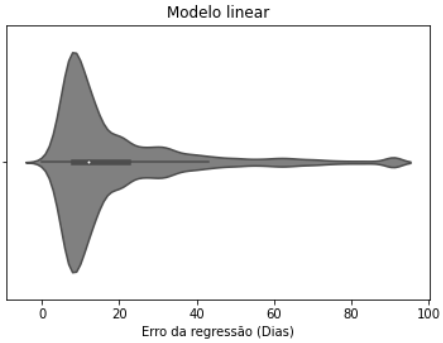
(a) Deep Learning



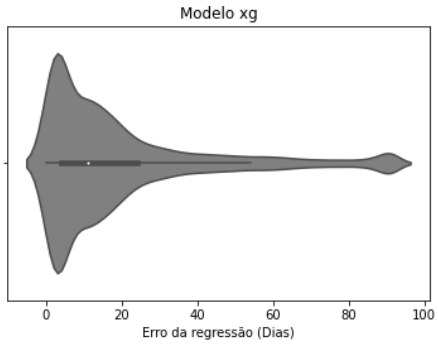
(b) Lasso



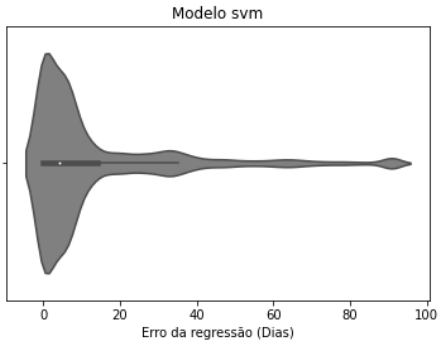
(c) RF



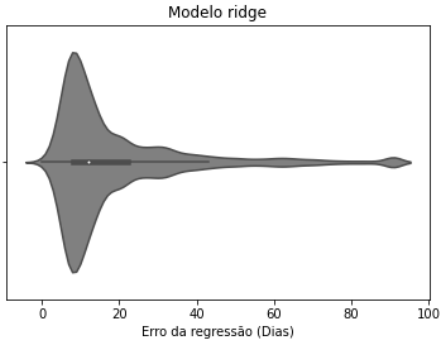
(d) Linear



(e) XGBoost



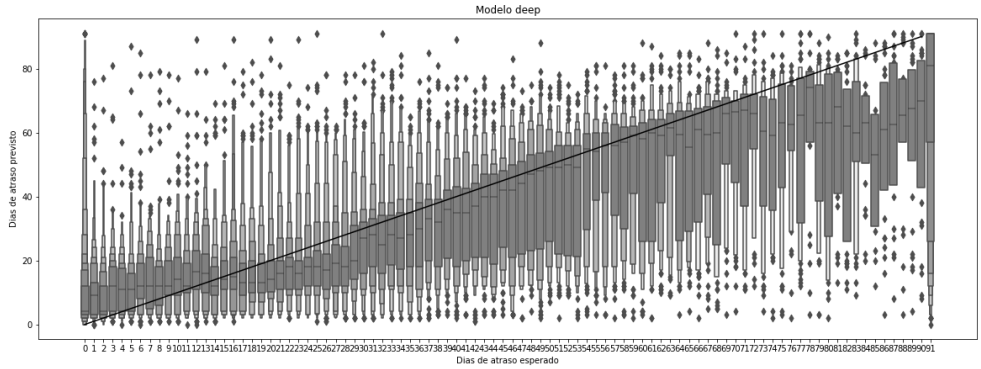
(f) SVM



(g) Ridge

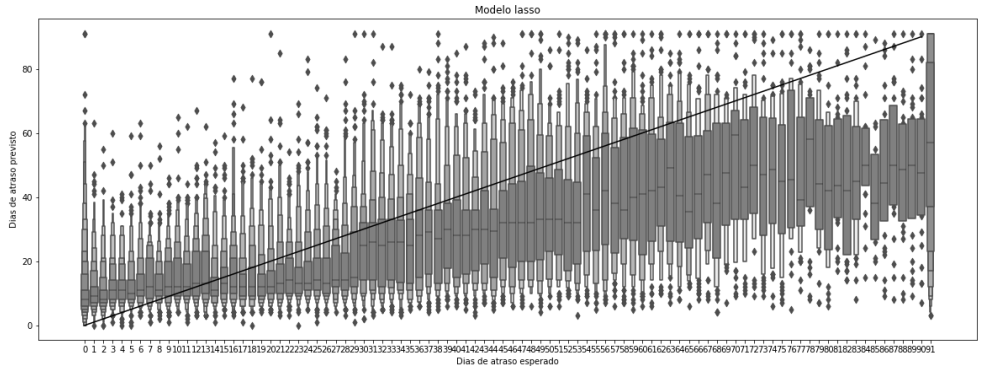
Fonte: Autoria própria.

Figura 2 – *Boxplot* do *Deep Learning*.



