UNIVERSIDADE DO VALE DO RIO DOS SINOS – UNISINOS

UNIDADE ACADÊMICA DE EDUCAÇÃO ONLINE

ESPECIALIZAÇÃO EM BIG DATA, DATA SCIENCE E DATA ANALYTICS

Daniel Lessa Januário

PREVENDO E IDENTIFICANDO AS CAUSAS DA EVASÃO DO EMPREGADO COM TÉCNICAS DE APRENDIZADO DE MÁQUINA

São Leopoldo

2019

UNIVERSIDADE DO VALE DO RIO DOS SINOS – UNISINOS

UNIDADE ACADÊMICA DE EDUCAÇÃO ONLINE

ESPECIALIZAÇÃO EM BIG DATA, DATA SCIENCE E DATA ANALYTICS

Daniel Lessa Januário

PREVENDO E IDENTIFICANDO AS CAUSAS DA EVASÃO DO EMPREGADO COM TÉCNICAS DE APRENDIZADO DE MÁQUINA

Trabalho de Conclusão de Curso apresentado como requisito parcial para a obtenção do título de Especialista em *Big Data*, *Data Science* e *Data Analytics*, pelo curso de Pós-Graduação Lato Sensu em *Big Data*, *Data Science* e *Data Analytics* da Universidade do Vale do Rio dos Sinos – UNISINOS.
Orientador: Profa. Dra. Josiane Brietzke Porto

São Leopoldo

2019

Prevendo e Identificando as Causas da Evasão do Empregado com Técnicas de Aprendizado de Máquina

Daniel L. Januário[1]

[1]Universidade do Vale do Rio dos Sinos (UNISINOS)
São Leopoldo – RS – Brazil

daniellessa.j@gmail.com

Abstract. *Employee turnover is considered one of the most common issues handled by companies´ HR department. This scenario impacts companies by both direct cost (e.g. advertising for a job opening and recruitment & selection process) and indirect costs (e.g. learning curve and pressure on remaining staff). Also, the costs persist until the performance of the new employee reaches the same level as the other cooworkers, which affect directlythe efficiency of the company. In this sense, this article aims to predict the employee turnover and identify the main reasons that would cause it. Applying the experimental and statistical research methods, the results obtained with the use of data science techniques it were possible through a predictive model with good generalizability for the worked data set, generating satisfactory answers for the prediction of the target event - employee turnover.*

Resumo. *Atualmente, a evasão de funcionários é um dos problemas mais enfrentados pela área de recursos humanos das empresas. Tal cenário impacta em custos diretos - publicidade para divulgação da vaga, recrutamento e seleção, assim como em custos indiretos – curva de aprendizagem e pressão sobre o pessoal remanescente. Além disso, até que o desempenho do funcionário contratado alcance o nível do empregado que foi desligado, os custos persistem, refletindo diretamente na eficiência da corporação. Sabendo deste contexto, este estudo visa desenvolver um classificador com o intuito de predizer a evasão dos empregados, e identificar as principais razões que levam a tal evento ocorrer, considerando cada instância avaliada pelo modelo. Aplicando os métodos de pesquisa experimental e estatístico, os resultados mostraram que através de técnicas de ciência de dados foi possível produzir um modelo preditivo com boa capacidade de generalização para o conjunto de dados trabalhado, gerando resposta satisfatória para a predição do evento alvo - evasão do funcionário.*

1. Introdução

A área de recursos humanos de uma empresa enfrenta diversos desafios, sendo um deles o de manter um quadro de funcionários qualificados, motivados e que se enquadrem nas necessidades da corporação. Segundo Jantan, Hamdan e Othman (2011), a gestão de talentos envolve decisões gerenciais e são muito incertas e difíceis, pois dependem de vários fatores, tais como experiência humana, conhecimento, preferência e julgamento.

Taxa de evasão (ou em inglês, *turnover*) é definida como a medida da proporção de empregados que acabam desligados da companhia, de forma voluntária ou involuntária, em relação à quantidade total de empregados ativos em um período. No momento que este índice sobe e há a necessidade de substituir os funcionários desligados, eleva-se também o custo envolvido com recursos para o recrutamento [Lu 2014]. Conforme Sikaroudi, Ghousi e Sikaroudi (2015), tais custos podem ser classificados como: custos diretos (e.g., publicidade

da posição, substituição, recrutamento e seleção, pessoal temporário e tempo de gestão) e custos indiretos, os quais estão relacionados à moral, pressão sobre o pessoal remanescente, custos de aprendizagem, qualidade do produto/serviço e memória organizacional. Tais custos continuam até que o funcionário contratado alcance o desempenho do funcionário desligado [De Coninck e Johnson 2009]. Ademais, uma importante relação observada indica que quanto maior a ineficiência da organização, mais próxima a empresa estará de um cenário de alta taxa de *turnover* [Hancock, Allen, Bosco, McDaniel e Pierce 2013].

Nesse cenário, as decisões embasadas por dados são cada vez mais necessárias, uma vez que, saber o que pode ocorrer com os funcionários é tão importante que as companhias podem tomar uma ação de precaução, antes que ocorra a evasão do mesmo. Assim, esse estudo pode ser representando pela seguinte questão de pesquisa: como prever e identificar as principais causas da evasão dos empregados de uma corporação, por meio de aplicação de técnicas de aprendizado de máquina? Seu objetivo principal é desenvolver um modelo que faça a previsão da evasão dos funcionários, assim como compreender os motivos de tal evento, através da aplicação de técnicas de aprendizado de máquina.

Para tanto, o presente trabalho possui como objetivos específicos: *(i)* Enriquecimento do conjunto de dados com a criação de novas informações; *(ii)* Gerar um *score* com a probabilidade tanto da evasão ocorrer quanto de não ocorrer, para cada funcionário avaliado; *(iii)* Determinar, através de inferência estatística, se o modelo gerado originalmente (sem redução dimensionalidade) é tão eficiente quanto um modelo gerado com redução da dimensionalidade, através de *Principal Component Analysis* (PCA).

Dessa forma, sabendo das consequências do problema de *turnover* de empregados, este estudo visa comprovar através de técnicas matemáticas e estatísticas, que é possível prever a evasão dos funcionários de uma corporação, assim como detectar as principais causas deste evento. A partir deste estudo, espera-se contribuir para um melhor entendimento das técnicas de ciência de dados aplicadas sobre o problema de pesquisa abordado.

Esse artigo está organizado da seguinte forma: a Seção 2 trata das definições conceituais dos temas envolvidos na pesquisa. A Seção 3 aborda as pesquisas relacionadas. A Seção 4 descreve o experimento desenvolvido. A Seção 5 informa os resultados obtidos no experimento. E por fim, a Seção 6 é uma nota conclusiva do trabalho de pesquisa.

2. Fundamentação Teórica

Nessa seção são abordadas as definições conceituais dos temas envolvidos na pesquisa, além dos trabalhos relacionados encontrados.

2.1 Tipos de Evasão do Emprego

O subgrupo de gestão de recursos humanos controla o fluxo de *turnover* de uma corporação e tem como papel principal tanto motivar quanto engajar seus funcionários em prol de melhorar a eficácia da força de trabalho. É comum que os cenários das evasões empresariais causem instabilidade nos empregados, o que por sua vez provoca eventuais prejuízos financeiros, perda de oportunidades e de credibilidade. Além disso, tais situações afetam o ambiente interno corporativo, gerando insegurança nos empregados que continuam na companhia. Ao prever tais eventos de evasão, a empresa pode reduzir grande parte da perda futura [Sikaroudi, Ghousi e Sikaroudi 2015].

De acordo com Lu 2014, existem dois principais tipos de evasão do emprego: de forma voluntária e involuntária. A Figura 1 mostra duas possibilidades de *turnover* do funcionário em relação ao seu emprego: de forma voluntária e involuntária. A primeira é quando o empregado deixa o seu trabalho a partir de situações que a empresa não pode controlar. Já a segunda ocorre em função de decisões tomadas pela corporação.

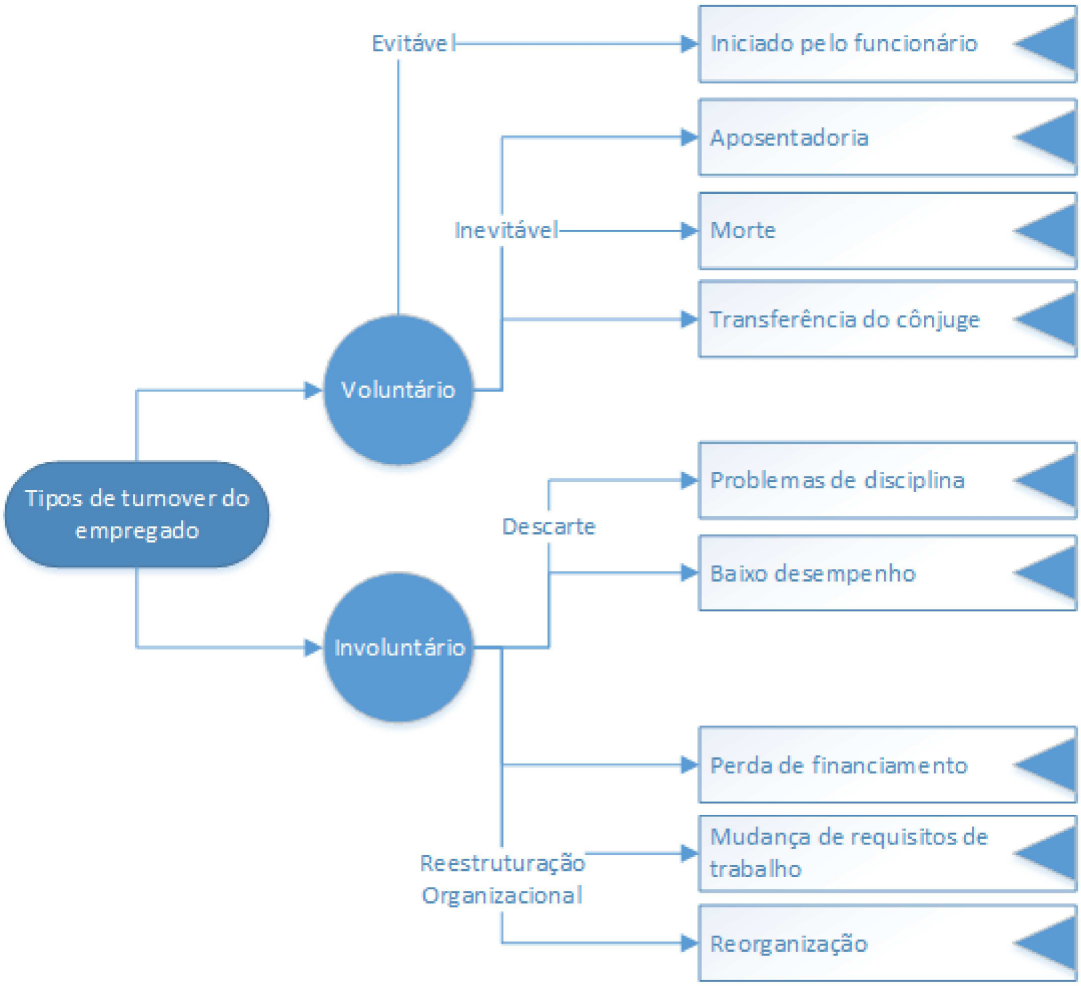

Figura 1. Representação dos tipos de evasão do emprego

Considerando o tipo de evasão voluntária na Figura 1 há duas variações:

- Evitável: quando o próprio funcionário requisita o desligamento;
- Inevitável: ocorre por motivos de aposentadoria, morte ou transferência do cônjuge do funcionário.