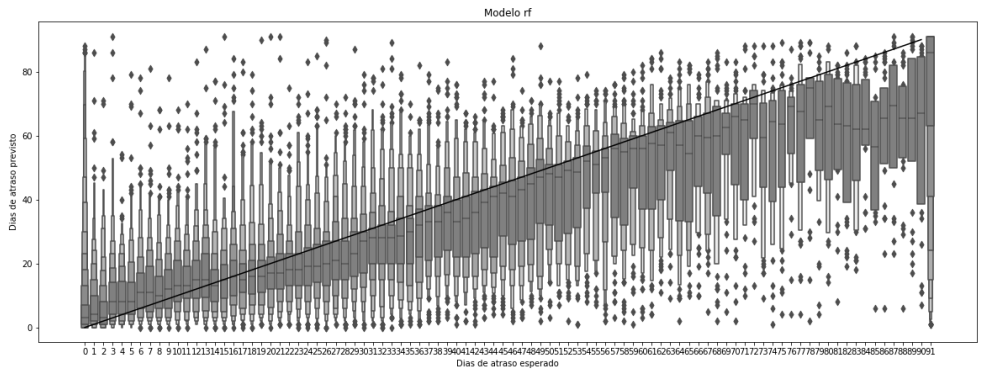
Fonte: Autoria própria.

Figura 3 – *Boxplot* do Lasso.



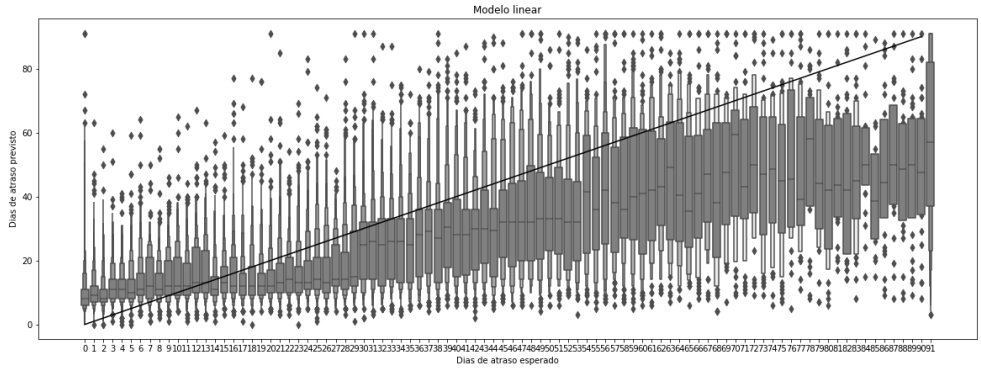
Fonte: Autoria própria.

Figura 4 – *Boxplot* do RF.



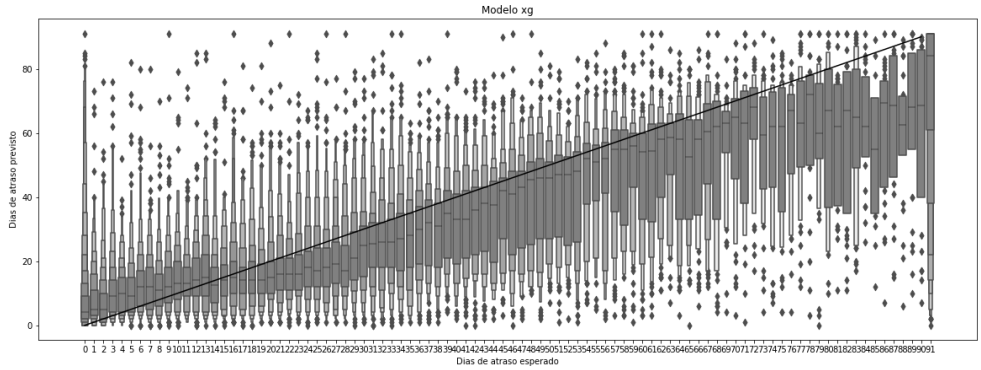
Fonte: Autoria própria.

Figura 5 – *Boxplot* do Linear.



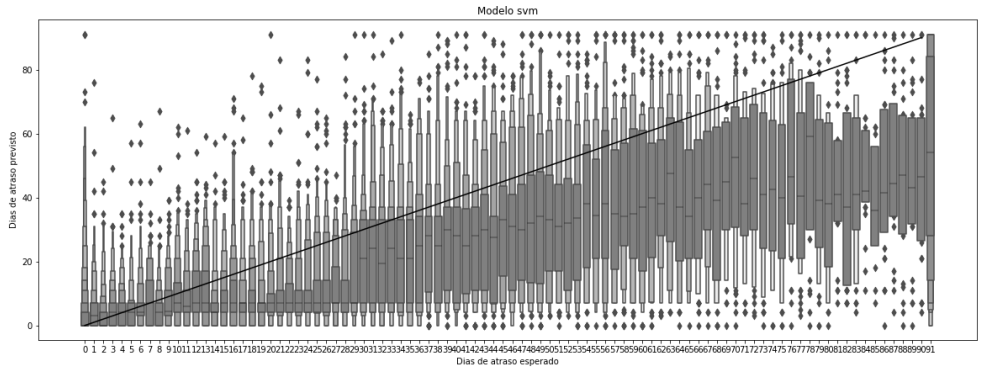
Fonte: Autoria própria.

Figura 6 – *Boxplot* do XGBoost.



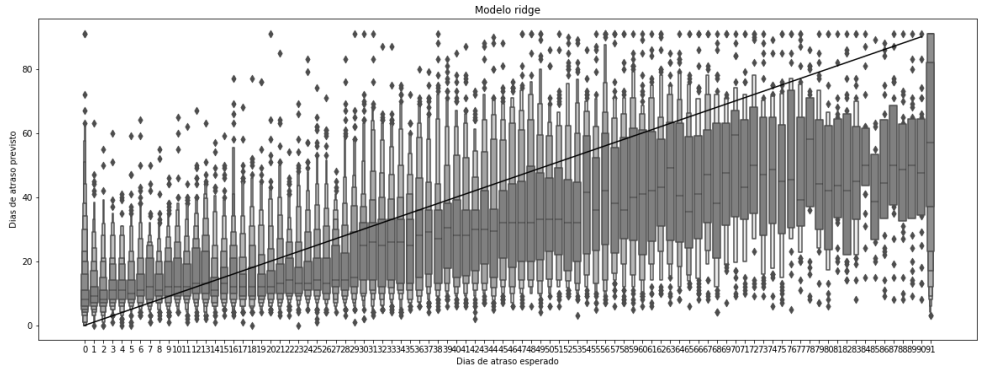
Fonte: Autoria própria.

Figura 7 – *Boxplot* do SVM.



Fonte: Autoria própria.

Figura 8 – *Boxplot* do Ridge.



Fonte: Autoria própria.

5 DISCUSSÕES

Após os experimentos é percebido que as execuções separado por grupo de corte (Tabela 4 à Tabela 10) traz melhor desempenho para os modelos quando o grupo de corte é menor (ver Tabela 11). O corte de 90 dias, por exemplo, foi o que obteve melhor resultado. Até mesmo a linha de corte de 365 dias já retorna um ganho de mais de 50% de desempenho em relação ao experimento sem corte. Isto ocorre devido aos *outliers*, existem alguns clientes que estão com mais de 1600 dias sem pagar a fatura.

Tabela 11 – Resultado consolidado por grupo de corte sem PCA.

	RMSE						
Corte (dias)	Deep	Lasso	Ridge	RF	SVM	XGBoost	Linear
90	**15,8662**	**17,6562**	**17,6533**	**13,8681**	**18,6549**	**14,3102**	**17,6532**
180	23,8124	26,9811	26,9793	20,4863	28,5894	20,7693	26,9792
365	37,8680	45,0832	45,0825	31,1946	50,3671	30,9508	45,0824
Sem corte	103,0612	128,3494	128,3479	67,6863	155,0548	60,5127	128,3475

Fonte: Autoria própria.

Na Tabela 12 é possível verificar que caso o requisito de desempenho de *hardware* seja limitado é possível utilizar o PCA para diminuir a dimensionalidade de 40 atributos para 18 e mesmo assim continuar com 98% do espaço amostral e retornado um desempenho ligeiramente inferior.

Tabela 12 – Resultado consolidado por grupo de corte com PCA.

	RMSE						
Corte (dias)	Deep	Lasso	Ridge	RF	SVM	XGBoost	Linear
90	**15,8781**	**17,8640**	**17,8633**	**14,2133**	**18,8308**	**14,7422**	**17,8632**
180	24,1601	27,2859	27,2861	21,1639	28,7692	21,8652	27,2856
365	39,8260	46,6737	46,6722	32,7602	52,4106	33,3498	46,6718
Sem corte	110,9998	136,4950	136,4960	76,9703	180,0408	76,3251	136,4953

Fonte: Autoria própria.

Como já mencionado não existe predição negativa para os dias de atraso. Caso o modelo encontrasse o valor negativo eles foram alterados *hardcode* (dentro no código-fonte), para zero para simular que o cliente pagou adiantado a fatura (ver Figura 9). Esta alteração é importante realizar para não influenciar negativamente o desempenho dos algoritmos no momento de realizar a análise de desempenho utilizando as métricas já descritas porque o *dataset* não possui valores negativos para o atributo alvo.

Na Figura 1 é possível verificar que os modelos erraram na maioria das vezes até 30 dias de erro. Isto é número muito bom por que se o modelo errar em até 30 dias o impacto para a

Figura 9 – Alteração dos valores preditos negativos.

Index	y_teste	pred	abs
0	0	-15.7753	15.7753
1	90	78.5913	11.4087
2	293	103.959	189.041
3	21	128.897	107.897
4	0	-13.0169	13.0169
5	16	46.7392	30.7392
6	34	-8.62839	42.6284
7	0	-22.1435	22.1435
8	0	93.7516	93.7516
9	0	11.678	11.678
10	0	-17.0046	17.0046
11	43	-6.81766	49.8177
12	30	48.8784	18.8784
13	1659	1555.29	103.714
14	31	25.4293	5.57074
15	33	28.741	4.25899

y_teste	pred	abs
0	0	0
90	78.5913	11.4087
293	103.959	189.041
21	128.897	107.897
0	0	0
16	46.7392	30.7392
34	0	34
0	0	0
0	93.7516	93.7516
0	11.678	11.678
0	0	0
43	0	43
30	48.8784	18.8784
1659	1555.29	103.714
31	25.4293	5.57074
33	28.741	4.25899

Fonte: Autoria própria.

empresa não é significativo.

A Tabela 13 mostra o resultado de cada algoritmo executado 30 vezes utilizando sementes aleatórias distintas. Com estes dados foi possível calcular os testes estatísticos utilizando o software R para verificar se existe diferença significativa entre os modelos.