Para evasões involuntárias do empregado existem também duas possibilidades:

- Descarte: a empresa demite o empregado em função de problemas de indisciplina ou baixo desempenho no âmbito profissional;
- Reestruturação organizacional: a corporação acaba desligando o funcionário por razões financeiras, mudança de requisitos de trabalho que exijam perfil profissional diferente do que o funcionário pode prover, ou ainda por uma reorganização corporativa ao qual esteja passando a empresa.

2.2. Aprendizado de Máquina

Conforme Facelli, Lorena, Gama e Carvalho (2017), através da crescente complexidade dos problemas a serem resolvidos computacionalmente e do volume exponencial de dados produzidos nos últimos anos, há uma tendência de reduzir a necessidade de intervenção humana nas decisões que dependem de especialistas. Para tanto, as técnicas de aprendizado de máquina são processos indutivos de hipóteses ou por aproximação de função, a partir da experiência passada, capaz de resolver o problema que se deseja tratar.

Na Figura 2 é demonstrada as subdivisões dos principais grupos de aprendizado de máquina. Os problemas de classificação e regressão são solucionados através da aprendizagem supervisionada, assim como os problemas de associação e agrupamento estão associados à aprendizagem não supervisionada.

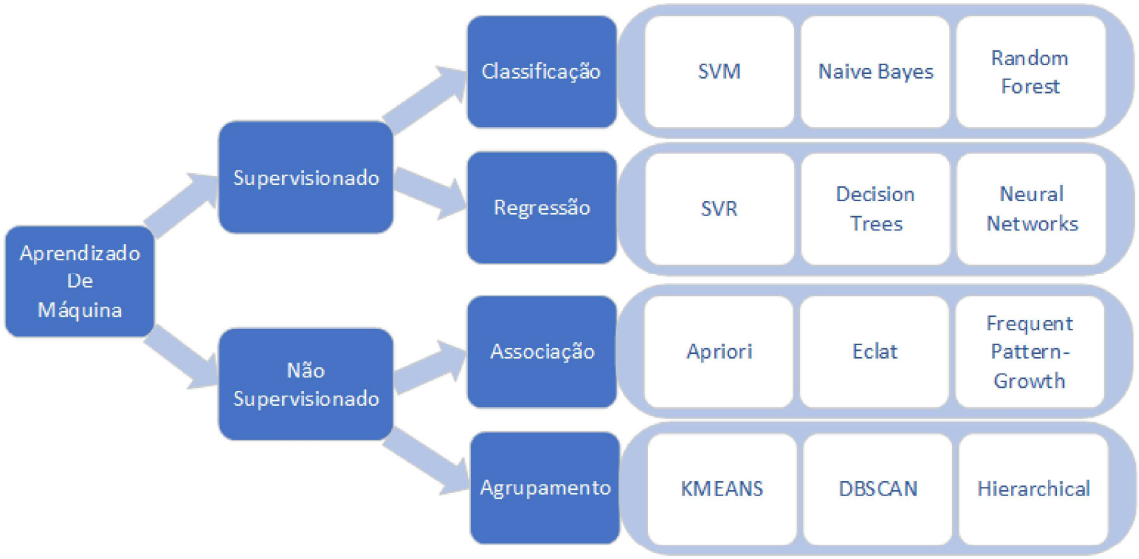

Figura 2. Principais grupos e algoritmos de aprendizado de máquina

No aprendizado supervisionado, o algoritmo adquire conhecimento através de uma coleção de pares de dados (entrada e saída, respectivamente, variáveis preditoras e variável alvo), onde o objetivo é produzir um modelo que faça as previsões da variável de saída (alvo). Já o aprendizado não supervisionado, tem como objetivo construir um modelo a partir de um conjunto de dados de entrada sem necessidade de um valor conhecido (e.g., variável alvo). Por exemplo, através de algoritmos não-supervisionados, se pode identificar as características dos clientes de uma loja [Burkov 2019].

2.3. Aprendizado Supervisionado

Nessa seção aborda-se o conceito dos algoritmos de aprendizagem supervisionada, aplicados no experimento deste artigo.

2.3.1. Algoritmos de Classificação

Pode-se aplicar essa modalidade de algoritmo supervisionado quando o objetivo da classificação for um conjunto finito de valores, ou ainda um booleano (Verdadeiro ou Falso). Um exemplo seria a classificação de animais, tendo como dados de entradas as suas características, e a saída a classe predita: cachorro, gato ou pássaro [Russel e Norvig 2013].

Nas subseções a seguir descreve-se o conceito dos principais algoritmos da categoria, que são aplicados no experimento deste artigo.

2.3.1.1. *Bagging Classifier*

Esse classificador produz replicações do conjunto de dados de treinamento por amostragem com reposição, onde todas têm o mesmo tamanho. Cada réplica do conjunto é entregue à um classificador efetuar o seu treinamento, fazendo com que N classificadores trabalhem em N conjuntos de dados, porém para o mesmo problema. Após, para obter a classificação final do modelo em conjunto com os N classificadores é aplicado um esquema de voto uniforme. Se um dos classificadores está abaixo do limiar esperado, este é deixado de fora do conjunto gerado [Facelli, Lorena, Gama e Carvalho 2017].

2.3.1.2 *Gradient Boosting Classifier*

Segundo Rao, Shi e Rodrigue (2018) é um método *ensemble*, derivado do *Gradient Descent*, *Boosting* e Árvore de Regressão, que utiliza técnica estatística não paramétrica. Gera N estimadores que realizam as previsões, passando por um processo de minimização dos resíduos gerados por cada modelo - diferença entre valor previsto e o valor observado, aplicando *Gradient Descent* para reduzir o erro das previsões.

Em outras palavras, cada modelo gera uma saída, onde os erros são aprendidos pelo próximo estimador. Logo, cria-se uma cadeia de estimadores que irão aprendendo com os erros anteriores. É um algoritmo robusto e que requer maior poder computacional.

2.3.1.3. *Random Forest Classifier*

É um modelo *ensemble* que combina classificadores homogêneos, que utiliza a abordagem de divisão e conquista para melhorar o desempenho do algoritmo. Trata-se de um conjunto de modelos de aprendizado de máquina individuais (no caso, árvores de decisão), que são unidos, criando ao final, um único modelo [Zhao, Hryniewicki, Cheng, Fu, Zhu 2018].

Segundo Facelli, Lorena, Gama e Carvalho (2017), a quantidade de árvores a serem criadas deve ser definida, onde para cada árvore de decisão, dinamicamente amostras de M preditores são escolhidos como candidatos ao conjunto completo P. Para que ocorra essa escolha, é considerado o voto majoritário de diferentes hipóteses, onde a variabilidade dos classificadores é reduzida. Sabendo que cada árvore criada gera um estimador e que cada estimador gera previsões, essas são combinadas para se obter o melhor resultado.

Entre as vantagens pode ser usada para identificar as características mais importantes do conjunto de dados, que levam a classificação da classe, além da fácil interpretação do modelo gerado assim como é um modelo robusto mesmo em dados com ruídos. No que tange às desvantagens, não é uma técnica ideal para variáveis categóricas com diferentes níveis [Veldhuis 2017].

2.3.1.4. *Logistic Regression*

A regressão logística binária objetiva modelar a relação entre a variável dependente (ou resposta) e suas respectivas variáveis independentes (ou explicativas), que podem ser numéricas e/ou categóricas. A variável resposta deve ser um valor binário, que assume duas possibilidades: 0 (falso) e 1 (verdadeiro, que é o evento de interesse).

O modelo é representado pela seguinte fórmula, onde ***b*** representa os parâmetros e ***X*** representa as variáveis dependentes:

$$logit(p) = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + \ldots + b_kX_k$$

Onde ***p*** é a probabilidade de um evento ocorrer.

$$odds = \frac{p}{1-p} = \frac{probability\ of\ presence\ of\ characteristic}{probability\ of\ absence\ of\ characteristic}$$

Odds, presente na fórmula anterior, representa a taxa entre a probabilidade que um evento possa ocorrer, p(Y = 1), sobre a probabilidade de o mesmo evento não ocorrer.

A definição da probabilidade de um evento ocorrer é definida pela fórmula abaixo:

$$p = \frac{1}{1 + e^{-logit(p)}}$$

Esse algoritmo é semelhante a regressão linear, fornecendo os coeficientes das variáveis dependentes para prever a probabilidade de presença da característica de interesse para um determinado evento [Veldhuis 2017].

2.3.1.5. *Linear Discriminant Analysis*

Segundo Tharwat, Gaber, Ibrahim e Hassanien (2017), esse algoritmo utiliza técnicas de estatística multivariada para discriminar e fazer a classificação de instâncias. Ele modela a distribuição das variáveis preditoras agrupando-as por cada possibilidade de classe, e então aplica o teorema de *Bayes* para estimar a probabilidade de um evento ocorrer.