A Figura 10 apresenta o teste de Friedman e Nemenyi. A *Critical Distance* (CD) foi de 1.644. O modelo XGBoost e o modelo *Random Forest* apresentam um CD = 1 (2-1) o que indica que não existe diferença estatística significativa entre eles, pois o valor da distância é menor que a distância crítica de 1.644. O modelo Deep Learning e o modelo *Random Forest* apresentam um CD = 2 (3-1) o que indica que existe diferença estatística significativa entre eles, pois o valor da distância é maior que a distância crítica de 1.644. Visualmente também é possível verificar rapidamente entre os modelos se existe ou não diferença estatística significativa. Caso não tenha linha corte entre os modelos não existe diferença. Ver por exemplo o modelo Linear, Ridge e Lasso que não existem diferentes entre eles. Já entre SVM e Ridge ou Linear existe. Entre SVM e Lasso não existe.

Com os resultados dos testes estatísticos e usando os modelos com a configuração dos *hyperparameters* da Tabela 2 e com o volume de dados utilizados neste trabalho temos os melhores resultados obtido para o *Random Forest* porém não há diferença estatística para o *XGBoost* e existe diferença signifitcativa a partir do Deep Learning. Com esta informação é possível escolher não executar o modelo Deep Learning que é o que exige mais recurso de

Tabela 13 – Resultado das 30 execuções com sementes aleatórias.

	RMSE						
Execução	Linear	Lasso	Ridge	RF	XGBoost	SVM	Deep
1	17,6072	17,6067	17,6072	13,7362	14,1564	18,6452	15,8164
2	17,6270	17,6276	17,6273	13,7905	14,2280	18,6569	15,7119
3	17,6556	17,6591	17,6558	13,8518	14,2806	18,7005	15,7359
4	17,5486	17,5487	17,5488	13,6510	14,0414	18,5706	15,5393
5	17,5094	17,5097	17,5095	13,7017	14,0856	18,5246	15,5781
6	17,6732	17,6735	17,6734	13,8680	14,2708	18,7317	15,6907
7	17,7796	17,7822	17,7795	13,9266	14,3183	18,8697	15,9592
8	17,7160	17,7125	17,7159	13,9673	14,3710	18,7415	15,7848
9	17,5914	17,5958	17,5915	13,7630	14,2238	18,6090	15,6961
10	17,6026	17,6050	17,6027	13,8415	14,1876	18,6520	15,5572
11	17,6306	17,6309	17,6306	13,8635	14,2877	18,6547	15,6979
12	17,5706	17,5729	17,5707	13,7786	14,2464	18,6166	15,7036
13	17,7548	17,7557	17,7550	13,9327	14,3363	18,8146	15,8840
14	17,7364	17,7406	17,7369	13,8377	14,2706	18,7745	15,7011
15	17,6622	17,6657	17,6622	13,8372	14,2177	18,6726	15,8404
16	17,6573	17,6596	17,6575	13,9162	14,3427	18,6447	15,7882
17	17,6141	17,6168	17,6143	13,8012	14,3046	18,6482	15,7078
18	17,4957	17,4972	17,4957	13,7121	14,1572	18,5002	15,5949
19	17,5622	17,5633	17,5622	13,7642	14,1236	18,5440	15,6138
20	17,5697	17,5732	17,5697	13,8688	14,2222	18,5805	15,6934
21	17,7029	17,7069	17,7030	13,8791	14,2280	18,7483	15,7181
22	17,7314	17,7301	17,7313	13,9300	14,4095	18,7814	15,8148
23	17,7513	17,7457	17,7514	13,8567	14,2699	18,8026	15,7638
24	17,6797	17,6823	17,6810	13,8248	14,2347	18,6914	15,6605
25	17,5082	17,5125	17,5082	13,5885	14,1085	18,5105	15,5140
26	17,6290	17,6323	17,6290	13,8049	14,2944	18,6791	15,6892
27	17,6383	17,6399	17,6383	13,7047	14,1283	18,6496	15,6281
28	17,7697	17,7730	17,7698	13,8596	14,3199	18,8044	15,7892
29	17,4743	17,4760	17,4743	13,7003	14,0745	18,4876	15,5481
30	17,5669	17,5708	17,5670	13,9221	14,2753	18,5742	15,9663
Média	17,6339	17,6355	17,6340	13,8160	14,2339	18,6627	15,7129

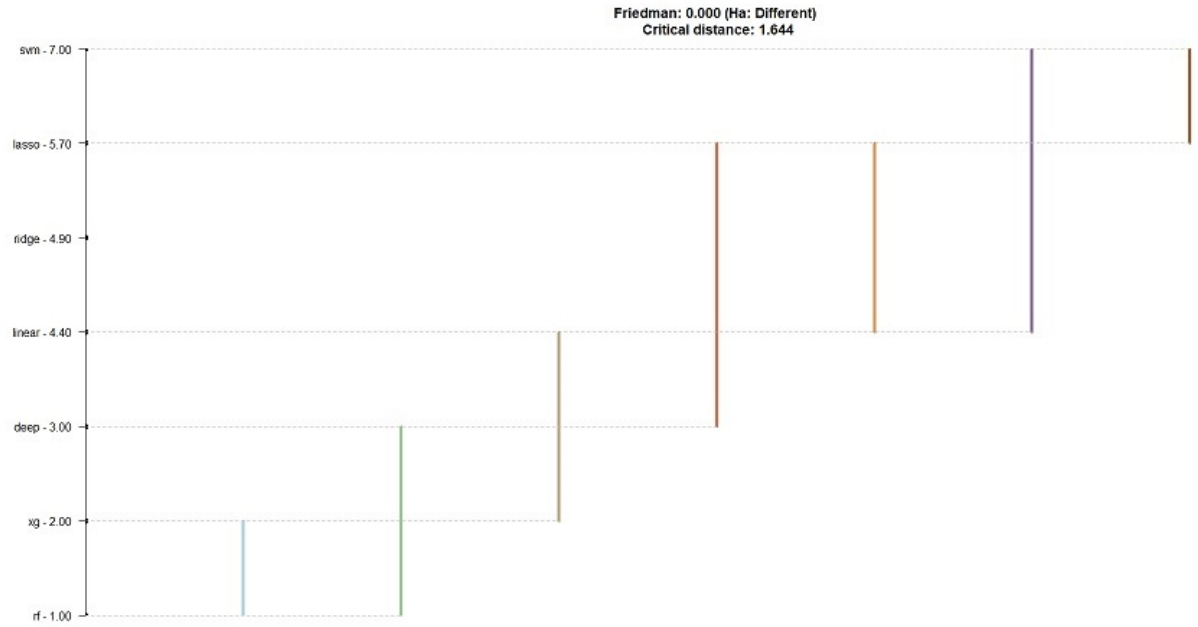
Fonte: Autoria própria.