O objetivo do algoritmo é ter maior poder de discriminação entre as populações avaliadas, buscando minimizar a probabilidade de má classificação, através da combinação linear das características observadas.

2.3.1.6. *K-Neighbors Classifier*

É um algoritmo não paramétrico que mede a distância (normalmente, a Euclidiana) da instância a ser predita em relação a todos os demais pontos já classificados. A partir disso, são identificados os K pontos mais próximos, onde K é a quantidade de vizinhos, para que através de uma votação majoritária destes, eleja-se o K dessa nova instância, prevendo o valor da sua classe com base na de seus vizinhos [Zhao, Hryniewicki, Cheng, Fu, Zhu 2018]. Essa eleição toma como base a classe que mais apareceu entre os K-vizinhos mais próximos da instância a ser classificada.

2.3.1.7. *Decision Tree Classifier*

Os métodos de árvores de decisão são conceitualmente fáceis de entender e interpretar, capazes de lidar com valores faltantes e com eficiente seleção de atributos [Zhao, Hryniewicki, Cheng, Fu, Zhu 2018].

Apresenta o formato de uma árvore, porém invertida: temos a raiz na parte superior e o nodo folha na extremidade inferior, que é onde ocorre a classificação. Basicamente, ela desenvolve um conjunto de regras - representam as condições de checagem, que visam uma

decisão para possíveis respostas que o atributo pode assumir. Os ramos interligam os nós, que servem como um caminho percorrido para o processo de decisão. Ao final da árvore, os nós folha - representam as saídas, ou a variável resposta do modelo, indicando a classe da instância avaliada [Alao, Adeyemo 2013].

Para classificação binária de classes, é possível usar a implementação padrão do algoritmo. Caso seja um problema de multi-classes, há a implementação com o algoritmo c4.5.

2.3.1.8. *Gaussian NB*

Conforme Valle e Ruz (2015), este algoritmo não considera a correlação entre as variáveis, por isso é conhecido como *Naive* (ingênuo). Baseia-se na independência entre as características, e usa o teorema da probabilidade de *Bayes* para predizer a classe de uma nova instância. Segue a fórmula matemática:

$$P(a \mid b) = P(b \mid a) \times P(a) / P(b)$$

Onde:

- P(a | b) é a probabilidade do evento "a" ocorrer em decorrência do evento "b";
- P(b | a) é a probabilidade do evento "b" ocorrer em decorrência do evento "a";
- P(a) é a probabilidade original do evento "a" ocorrer em relação à todas instâncias do conjunto de dados;
- P(b) é a probabilidade original do evento "b" ocorrer em relação à todas instâncias do conjunto de dados.

Por se tratar de um algoritmo que classifica rapidamente as instâncias, pode ser usado em previsões em tempo real. Assim como é indicado para classificação de textos e filtragem de *spam*.

2.3.1.9. *Support Vector Machine*

Support Vector Machine (SVM) são classificadores versáteis, que se ajustam a modelos lineares e não lineares. Num primeiro momento, este algoritmo faz a classificação das instâncias com base nos dados de treino. Após isso, procura encontrar uma linha de separação entre as classes, chamada de hiperplano, que visa maximizar a distância entre as classes [Ying 2017].

Conforme Sisodia, Vishwakarma, Pujahari (2018), a distância entre o hiperplano e o primeiro ponto encontrado nas extremidades de cada classe é chamada de margem. Já os pontos de dados que ficam sobre as margens, são chamados de vetores de suporte. A classificação das novas instâncias irá se apoiar do hiperplano gerado pelo modelo.

2.4. Aprendizado Não Supervisionado

Nesse item são abordados os conceitos dos algoritmos de aprendizagem não supervisionada, aplicados no experimento deste artigo.

Algoritmos de clusterização ou de agrupamento geralmente são usados em problemas onde se quer obter o conhecimento amplo sobre grupos com características em comum. A seguir é descrito o conceito do algoritmo *Kmeans*, que foi aplicado no experimento deste artigo.

Segundo Pednekar (2019), este algoritmo busca padrões através da similaridade recorrente encontrada em grupos de dados, que são chamados de clusters. Por se tratar de um modelo não supervisionado e não possuir a informação da classe das instâncias, é aplicado para efetuar a descoberta do conhecimento e obter o rótulo de cada elemento.

As instâncias de um cluster devem ter uma alta similaridade entre si - grupos homogêneos, porém diferentes entre os demais clusters. Através do cálculo da distância Euclidiana, temos a forma de medir a similaridade entre as instâncias e os *K-centroides*, onde a menor distância entre estes definirá o cluster de cada elemento.

Segue a fórmula para o cálculo da distância Euclidiana:

$$distEuclidiana = \sqrt{(|Xi1-Xj1|^2 + |Xi2-Xj2|^2 \ldots + |Xin-Xjn|^2)}$$

Onde efetua-se a raiz quadrada da soma das diferenças entre os atributos Xi e Xj.

Conforme Bholowalia e Kumar (2014), o algoritmo *Kmeans* é sensível a quantidade de clusters que devem ser produzidos para análise. Caso um valor não adequado seja indicado como quantidade de clusters para produção do modelo, corre-se o risco de afetar diretamente a homogeneidade dos grupos assim como a generalização em relação às características dos clusters gerados.

Uma das formas de determinar o valor ideal para o conjunto de dados que será avaliado, é através do método de *Elbow*. Este consiste em avaliar a taxa da variância em relação a quantidade de agrupamentos, onde o ponto ideal é quando a variabilidade dos dados em relação ao número de clusters não seja significativo.

2.5. Redução da Dimensionalidade com PCA

A dimensionalidade é a quantidade de atributos de um conjunto de dados que representam as suas características. Conforme aumenta a quantidade de atributos a serem processados, maior a complexidade das técnicas de aprendizado de máquina, impactando negativamente o desempenho dos algoritmos. Logo, a motivação para reduzi-la no campo da ciência de dados, seria otimizar o custo de medição e precisão do modelo. Para tanto, o objetivo é representar o conjunto de dados original em um novo plano com dimensionalidade reduzida, mantendo as características dos dados [Facelli, Lorena, Gama e Carvalho 2017].

Segundo Aït-Sahalia e Xiu (2017), *Principal Component Analysis* (PCA) usa o método de extração de atributos (variáveis importantes) através de técnicas da estatística multivariada, que visam transformar o valor dos atributos do conjunto de dados original em outro conjunto de variáveis agrupadas, com um número menor de dimensões, denominadas de componentes principais.

Para cada componente principal criado, há uma combinação linear de N características, onde o primeiro componente principal terá a maior variância nos dados. Conforme aumenta a variabilidade capturada, cresce também a quantidade de informações capturadas pelo componente principal. A partir do segundo componente, estes capturam a variabilidade remanescente do conjunto de dados original. Logo, quanto maior a quantidade de componentes principais, há uma queda gradual esperada na variabilidade dos dados de cada componente subsequente [Øyvind, Myklebust, Hallén e Federolf 2017].

Conforme Rousson e Gasser (2004), os componentes gerados pelo PCA não permitem a fácil interpretabilidade dos dados: caso essa seja uma premissa do projeto, o PCA não deve ser utilizado.

2.6. Validação Cruzada para Avaliação dos Algoritmos

Na validação cruzada. o conjunto de dados é dividido em K subconjuntos de dados de tamanhos aproximadamente iguais. Aleatoriamente, uma das K-partições é selecionada para ser usada como validação do modelo. As demais K-partições serão usadas como subconjunto de dados para treinar o modelo. Esse fluxo se repete até que todas as partições tenham sido usadas como validação do modelo. O desempenho final é dado pela média dos desempenhos observados sobre cada subconjunto (ou partição) de dados [Li e Cui 2019].

Uma das críticas quanto à essa técnica é que quando K>2, não se tem independência completa entre os subconjuntos de treinamento: parte dos dados de treino são compartilhadas entre as partições [Facelli, Lorena, Gama e Carvalho 2017]. A Figura 3 demonstra o processo de validação cruzada, com K=3. A média das validações do estimador irá gerar a acurácia do modelo.

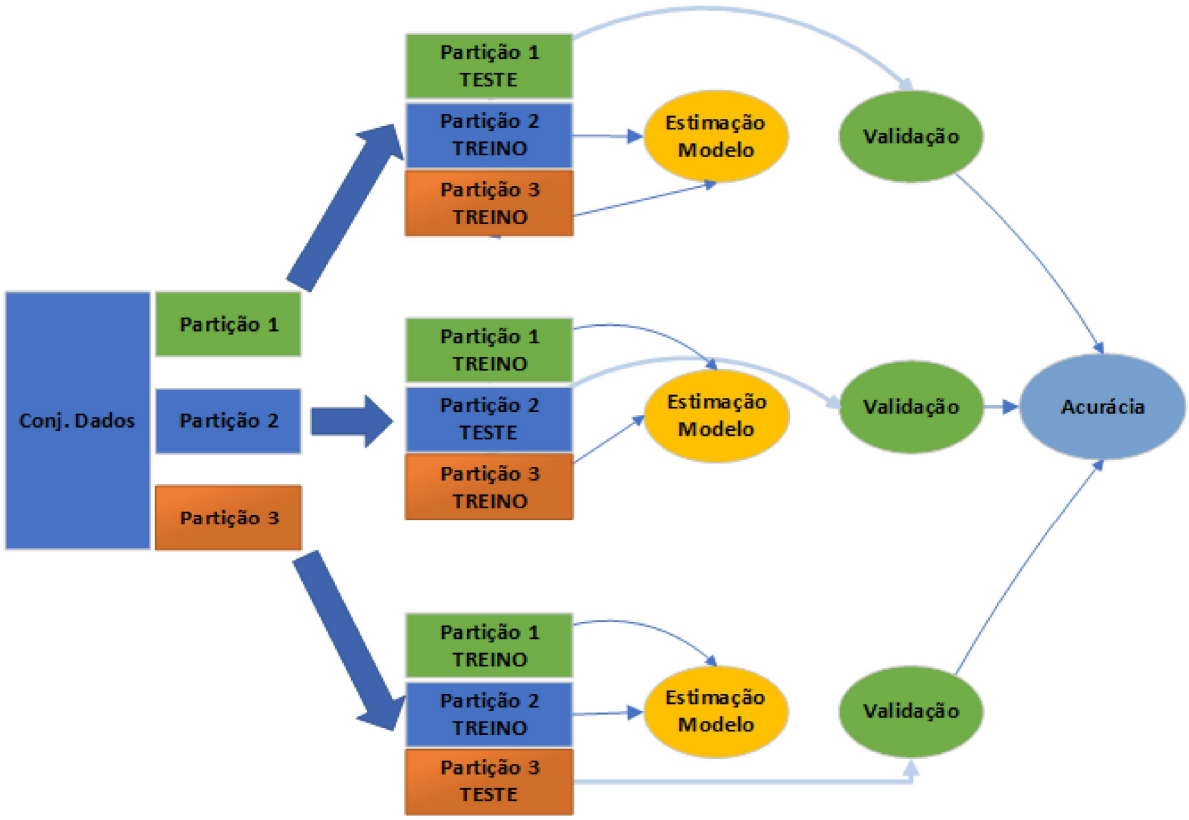

Figura 3. Exemplo de validação cruzada com K=3.

2.7. Métricas para Avaliação de Modelos Preditivos de Classificação

Para compreender a eficiência de um modelo, é necessário avaliar algumas métricas. Basicamente, estes indicadores partem da matriz de confusão, sendo essa a forma de avaliar os classificadores.

Conforme Zeng (2019), a matriz de confusão mede a precisão de um modelo comparando valores previstos com valores reais, possibilitando encontrar taxas de erro e acerto. A Figura 4 ilustra a estrutura de uma matriz de confusão binária.

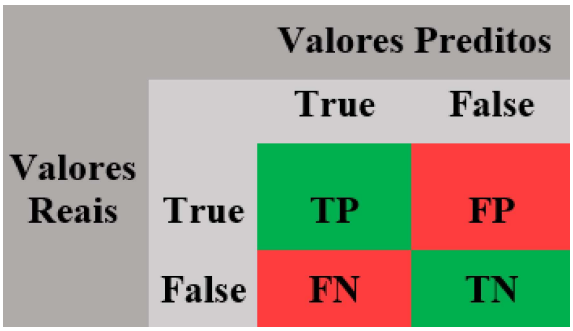

Figura 4. Matriz de confusão binária e suas respectivas categorias

Onde:

- *True Positive* (TP) é o número de positivos que estão corretamente classificados como positivos;
- *False Positive* (FP) é o número de negativos que estão incorretamente classificados como positivos;
- *True Negative* (TN) é o número de negativos que estão corretamente classificados como negativos;
- *False Negative* (FN) é o número de positivos que estão incorretamente classificados como negativos;

O Quadro 1 descreve o conceito das principais métricas para avaliar modelos preditivos voltados à problemas de classificação.

Métricas de Classificação	**Descrição**
Acurácia (*Accuracy*)	Proporção de instâncias classificadas corretamente (TP + TN) em relação ao total de instâncias preditas [Yousaf 2016]. $Accuracy = (true\ positives + true\ negatives) / (total\ of\ prediction)$
Precisão ou Taxa Preditiva Positiva (*Precision*)	Proporção de instâncias classificadas como positivas (TP) em relação ao domínio de instâncias que realmente são positivas (TP + FP) [Burkov, 2019]. $Precision = true\ positives / (true\ positives + false\ positives)$
Sensibilidade ou Taxa de Verdadeiros Positivos (*Sensitivity* ou *Recall*)	Proporção de instâncias corretamente classificadas como positivas (TP) em relação ao domínio de instâncias preditas como positivas (TP + FN). Essa medida indica o quão eficiente é o modelo em reconhecer amostras positivas [Alao, Adeyemo 2013]. $Sensitivity = true\ positives / (true\ positives + false\ negatives)$
Especificidade ou Taxa de Verdadeiros Negativos (*Specificity*)	Proporção de instâncias corretamente classificadas como negativas (TN) em relação ao domínio de instâncias preditas como negativas (TN + FP). Essa medida indica o quão eficiente é o modelo em reconhecer amostras negativas [Yousaf 2016]. $Specificity = true\ negatives / (true\ negatives + false\ positives)$
F1-Score ou *F-Measure*	Mensura a proporção entre as medidas *Precision* e *Recall*, considerando o balanço das taxas preditiva positiva e verdadeiros positivos [Lu 2014]. $F1\text{-}Score = 2 \times ((Precision \times Recall) / (Precision + Recall))$
Área Sob a Curva ROC (*Area Under The ROC Curve* (*AUC*))	Tem como objetivo mensurar a probabilidade do modelo classificar um exemplo positivo aleatório mais do que um exemplo negativo aleatório. Considera as medidas das taxas de verdadeiros positivos (sensibilidade) e verdadeiros negativos (especificidade) para projetar o gráfico da área sob a curva *ROC* [Yedida, Reddy, Vahi, Jana, Gv e Kulkarni 2018].

Quadro 1. Métricas para avaliação de modelos preditivos em problemas de classificação

Tais métricas são extraídas do modelo preditivo a partir da apresentação de amostras de teste, e seus valores devem satisfazer os limiares estabelecidos para que o modelo possa ser considerado apropriado à resolução do problema.

2.8. Métricas para Avaliação de Modelos Preditivos de Agrupamento

Uma das medidas usadas para compreender a eficiência de um modelo de agrupamento é o coeficiente de silhueta. A seguir, é descrito o conceito desta métrica.

2.8.1. Coeficiente de Silhueta

Segundo Starczewski e Krzy˙zak (2015), representa uma medida de quanto um objeto é semelhante ao seu próprio cluster comparado aos outros clusters. A fórmula para calcular o coeficiente de silhueta pode ser representada a seguir:

$$S(\mathbf{x}) = \frac{b(\mathbf{x}) - a(\mathbf{x})}{max\left(a(\mathbf{x}), b(\mathbf{x})\right)}$$

Onde:

- a(x) é a distância média intra-cluster (densidade dos elementos de um mesmo grupo, ou seja, o quão próximos estão os elementos de um mesmo grupo);
- b é a distância média entre todas as amostras e o cluster mais próximo do qual as amostras não fazem parte;
- A escala do resultado do coeficiente de silhueta fica entre -1 e 1, onde quanto maior este valor, maior a densidade entre os elementos do mesmo grupo em relação aos demais clusters (distância intra-cluster menor e distância inter-cluster maior).

2.9. Teste de Hipótese

Conforme Sweeney, Williams e Anderson, teste de hipótese é aplicado quando se deseja comprovar uma hipótese científica baseado em dados amostrais, assumindo um risco controlado acerca do estudo populacional. Os requisitos necessários para proceder com a interpretação de um teste de hipótese com *t-student* são: a hipótese e o nível de significância.

2.9.1. A Hipótese

A hipótese científica se fundamenta em um problema e na sua resolução, tendo as seguintes possibilidades:

- Nula (H0): se rejeitada, há confirmação da hipótese científica.
- Alternativa (Ha): se a hipótese nula for rejeitada, a hipótese alternativa é aceita como hipótese científica válida.

Importante lembrar que se não houver a rejeição da hipótese nula, o estudo torna-se inconclusivo.

2.9.2. Nível de Significância

É a margem de erro tolerável que sustentará a rejeição da hipótese nula. Quando o teste estatístico ocorre sobre uma amostra dos dados, há possibilidade de ocorrerem erros. São estes:

- Erro tipo I: rejeitar H0 na amostragem de dados, porém a hipótese de nulidade é aceita para a população.
- Erro tipo II: não rejeitar H0 na amostragem de dados, porém a hipótese de nulidade é rejeitada para a população.

A Figura 5 a seguir ilustra as possibilidades de tipos de erro em testes de hipóteses:

Figura 5. Tipos de erro em testes de hipóteses

Quanto menor o nível de significância, maior a probabilidade de ocorrer o erro tipo II. Para valores maiores, a tolerância à erros será proporcional.

2.9.3. Teste de Hipóteses para Verificação de uma Média com *T-Student*:

O teste T de *Student* pode ser usado na comparação de apenas duas médias, e suas variações são em relação às hipóteses testadas. Segundo Kuyven (2010), os passos para o teste de hipótese são os apresentados a seguir:

1º) Formulação das hipóteses nula e alternativa a respeito de um parâmetro populacional.

2º) Determinação (escolha) do nível de significância do teste.

3º) Obtenção de uma amostra aleatória que permita obter uma estimativa pontual do parâmetro populacional que está sendo testado.

4º) Cálculo da estatística amostral do teste, considerando a fórmula abaixo:

$$t_{calc} = \frac{\overline{X} - \mu_0}{S/\sqrt{n}}$$

Onde:

- $\overline{X}$ é a média amostral.
- μ é o valor do teste unilateral.
- S é o desvio padrão amostral.
- N é o número de graus de liberdade, dado pela quantidade de elementos da amostra subtraída de 1.

5º) Determinação da região de rejeição da hipótese nula do teste e, quando possível, do valor-p associado à estatística do teste.

6º) Conclusão formal do teste. Abaixo, segue regra de decisão (pela região de rejeição de H0):

- Se |estatística do teste| >= |valor crítico|, rejeita-se H0 e é informado na conclusão que há evidências suficientes para afirmar H1.

Se |estatística do teste| < |valor crítico|, aceita-se H0 e é informado na conclusão que não há evidências suficientes para afirmar H1.

2.9.4. Valores da Tabela T

O Figura 6 abaixo representa os valores críticos de T de *Student* para testes unilaterais, considerando os níveis de significância de 0,0005 (0,005 %) até 0,1 (10%).