hardware e executar o *Random Forest* ou o *XGBoost* que exigem menos recurso e o desempenho foi estatisticamente superior.

Da Figura 2 até Figura 8 mostra o gráfico de desempenho dos algoritmos corroborando com a análise estatística realizada na qual o *Random Forest* e XGBoost tiveram os melhores resultados e estatisticamente iguais. No eixo x estão os valores esperados e no eixo y os valores preditos pelos algoritmos. O modelo perfeito seria que a mediana ficasse em cima da linha de 45 graus. É possível perceber que os três melhores modelos a mediada fica próxima da linha de 45 grau enquanto os outros modelos a mediana se afasta (ver pior modelo Figura 7).

Um desafio que todo cientista de dados tem é como explicar o resultado do modelo para os *stakeholders*. Geralmente os algoritmos são uma caixa preta que não é possível verificar o porquê ele chegou ao resultado. Este trabalho utilizou o algoritmo SHAP para interpretar o

Figura 10 – Resultado do cálulo estatístico.



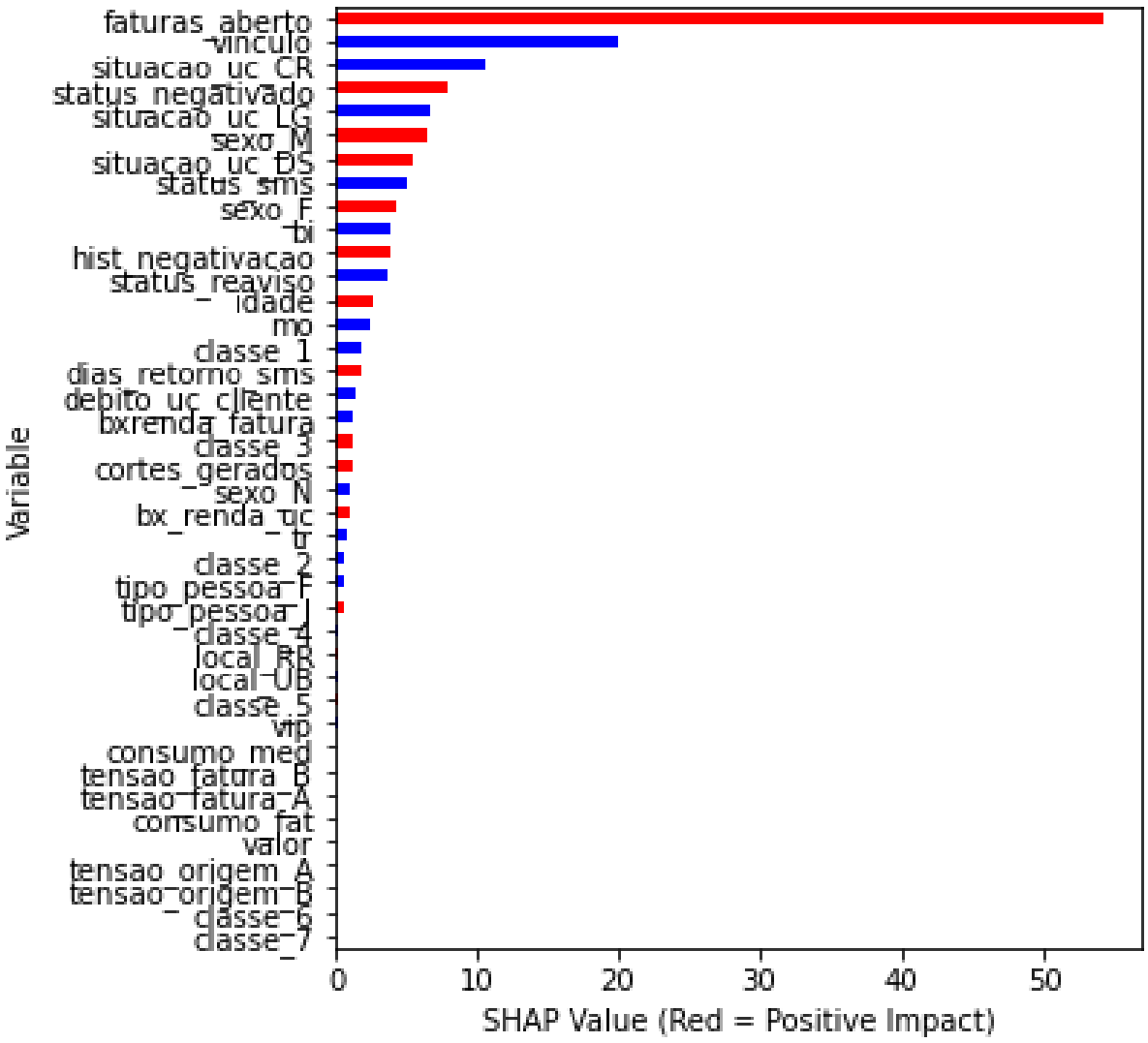
Fonte: Autoria própria utilizando o software R em conjunto com a biblioteca TStools.

resultados dos modelos.

A Figura 11 mostra o gráfico de relacionamento de cada variável preditora com a variável algo "dias em atraso", na qual lista as variáveis mais significativas em ordem decrescente. As variáveis superiores contribuem mais para o modelo do que as inferiores e, portanto, têm alto poder preditivo. Este gráfico também demonstra as seguintes informações:

- Importância do recurso: as variáveis são classificadas em ordem decrescente;
- Impacto: o tamanho da barra horizontal mostra se o efeito desse valor está associado a uma previsão mais alta ou mais baixa;
- Valor original: a cor mostra se essa variável é alta (em vermelho) ou baixa (em azul) para aquela observação;
- Correlação: alta quantidade de faturas abertas “faturas_aberto” tem um impacto alto e positivo na variável alvo "dias em atraso". Traduzindo para *stakeholders*, se um cliente tiver muitas faturas abertas a chance dele não pagar a próxima aumenta muito, o que faz sentido no mundo real. O “alto” vem da cor vermelha e o impacto “positivo” é mostrado pelo tamanho da barra. Da mesma forma, diremos que se o SMS foi enviado para o cliente vai contribuir para diminuir o valor do atraso.

Figura 11 – Relacionamento de cada variável



Fonte: Autoria própria utilizando o algoritmo SHAP.

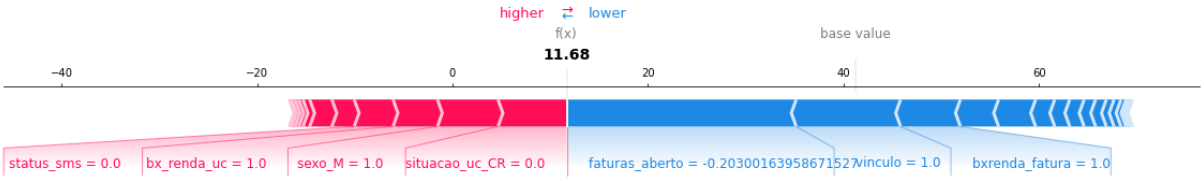
Os valores podem ser feitos para faturas individuais. De Figura 12 à Figura 15 apresenta-se faturas escolhidas aleatoriamente para exemplificar a interpretação.

A Figura 12 mostra o valor de 11 dias predito pelo modelo. Os atributos que empurram a previsão para cima (para a direita) são mostrados em vermelho, e aqueles que empurram a previsão para baixo são exibidos em azul. Para esta fatura se o cliente não tiver outras faturas em aberto contribui para diminuir quando ele irá pagar (“faturas_aberto”). Outro atributo que ajuda a diminuir os dias de pagamento é se o cliente da fatura é o mesmo da *Unidade Consumidora* (UC) (“vínculo” = 1), o que faz sentido para o mundo real, porque se a fatura é do mesmo cliente que esta atualmente na UC ele vai pagar o mais rápido possível para não ser cortado. Na Figura 13 é possível ver a relação do atributo “vínculo” pelo lado negativo, ou seja, se o cliente da fatura não é o mesmo da UC atual ele não tem pressa em pagar porque ele já saiu do local e outra pessoa

esta morando, consequentemente este atributo contribui para elevar muito a predição de quanto vai pagar.

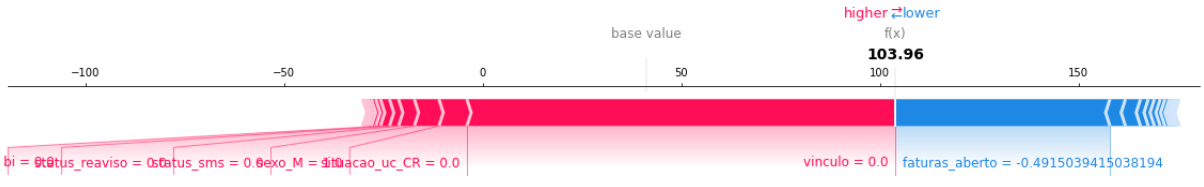
A Figura 14 mostra uma fatura que foi predita que o cliente vai pagar em 371 dias. Dois atributos foram decisivos para esta predição, o cliente da fatura não é o mesmo da UC atual (“vinculo” = 0) e principalmente ele já existe duas faturas abertas para o cliente. O que faz sentido no mundo real porque quanto mais faturas o cliente estiver devendo, a chance dele não pagar uma nova fatura aumenta. A Figura 15 mostra um caso exagerado no qual o cliente foi predito que irá pagar em 1555 dias. Foi um conjunto de atributos com grande efeito. Primeiro se o cliente estiver negativado na praça (“status_negativado” = 1), o que faz sentido no mundo real. Segundo ele possui 8 faturas sem pagamento (“faturas_aberto” = 1) e a UC esta desligada (“situação_uc_DS” = 0), ou seja, a UC já está desligada, provavelmente ele não está mais na residência porque esta sem energia e ele deve oito faturas e esta negativado na praça, por isso esse é um dos casos com o atraso mais longo.