ν \ α	0.1	0.05	0.025	0.01	0.005	0.001	0.0005
1	3.078	6.314	12.076	31.821	63.657	318.310	636.620
2	1.886	2.920	4.303	6.965	9.925	22.326	31.598
3	1.638	2.353	3.182	4.541	5.841	10.213	12.924
4	1.533	2.132	2.776	3.747	4.604	7.173	8.610
5	1.476	2.015	2.571	3.365	4.032	5.893	6.869
6	1.440	1.943	2.447	3.143	3.707	5.208	5.959
7	1.415	1.895	2.365	2.998	3.499	4.785	5.408
8	1.397	1.860	2.306	2.896	3.355	4.501	5.041
9	1.383	1.833	2.262	2.821	3.250	4.297	4.781
10	1.372	1.812	2.228	2.764	3.169	4.144	4.587
11	1.363	1.796	2.201	2.718	3.106	4.025	4.437
12	1.356	1.782	2.179	2.681	3.055	3.930	4.318
13	1.350	1.771	2.160	2.650	3.012	3.852	4.221
14	1.345	1.761	2.145	2.624	2.977	3.787	4.140
15	1.341	1.753	2.131	2.602	2.947	3.733	4.073
16	1.337	1.746	2.120	2.583	2.921	3.686	4.015
17	1.333	1.740	2.110	2.567	2.898	3.646	3.965
18	1.330	1.734	2.101	2.552	2.878	3.610	3.922
19	1.328	1.729	2.093	2.539	2.861	3.579	3.883
20	1.325	1.725	2.086	2.528	2.845	3.552	3.850
21	1.323	1.721	2.080	2.518	2.831	3.527	3.819
22	1.321	1.717	2.074	2.508	2.819	3.505	3.792
23	1.319	1.714	2.069	2.500	2.807	3.485	3.767
24	1.318	1.711	2.064	2.492	2.797	3.467	3.745
25	1.316	1.708	2.060	2.485	2.787	3.450	3.725

Figura 6. Valores críticos de T de *Student* para testes unilaterais

2.10. Metodologia para Desenvolvimento de Projetos de Ciência de Dados - CRISP-DM

Conforme Shafique e Qaiser (2014), trata-se de um *framework* construído para prover uma estrutura com diretrizes para projetos de mineração de dados. Tal *framework* é constituído de 6 fases, exibidas na Figura 7 [Hui, p. 7, 2019].

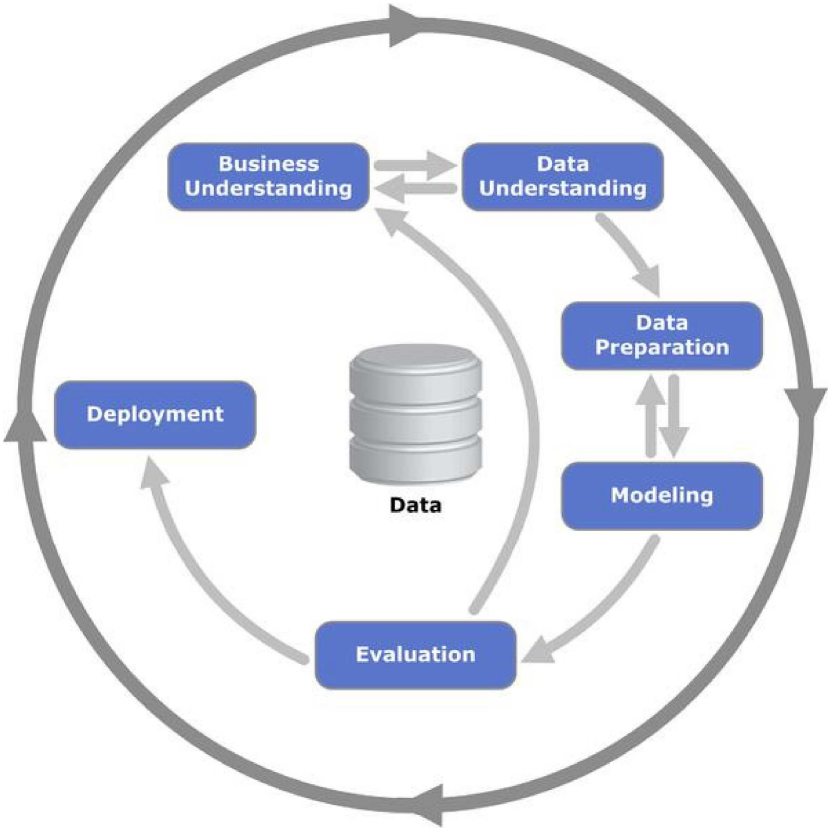

Figura 7. Etapas do modelo CRISP-DM

Seguem as fases do modelo [Shafique e Qaiser, 2014]:

1º) *Business Understanding*: entendimento dos requisitos e objetivos do projeto através da perspectiva do negócio, definindo o problema a ser solucionado, os critérios de sucessos, assim como entender bem as terminologias e termos técnicos do ramo onde a mineração será aplicada.

2º) *Data Understanding*: aquisição da coleção de dados com o intuito de verificar a qualidade dos dados, assim como explorá-los na tentativa de captar *insights* para a geração de hipóteses que possam levar às respostas do problema do negócio.

3º) *Data Preparation*: seleção e preparação dos dados que formarão o conjunto de dados final. Essa fase inclui diversas etapas, tais como:

- *Data Cleaning*: detecção, remoção e ajuste de anomalias contidas nos dados.
  - Valores fora da curva (*ouliers*).
  - Nan (NULL): valores ausentes.
  - Representações duplicadas inesperadas.
- *Data Wrangling* (ou *Munging*): transformando dados em um formato que facilita o trabalho para o modelo de mineração de dados.
  - Particionamento dos dados (criação do conjunto de dados de treino, validação e de teste).
  - Transformações (normalização, padronização, ajuste de escala, binarização, pivoteamento e outras).

    - Substituição de dados (corte, divisão, fusão, codificação de valores categóricos e outras).
    - Ponderação e Seleção (ponderação dos atributos, otimização automática, entre outras).
    - Geração de Atributos (geração de IDs, composição de atributos, entre outras).
- *Feature Engineering*: seleção de atributos para analisar as que mais impactam na resolução do problema.
    - Debate com especialistas do negócio ou teste dos recursos informacionais.
    - Seleção de recursos.
    - Validação de como os recursos funcionam com seu modelo.
    - Incremento de recursos informacionais, se necessário.
    - Retomada do debate com especialistas do negócio para criação de mais recursos.

4º) *Modeling*: seleção e aplicação de técnicas de modelagem de dados. Nessa fase, são construídos "N" modelos, onde é possível calibrar as configurações dos parâmetros, a fim de encontrar os valores que melhor se ajustam aos dados contidos para o algoritmo escolhido.

5º) *Evaluation*: nessa fase, o foco é na avaliação dos modelos gerados. A interpretação depende do(s) algoritmo(s) aplicado(s), e os resultados são confrontados com os objetivos do projeto.

6º) *Deploy*: essa fase foca na organização, construção e apresentação do relatório dos resultados, assim como na definição de como será usado o conhecimento obtido com a construção do modelo final.

2.10.1. Fase de Preparação dos Dados

Contida na 3ª fase do framework CRISP-DM, essa é uma das etapas de maior trabalho em um projeto de ciência de dados.

Segundo Facelli, Lorena, Gama e Carvalho (2017, p. 29), um conjunto de dados pode conter "*ruídos ou imperfeições, valores incorretos, inconsistentes, duplicados ou ausentes; os atributos podem ser independentes ou relacionados; os conjuntos de dados podem apresentar muitos ou poucos objetos*". Os procedimentos de preparação dos dados são aplicados para [Facelli, Lorena, Gama e Carvalho, p. 29, 2017]: "*melhorar a qualidade dos dados por meio da eliminação ou minimização de problemas citados*", onde levam "*à construção de modelos mais fiéis à distribuição real dos dados*". Também é importante citar que há técnicas para adequar os dados para a utilização de determinado(s) algoritmo(s).

A seguir são apresentados os problemas comumente encontrados na fase de preparação dos dados em projetos de ciência de dados e alternativa(s) de ajuste.

2.10.2. *Outliers*

A partir do gráfico de histograma é possível avaliar a distribuição de dados quantitativos. Conforme Sweeney, Williams e Anderson (2014), *Z-Score* é uma medida de distribuição relativa que pode ser interpretada como a quantidade de desvios padrões que determinado valor está afastado da média dos valores. É uma das formas de identificar se determinado valor de uma observação será considerado ou não um *outlier*. Para calcular o *Z-Score* de cada valor da amostra (coluna por coluna), é considerada a fórmula abaixo:

$$Z\text{-Score} = ((Xi - \mu) / \sigma))$$

Onde:

- Xi é o valor da posição.
- é a média dos valores para a coluna em questão.
- σ é o desvio padrão dos valores para a coluna em questão.

Se o valor do *Z-Score* calculado para cada posição relativa for menor ou igual a -3 ou maior ou igual a +3, então o registro é considerado um *outlier*. Após a identificação dessa característica, há algumas alternativas para lidar com esse problema, ao qual Facelli, Lorena, Gama e Carvalho (2017) compreendem da seguinte forma:

- Técnicas de encestamento: suavização do valor de um atributo com a média ou a mediana dos valores contidos em sua respectiva cesta de valores - as instâncias são divididas em faixas de valores previamente ordenadas, compondo cestas com a mesma quantidade de observações.
- Técnicas baseadas em regressão ou classificação: dado um valor quantitativo com ruído, a função de regressão consegue estimar o seu valor real. Caso o valor a ser estimado seja simbólico, técnicas de classificação podem ser aplicadas.

Uma outra abordagem é eliminar as observações do conjunto de dados, porém, não é indicada quando poucos atributos possuem valores fora da curva.

2.10.3. Valores Ausentes

Caracteriza-se pela ausência de valores para atributos de algumas instâncias [Sweeney, Williams e Anderson 2014]. Alternativas para este problema são:

- Eliminar as observações do conjunto de dados principalmente quando o atributo com valor ausente indica a classe da instância (se for um problema de classificação). Não é indicado eliminar as observações, quando poucos atributos possuem valores ausentes;
- Preencher manualmente os valores ausente. Alternativa não factível, quando a quantidade de ocorrências do problema é grande;
- Utilizar alguma heurística de preenchimento automático dos valores ausentes (média, mediana, moda);
- Aplicar técnicas e algoritmos de aprendizado de máquina para preenchimento dos valores ausentes.

2.10.4. Registros Duplicados

Caracteriza-se por um conjunto de dados ter tanto observações como atributos redundantes [Sweeney, Williams e Anderson 2014]. Alternativas para este problema são identificar e eliminar as observações redundantes do conjunto de dados.

2.11. Portal *Kaggle*

O *Kaggle* tem como propósito geral apoiar a comunidade de ciência de dados. É um portal com ambiente *Jupyter Notebook* para execução de códigos em linguagem de programação R e Python, sendo mantido pelo *Google*. Além de interface para execução de códigos fonte, serve como repositório para a comunidade publicar seus conjuntos de dados e códigos, para acesso ao público em geral.