Figura 12 – Resultado da interpretação do modelo para os *stakeholders* de uma fatura isolada



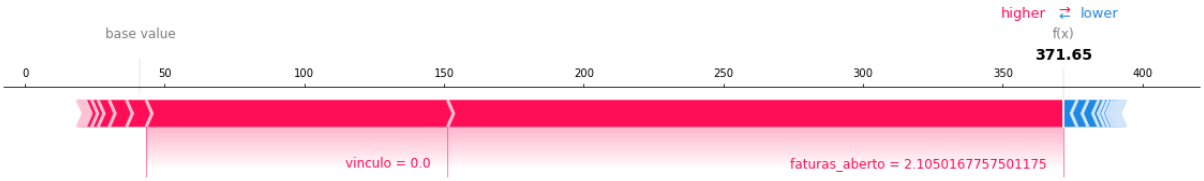
Fonte: Autoria própria utilizando o algoritmo SHAP.

Figura 13 – Resultado da interpretação do modelo para os *stakeholders*.de uma fatura isolada



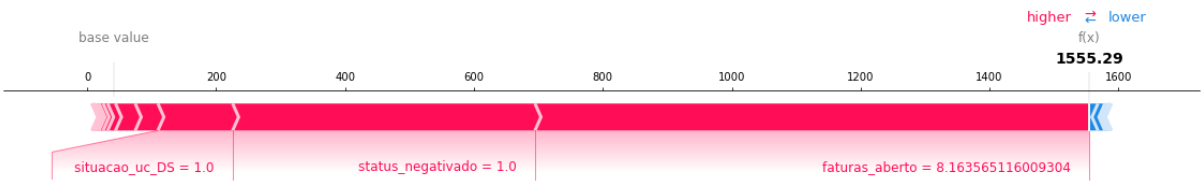
Fonte: Autoria própria utilizando o algoritmo SHAP.

Figura 14 – Resultado da interpretação do modelo para os *stakeholders*.de uma fatura isolada



Fonte: Autoria própria utilizando o algoritmo SHAP.

Figura 15 – Resultado da interpretação do modelo para os *stakeholders*.de uma fatura isolada



Fonte: Autoria própria utilizando o algoritmo SHAP.

6 CONCLUSÕES E PERSPECTIVAS

Este trabalho teve como objetivo encontrar um modelo para ajudar na predição do problema de PECLD utilizando diversos algoritmos de aprendizagem de máquina da literatura.

Para avaliar a metodologia proposta foram realizados experimentos utilizando dados reais de uma Distribuidora de Energia Elétrica. Os modelos depois de testados foram avaliados entre si para verificar se existia diferença significativa.

De modo geral, os resultados obtidos foram promissores, o que permite concluir que é possível utilizar os conceitos deste trabalho em um ambiente real, substituindo a decisão humana entre qual unidade consumidora cortar por um modelo automatizado. No entanto, cabe ressaltar que para automatizar este processo em um ambiente real diversos etapas fora do escopo deste trabalho devem ser realizadas como, por exemplo, criar um serviço que executa o modelo automaticamente para quando uma fatura nova for gerada em determinado sistema da empresa chame este serviço e já mostre o resultado online para o usuário.

Outro ponto relevante para colocar os conceitos deste trabalho em produção é que será necessário realizar um acompanhamento para verificar se os resultados encontrados nos testes pelos modelos são próximos no ambiente real, porque pode haver casos de clientes que não estavam nos dados de treinamento.

Treinar o modelo periodicamente com dados mais atualizados também é interessante porque os dados podem ter mudados. Por exemplo, durante o período de *lockdown* devido ao vírus clientes podem ter deixados de pagar as faturas porque perderam o emprego. Esta informação não está presente nos dados desde trabalho.

A separação dos grupos de corte criado nos experimentos mostrou que o *stakeholder* irá ter a possibilidade de escolher qual corte utilizar e quando menor for o corte maior será o desempenho dos modelos. Ele pode, por exemplo, realizar um tratamento especial para os *outliers* que estão com mais de 365 sem pagar a fatura e utilizar o modelo de corte de 90 dias que retirar estes valores e consequentemente obter um melhor desempenho no modelo.

Como trabalhos futuros o uso da técnica de *reframing* (LAKSHMANAN *et al.*, 2020) para transformar o problema de regressão em um problema de classificação pode ser realizado. Outra abordagem complementar a este trabalho que pode ser realizada é que depois que o modelo proposto neste trabalho encontrou o resultado de quando o cliente irá pagar a conta, utilizar estes dados como entrada de um algoritmo de otimização e juntos com outros dados, como

por exemplo, quantidade de equipes disponíveis, custos de desligamento, valor da fatura etc. encontrar uma ordem de desligamento otimizada para maximar o lucro.

REFERÊNCIAS

BASTOS, João A. Forecasting bank loans loss-given-default. **Journal of Banking & Finance**, p. 2510–2517, 2010.

BREIMAN, L.; FRIEDMAN, J. H.; OLSHEN, R. A.; STONE, C. J. **Classification and Regression Trees**. Monterey, CA: Wadsworth and Brooks, 1984.

CHAWLA, N. V.; BOWYER, K. W.; HALL, L. O.; KEGELMEYER, W. P. Smote: Synthetic minority over-sampling technique. **Journal of Artificial Intelligence Research**, AI Access Foundation, v. 16, p. 321–357, Jun 2002. ISSN 1076-9757. Disponível em: http://dx.doi.org/10.1613/jair.953.

CHEN, D.Y. **Análise de Dados com Python e Pandas**. *[S.l.]*: Novatec, 2018.

CHEN, Tianqi; GUESTRIN, Carlos. **XGBoost: A Scalable Tree Boosting System**. 2016. Cite arxiv:1603.02754Comment: KDD'16 changed all figures to type1. Disponível em: http://arxiv.org/abs/1603.02754.

Datta, A.; Sen, S.; Zick, Y. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. *In*: **2016 IEEE Symposium on Security and Privacy (SP)**. *[S.l.: s.n.]*, 2016. p. 598–617.

GRANATYR, J. **Modelo Afetivo de Reputação utilizando Personalidade e Emoção**. *[S.l.]*: Novas Edicoes Academicas, 2018.

GRUS, J. **Python para análise de dados**. *[S.l.]*: Alta Books, 2016.

KAZIL, J.; JARMUL, K. **Data Wrangling with Python**. *[S.l.]*: O'Reilly, 2016.

KIRKPATRICK, Charles D.; DAHLQUIST, Julie. **Technical Analysis: the Complete Resource for Financial Market Technicians**. Upper Saddle River, New Jersey, USA: Pearson Education, 2011.

LAKSHMANAN, V.; ROBINSON, S.; MUNN, M. **Machine Learning Design Patterns**. *[S.l.]*: O'Reilly Media, Inc., 2020.

LIPOVETSKY, Stan; CONKLIN, Michael. Analysis of regression in game theory approach. **Applied Stochastic Models in Business and Industry**, v. 17, p. 319 – 330, 10 2001.

LUNDBERG, Scott M; LEE, Su-In. A unified approach to interpreting model predictions. *In*: GUYON, I.; LUXBURG, U. V.; BENGIO, S.; WALLACH, H.; FERGUS, R.; VISHWANATHAN, S.; GARNETT, R. (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2017. v. 30. Disponível em: https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

LUNDBERG, Scott M; NAIR, Bala; VAVILALA, Monica S; HORIBE, Mayumi; EISSES, Michael J; ADAMS, Trevor; LISTON, David E; LOW, Daniel King-Wai; NEWMAN, Shu-Fang; KIM, Jerry *et al.* Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. **Nature Biomedical Engineering**, Nature Publishing Group, v. 2, n. 10, p. 749, 2018.

MCKINNEY, W. **Python para análise de dados**. *[S.l.]*: Novatec, 2018.

P, LJitendra Nath; SRINIVASAN, Maheshwaran. Predicting probability of loan default. *In*: . *[S.l.: s.n.]*, 2011.

PADOVEZE, C.L. **Manual de contabilidade básica: uma introdução à prática contábil**. Atlas, 1996. ISBN 9788522415199. Disponível em: https://books.google.com.br/books?id=iLU7AAAACAAJ.

PAHWA, Kunal; AGARWAL, Neha. Stock market analysis using supervised machine learning. *In*: **2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)**. *[S.l.: s.n.]*, 2019. p. 197–200.

PIERSON, L.; MACHADO, E.V. **Data Science Para Leigos**. ALTA BOOKS, 2019. ISBN 9788550804804. Disponível em: https://books.google.com.br/books?id=DoAgwQEACAAJ.

PROVOST, F.; FAWCETT, T.; BOSCATO, M. **Data Science Para Negócios**. ELSEVIER/ALTA BOOKS, 2018. ISBN 9788576089728. Disponível em: https://books.google.com.br/books?id=c4lAvgAACAAJ.

QUINLAN, J. R. Induction of decision trees. **MACH. LEARN**, v. 1, p. 81–106, 1986.

RAY, Susmita. A quick review of machine learning algorithms. *In*: **2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)**. *[S.l.: s.n.]*, 2019. p. 35–39.

RIBEIRO, Marco Tulio; SINGH, Sameer; GUESTRIN, Carlos. **"Why Should I Trust You?": Explaining the Predictions of Any Classifier**. 2016.

RUSSELL, S.; NORVIG, P. **Inteligência Artificial**. *[S.l.]*: CAMPUS - RJ, 2004.

SHRIKUMAR, Avanti; GREENSIDE, Peyton; KUNDAJE, Anshul. Learning important features through propagating activation differences. *In*: PRECUP, Doina; TEH, Yee Whye (Ed.). **Proceedings of the 34th International Conference on Machine Learning**. International Convention Centre, Sydney, Australia: PMLR, 2017. (Proceedings of Machine Learning Research, v. 70), p. 3145–3153. Disponível em: http://proceedings.mlr.press/v70/shrikumar17a.html.

TAN, P.N.; STEINBACH, M. .; KUMAR, V. **Introdução ao Data Mining**. *[S.l.]*: Ciência Moderna, 2009.

YAZDI, Kasra Majbouri; YAZDI, Adel Majbouri; KHODAYI, Saeid; HOU, Jingyu; ZHOU, Wanlei; SAEDY, Saeed. Improving fake news detection using k-means and support vector machine approaches. **International Journal of Electronics and Communication Engineering**, World Academy of Science, Engineering and Technology, v. 14, n. 2, p. 38 – 42, 2020. ISSN eISSN: 1307-6892. Disponível em: https://publications.waset.org/vol/158.