Também conta com duas trilhas: uma para aprendizado e outra para competições. A trilha de aprendizado introduz os principais conceitos da área e reserva parte para prática em exercícios direcionados, conforme a evolução no conteúdo. Já, na trilha de competição é possível participar de desafios diversos de forma individual ou em times, onde a resolução e a eficiência são o objetivo dos participantes. Ao final, o modelo é submetido e testado: a posição do ranking é baseada na eficiência do modelo.

3. Pesquisas Relacionadas

Nessa seção são apresentados trabalhos relacionados ao tema, compreendendo o contexto do estudo e parâmetros aplicados.

3.1. *Using Decision Tree to Analyze the Turnover of Employees*

Ying (2017) levanta os seguintes problemas de pesquisa: quais fatores tem influência na evasão do funcionário? Como prever a evasão do empregado? Dentro dos principais problemas apontados para identificar as causas da evasão do empregado, estão:

- Ruídos durante conversa de desligamento: normalmente, as empresas conversam com os funcionários durante o período de desligamento para entender as razões de uma evasão. Nem sempre o empregado expressa o real sentimento, gerando uma resposta não muito clara sobre os motivos do seu desligamento.
- Grande parte das tomadas de decisões são influenciadas por fatores subjetivos.

A seguir, alguns pontos que levaram o autor a focar apenas em algoritmos de árvore de decisão:

- Velocidade na geração do modelo de classificação.
- Facilidade na interpretação das regras em formato estruturado.
- É um algoritmo que normalmente tem bons resultados em conjuntos de dados de recursos humanos.
- Através do uso da entropia e taxa de ganho, identifica as variáveis preditoras e seus respectivos pesos, facilitando a identificação das características mais importantes.

3.2. *Predicting Voluntary Turnover Through Human Resources Database Analysis*

Rombaut e Guerry (2018) constataram um problema recorrente da área de recursos humanos, para prever o *turnover* voluntário: se baseiam nas pesquisas aplicadas em seus funcionários periodicamente. Tal metodologia contêm uma baixa taxa de resposta, ou seja, não são eficazes devido à insuficiência e qualidade das respostas, o que não permite tirar conclusões generalizadas.

A proposta deste estudo é poder prever a evasão voluntária do funcionário através de métodos quantitativos sem necessitar de pesquisas suplementares. Para tanto, são aplicadas técnicas de mineração de dados, usando o algoritmo de regressão logística, tanto para avaliar as características mais importantes, quanto para classificar as instâncias.

3.3. *Employee Turnover Prediction Using Machine Learning Based Methods*

Lu (2014) elaborou um estudo sobre a predição da evasão do empregado, considerando os contextos abaixo:

- Histórico de transição de carreira: são mapeados e construídos grafos da transição da carreira de cada funcionário, possibilitando estruturar o histórico de cargos, período

de tempo e empresas pelo qual passou o empregado. Tais históricos são coletados de redes sociais e, a partir dos grafos gerados, o autor enriquece a base de dados com os seguintes atributos: quantidade de vizinhos para cada nodo, grau dos vértices de entrada e saída, similaridade média de Jaccard de um nó e seus vizinhos;

- Informações pessoais e desenvolvimento acadêmico.

O autor desenvolveu um classificador binário, baseado em um problema de aprendizado de máquina supervisionada, onde o objetivo é avaliar se o empregado irá evadir ou não a empresa. Conclui-se que a abordagem de classificação coletiva do experimento – dados demográficos e da rede social do empregado, fornecem melhor acurácia comparada aos modelos tradicionais.

3.4. Análise Comparativa

O Quadro 2, a seguir exibe um comparativo entre as pesquisas relacionadas supracitadas, considerando as técnicas utilizadas nos experimentos.

Referência	Técnicas de Preparação	Algoritmos	Métricas de Avaliação
Ying (2017)	Divisão dos dados em treino e teste - *Holdout* (75% para treino e 25% para teste)	Algoritmo de classificação: - Árvore de Decisão	- Validação cruzada, com *K-fold*=10 - Acurácia - Matriz de Confusão - Curva ROC - Erro Quadrático Médio Normalizado (*NMSE - Normalized Square Error*)
Rombaut e Guerry (2018)	Pré-processamento: - Criação de uma variável *dummy* para representar dois grupos de anos previamente agrupados Técnica de seleção de variáveis: - Através da aplicação do algoritmo de regressão logística, identificação e remoção das variáveis que tiveram baixo nível de significância (coeficiente P value > 0,05)	Algoritmo de classificação: - Regressão Logística	- Especificidade - Sensibilidade - Curva ROC
Lu (2014)	Pré-processamento: - Valores ausentes: técnica de preenchimento com base no valor da moda de cada grupo - Registros duplicados: remoção dos registros em duplicidade, deixando apenas um dos registros na base de dados - Ruídos: remoção Engenharia de Recursos: - Enriquecimento com 15 novas variáveis preditoras à partir do conjunto de dados original. Técnicas de Seleção de Variáveis: - *F-Score* combinado com o algoritmo SVM associadas a validação cruzada	Algoritmo para construção de grafos: - Para mapeamento das transições de carreira do empregado, onde o armazenamento destes é efetuado em banco de dados orientado à grafos (*Neo4j*) Algoritmos de Classificação: - *Support Vector Machines* - *Decision Table/Naive Bayes*, um classificador híbrido	- Validação cruzada, com *K-fold*=10 - Acurácia - Precisão - Sensibilidade - *F- Score* - Matriz de Confusão

	com *K-Fold*=5 - Qui-Quadrado - Ganho de Informação	- Redes Neurais Artificiais (*Multilayer Perceptron*)	
Januário (2019): presente pesquisa	Pré-processamento: - Registros duplicados: remoção dos registros em duplicidade, deixando apenas um dos registros na base de dados - Ruídos: identificação com *Z-Score*, com a remoção das instâncias Engenharia de Recursos: - Enriquecimento com 2 novas variáveis preditoras a partir do conjunto de dados original, através de técnicas de *clustering* - Codificação de valores categóricos para numéricos através *label enconding* - Padronização de escalas em colunas quantitativas - Técnicas de criação de instâncias para ajuste de classe desbalanceada através de *oversampling* - Para um dos modelos é aplicado redução da dimensionalidade com *PCA* (6 componentes)	Algoritmo de classificação: - *Bagging Classifier* - *Gradient Boosting Classifier* - *Random Forest Classifier* - *Logistic Regression* - *Linear Discriminant Analysis* - *K-Neighbors Classifier* - *Decision Tree Classifier* - *Gaussian NB* - *SVC* Algoritmo de agrupamento: - *Kmeans* (k = 4)	- Validação cruzada, com *K-fold*=15 Algoritmo de classificação: - Matriz de confusão - *Accuracy* - *Precision* - *Recall* - *Specificity* - *F1-Score* - *ROC Curve* Algoritmo de agrupamento: - Método de *Elbow* (detecção da quantidade ideal de clusters) - Coeficiente de Silhueta

Quadro 2. Comparativo de estudos de caso sobre evasão no ambiente corporativo

Considerando-se os estudos do Quadro 2 observa-se similaridade com o presente estudo, no uso da técnica de validação cruzada para avaliar a capacidade de generalização, assim como na utilização das seguintes métricas para avaliação do modelo: matriz de confusão, sensibilidade, acurácia e curva ROC.

Basicamente, se diferem na questão da preparação dos dados, principalmente, na etapa de engenharia de recursos, onde o estudo de Lu (2014) enriqueceu de forma positiva o experimento em relação aos demais, com a abordagem de classificação coletiva. A etapa de *feature engineering* requer criatividade e conhecimento de negócio, possibilitando agregar maior poder preditivo ao modelo.

Todos os autores são unânimes no mesmo ponto de melhoria: ter mais informações a respeito do funcionário. A generalização de um modelo depende de características das instâncias que expliquem bem o evento alvo – a ocorrência ou não da evasão do funcionário. Como complemento, modelos híbridos podem servir como uma variação na construção de modelos preditivos.

4. Experimento

Nessa seção descreve-se como foi desenvolvido o experimento desde a infraestrutura necessária, a aquisição dos dados, etapas de preparação, transformação, engenharia de recursos, pré-processamento, modelagem base e construção do modelo preditivo.

4.1. Infraestrutura

Para suportar o desenvolvimento do experimento foi usada a plataforma *Jupyter Notebook* do *Google Colab* (Google, 2019), onde executou-se sobre um container com cerca de 16 Gb de memória RAM, em ambiente de computação em nuvem. Foi usada a linguagem de programação *Python*, assim como bibliotecas comumente usadas nas etapas de um projeto de dados, tais como: *pandas*, *numpy*, *scipy*, *seaborn*, *matplotlib*, *imblearn*, *sklearn*, *lime*.

A base de dados selecionada para avaliação foi extraída do portal *Kaggle*, de um desafio denominado *Employee Churn Prediction*, acessível pelo link https://www.kaggle.com/c/employee-churn-prediction/data. Os dados estão no formato tabular, em um arquivo no formato CSV (valores separados por vírgula). A opção por este conjunto de dados se deve à dificuldade encontrada de conseguir empresas aptas a fornecerem dados sensíveis e de qualidade sobre seus colaboradores. Logo, ficou concebido que o trabalho seria orientado a conjuntos de dados públicos.

4.2. Descrição do Conjunto de Dados

Trata-se de um conjunto de dados de recursos humanos contendo 14.999 instâncias, onde cada linha é a observação individual de um funcionário. Abaixo, o Quadro 3 descreve as características contidas na base de dados.

Nome do Atributo	Descrição	Tipo de Dados
satisfaction_level	Valor do nível de satisfação atual	float64
last_evaluation	Valor da última avaliação de performance	float64
number_project	Qtd. de projetos de atuação	int64
average_montly_hours	Média da qtd. horas trabalhadas no mês	int64
time_spend_company	Total da qtd. de horas de deslocamento p/ o Trabalho	int64
work_accident	Se ocorreram acidentes de trabalho (0 = Não, 1 = Sim)	bool
left	CLASSE: situação do empregado (0 = Ativo, 1 = Evadiu/Desligado)	bool
promotion_last_5years	Promoção nos últimos 5 Anos (0 = Não, 1 = Sim)	bool
department_name	Nome do setor de atuação	object
salary_category_name	Descrição da categoria do salário	object

Quadro 3. Variáveis dispostas no conjunto de dados

4.3. Preparação dos Dados

Nessa etapa, o conjunto de dados é avaliado quanto aos valores fora da curva, valores ausentes e registros duplicados. Após a detecção de tais valores, técnicas são aplicadas com o intuito de efetuar uma limpeza e qualificação no conjunto de dados.

4.3.1. Ajuste de Valores Fora da Curva (*Outliers*)

Na Figura 8 são apresentados gráficos de histograma, exibindo a distribuição dos valores em cada uma das variáveis quantitativas contidas no conjunto de dados do experimento. Há um total de 376 valores fora da curva, sendo que a variável *time_spend_company* evidencia melhor o problema, pois há observações com valores distantes da aglomeração padrão.

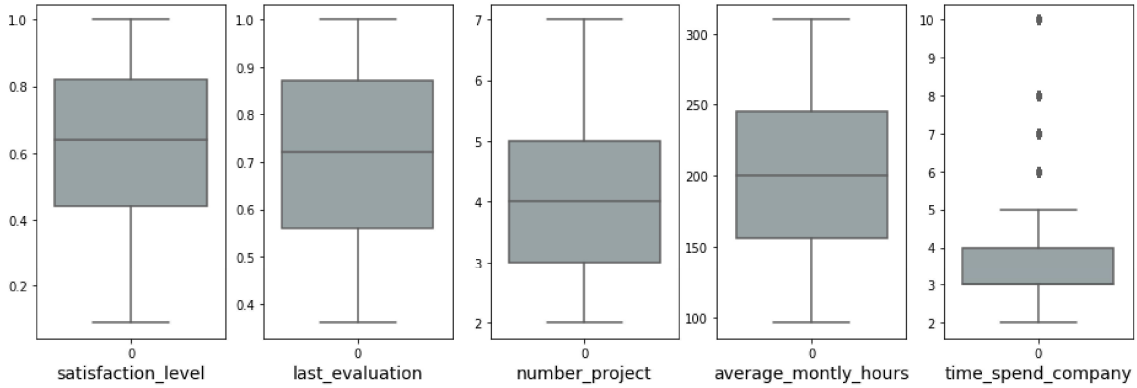

Figura 8. Distribuição dos dados do experimento

Como forma de detectar os valores fora da curva, é aplicada a técnica do *Z-Score*. Para os registros que apresentam ruídos em sua distribuição, é efetuada a remoção das instâncias com as incidências encontradas.

Nenhum dos atributos apresenta valores ausentes. Foram detectados 2.820 registros redundantes, sendo mantido no conjunto de dados apenas a primeira instância de cada uma das observações redundantes, junto com as demais não redundantes.

4.4. Engenharia de Recursos

Como forma de enriquecer o conjunto de dados foi efetuada a criação de novas variáveis com base nas já existentes. O processo de engenharia de recursos demanda criatividade e domínio do negócio [Burkov, 2019].

4.4.1. Criação de Variáveis com Técnicas de Agrupamento de Dados

Através do algoritmo de *clustering Kmeans*, possibilitou-se a criação de dois novos atributos: *EmployeeGroup* e *EmployeeSimilarity*. O algoritmo *Kmeans* requer um número de *clusters* para gerar o agrupamento e identificar o grupo de cada instância do conjunto de dados. Para tanto, é usado o método de *Elbow*, o qual visa justamente entender qual o valor ideal de *clusters* para o conjunto de dados em questão.

A Figura 9 demonstra que k=4 é o melhor valor, uma vez que quando K>=4 não há variação significativa na homogeneidade das instâncias em relação aos seus respectivos centroides.

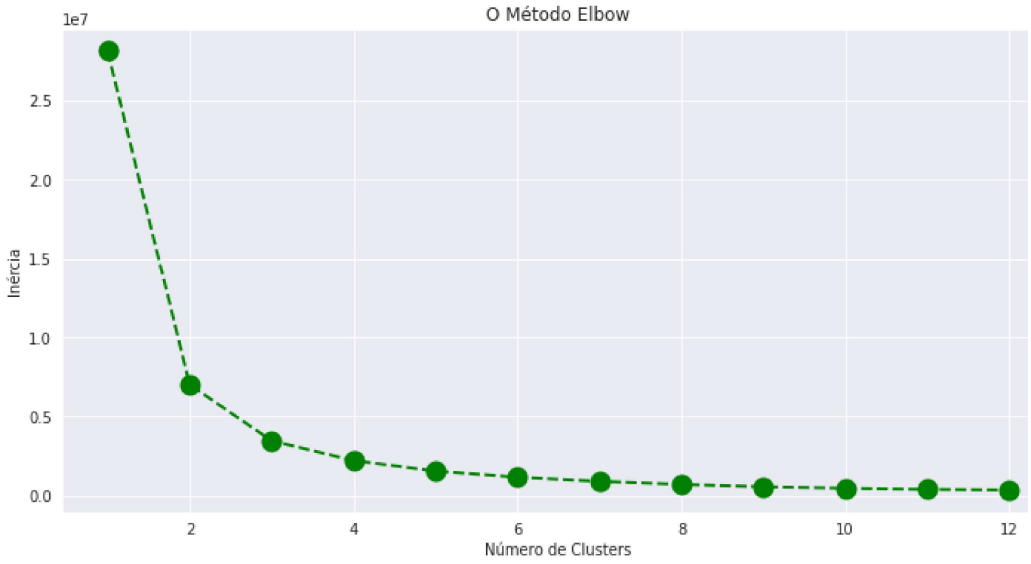

Figura 9. Variação da homogeneidade das instâncias em relação aos respectivos centroides

Aplicando o algoritmo de agrupamento com 4 clusters, o valor do coeficiente da silhueta é de 0,52, representando o quão semelhante está um objeto ao seu próprio *cluster* comparado com os demais *clusters*.

Efetuado o agrupamento, possibilitou-se criar a primeira variável: *EmployeeGroup*, que se trata do rótulo - número do grupo, que o algoritmo criou para cada uma das instâncias. Para criar a segunda variável, *EmployeeSimilarity*, foi necessário encontrar a similaridade (grau de distância em que está a instância em relação ao seu próprio cluster) dos registros com relação aos grupos atribuídos. Para cada instância, o algoritmo gera uma distância em relação a cada centroide, sendo que a menor delas indica o seu cluster e a distância até este. A Figura 10 exibe os 5 primeiros registros contendo apenas as colunas *EmployeeGroup* e *EmployeeSimilarity*.

	employee_group	employee_similarity
0	0.0	15.600738
1	3.0	3.726462
2	3.0	8.337005
3	1.0	3.003202
4	0.0	17.587536

Figura 10. Observações do conjunto de dados do experimento, após novas colunas

4.4.2. Codificando Valores Categóricos

No conjunto de dados, há duas variáveis categóricas - *department_name* e *salary_category_name*, que passam por codificação de valores categóricos para numéricos através da técnica conhecida como *label enconding*. Basicamente, os valores das categorias são associados a números. A Figura 11 exibe os 5 primeiros registros contendo apenas as

colunas originais - *department_name* e *salary_category_name*, e as novas criadas a partir da técnica de *label enconding* - *department_name_ID* e *salary_category_name_ID*.

department_name	salary_category_name	department_name_ID	salary_category_name_ID
sales	low	7	1
sales	medium	7	2
sales	medium	7	2
sales	low	7	1
sales	low	7	1

Figura 11. Observações do conjunto de dados do experimento, após *label enconding*

4.4.3. Padronização

É exatamente o mesmo cálculo do *Z-Score*, porém dessa vez é aplicado em todas as colunas quantitativas e não apenas para as que continham *outliers*. A padronização tem as propriedades de um padrão normalmente distribuído (distribuição Gaussiana), com a média = 0 e o desvio padrão = 1 [Burkov 2019].

4.5. Balanceamento das Instâncias em Relação aos Valores Possíveis da Classe

Avaliando a distribuição de valores da classe *left* (situação da evasão), percebe-se que a variável alvo está desbalanceada, conforme mostra a Quadro 4.

Classe	Proporção
Evadiu	83,13%
Não Evadiu	16,87%

Quadro 4. Balanceamento original de instâncias agrupado por valores da classe

Segundo Facelli, Lorena, Gama e Carvalho (2017), trata-se de um tópico da área de classificação de dados, onde conjuntos de dados com classes desbalanceadas podem prejudicar o desempenho do classificador, ao qual o modelo tende a favorecer a classificação de novos dados na classe majoritária. Para considerar aceitável um cenário destes - classes desbalanceadas, a acurácia preditiva do classificador para a classe minoritária deve ser maior que a acurácia obtida atribuindo todo novo objeto à classe majoritária. Caso contrário, técnicas de balanceamento de classes deverão ser implementadas. A seguir são apresentadas algumas opções para solucionar este problema:

- Balanceamento natural: geração de novos dados pelo mesmo processo que gerou o conjunto atual - essa é a melhor prática indicada, mas nem sempre possível;
- Balanceamento sintético com *oversampling*: incremento de instâncias à classe minoritária. Existe o risco das novas instâncias adicionadas representarem situações que nunca ocorrerão, o que pode induzir um modelo inadequado de dados, além de problemas de *overfitting* - modelo supera justado aos dados de treinamento, porém não consegue generalizar bem novos dados de entrada. Ou seja, o modelo "decorou" os dados de treinamento;

- Balanceamento sintético com *undersampling*: dados da classe majoritária são eliminados do conjunto de dados. Existe o risco de dados de grande importância serem perdido, o que pode afetar diretamente a indução do modelo correto, além do problema de *underfitting* - modelo com baixa performance junto aos dados de treinamento.

Para o conjunto de dados do experimento foi aplicada a técnica de *oversampling*, a fim de obter novos exemplos sintéticos para a classe minoritária. Para tanto, é aplicado o algoritmo *SMOTE - Synthetic Minority Oversampling Technique*. Após a criação dos exemplos, a classes fica balanceada conforme mostra o Quadro 5.

Classe	Proporção
Evadiu	50%
Não Evadiu	50%

Quadro 5. Balanceamento de instâncias pós processo de *oversampling*

4.6. Modelagem Base

Nesta subseção, após todos os tratamentos de dados efetuados foi aplicada a modelagem base, no intuito de definir o algoritmo que melhor se ajusta ao conjunto de dados e ao problema.

4.6.1. Divisão dos Dados - Treino e Teste

A técnica escolhida para divisão dos dados é a *holdout*, onde basicamente é definida a proporção de instâncias do conjunto de dados original que irão representar os dados de treino e de teste [Dutta e Barai 2019]. No caso do experimento, ficou definido da seguinte forma: 70% para o conjunto de dados de treino e 30% para o conjunto de dados de teste.

4.7. Construção dos Modelos

Para o experimento, é aplicado *cross validation* sobre o conjunto de dados de treino - os 70% do *holdout*, com K=15. A validação cruzada é aplicada em um algoritmo por vez. Os algoritmos que recebem os dados do processo de *cross validation* são relacionados a seguir: *Bagging Classifier, Gradient Boosting Classifier, Random Forest Classifier, Logistic Regression, Linear Discriminant Analysis, K-Neighbors Classifier, Decision Tree Classifier, Gaussian NB e SVC.*

Ou seja, para cada um dos 9 algoritmos avaliados, 15 partições são treinadas e testadas, e a média destas resulta na acurácia do modelo.

4.7.1. Sem Redução da Dimensionalidade

Nessa subseção é descrito o processo de treinamento dos modelos de classificação, sem a redução da dimensionalidade usando o conjunto de dados de treino.

4.7.1.1. Parâmetros

O Quadro 6 informa os parâmetros usados para cada um dos algoritmos de classificação na validação cruzada. Todos os algoritmos usados são implementados pela biblioteca *scikit-learn*, disponível em: https://scikit-learn.org/stable/.

Algoritmo	Parâmetros
Bagging Classifier	base_estimator=None, bootstrap=True, bootstrap_features=False, max_features=1.0, max_samples=1.0, n_estimators=10, n_jobs=None, oob_score=False, random_state=None, verbose=0, warm_start=False
Gradient Boosting Classifier	criterion='friedman_mse', init=None, learning_rate=0.1, loss='deviance', max_depth=3, max_features=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, n_estimators=100, n_iter_no_change=None, presort='auto', random_state=None, subsample=1.0, tol=0.0001, validation_fraction=0.1, verbose=0, warm_start=False
Random Forest Classifier	bootstrap=True, class_weight=None, criterion='gini', max_depth=None, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, n_estimators='warn', n_jobs=None, oob_score=False, random_state=None, verbose=0, warm_start=False
Logistic Regression	C=1.0, class_weight=None, dual=False, fit_intercept=True, intercept_scaling=1, l1_ratio=None, max_iter=100, multi_class='warn', n_jobs=None, penalty='l2', random_state=None, solver='warn', tol=0.0001, verbose=0, warm_start=False
Linear Discriminant Analysis	n_components=None, priors=None, shrinkage=None, solver='svd', store_covariance=False, tol=0.0001
K-Neighbors Classifier	algorithm='auto', leaf_size=30, metric='minkowski', metric_params=None, n_jobs=None, n_neighbors=5, p=2, weights='uniform'
Decision Tree Classifier.	class_weight=None, criterion='gini', max_depth=None, max_features=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, presort=False, random_state=None, splitter='best'
Gaussian NB.	priors=None, var_smoothing=1e-09
SVC.	C=1.0, cache_size=200, class_weight=None, coef0=0.0, decision_function_shape='ovr', degree=3, gamma='auto_deprecated', kernel='rbf', max_iter=-1, probability=False, random_state=None, shrinking=True, tol=0.001, verbose=False

Quadro 6. Parâmetros aplicados nos algoritmos de classificação

4.7.1.2. Processo de Validação Cruzada

Para o experimento, o conjunto de dados de treino é divido em 15 subconjuntos de dados para a construção do modelo usando a técnica de *cross validation*. A Figura 12 exibe a métrica da acurácia dos modelos em gráfico de *boxplot*, onde é possível avaliar a variação dos resultados de cada K-subconjunto de dados que os algoritmos obtiveram durante a validação cruzada. Pode-se concluir que *Bagging Classifier*, *Gradient Boosting Classifier* e *Random Forest Classifier* tem as menores variações, logo os resultados ficam concentrados próximos da média, enquanto que o algoritmo *Gaussian NB* tem uma alta variância da acurácia para a geração do seu modelo.

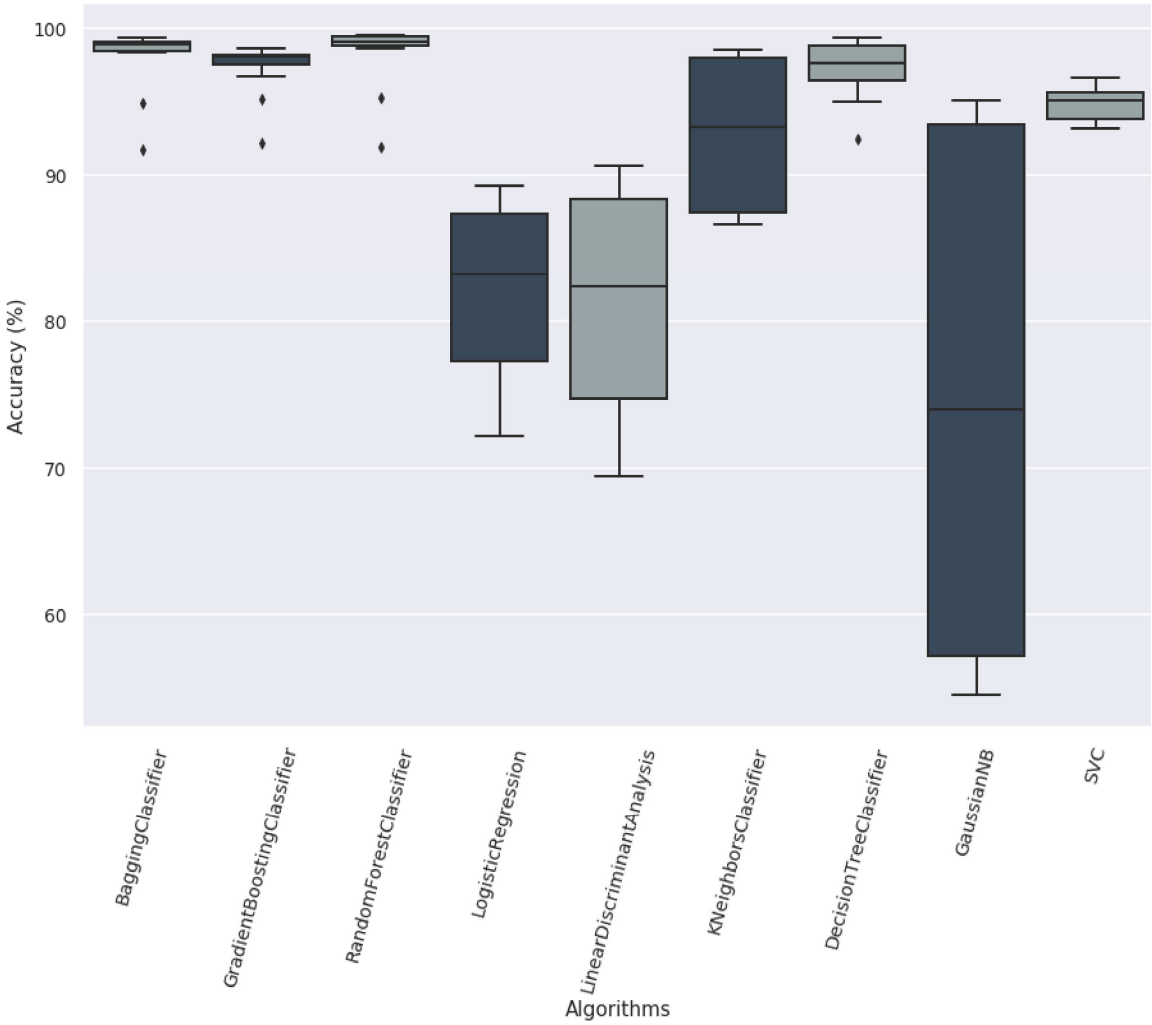

Figura 12. Acurácia dos modelos de classificação com validação cruzada

O Quadro 7 demonstra a acurácia, o desvio padrão e o tempo de aprendizado levado por cada um dos algoritmos selecionados.

Algoritmos	**Acurácia**	**Desvio Padrão**	**Tempo de Execução (HH:MM:SS)**
Bagging Classifier	98.32%	1.85%	00:00:23
Gradient Boosting Classifier	97.42%	1.77%	00:00:30
Random Forest Classifier	98.34%	2.14%	00:00:18
Logistic Regression	82.01%	5.10%	00:00:18
Linear Discriminant Analysis	81.29%	6.66%	00:00:17
K-Neighbors Classifier	92.86%	4.97%	00:00:17
Decision Tree Classifier	97.49%	1.87%	00:00:17
Gaussian NB	72.61%	21.85%	00:00:17
SVC	94.87%	1.17%	00:01:30

Quadro 7. Acurácia, desvio padrão e tempo de execução dos algoritmos de classificação

O tempo de treinamento é um importante balizador em relação ao algoritmo a ser escolhido para a resolução do problema. Considerando os critérios do Quadro 7, o algoritmo que tem a melhor acurácia com baixo tempo de treinamento é o *Random Forest Classifier*.

Já *SVC – Support Vector Classifier*, tem um custo maior de tempo para o treinamento do modelo, apesar de entregar uma boa acurácia com baixo desvio padrão.

4.7.1.3. *Random Forest Classifier:* o algoritmo selecionado

Considerando que *Random Forest Classifier* teve a melhor acurácia no processo de validação cruzada, ele é o selecionado no prosseguimento das etapas seguintes. Este modelo passa por novo treinamento, agora com método *holdout* para seleção de dados de treino e teste. A estratégia 70/30 é mantida (70% dados para treino e 30% dados de teste).

4.7.2. Com Redução da Dimensionalidade

Nessa subseção é descrito o processo de treinamento dos modelos de classificação, considerando a redução para 2 dimensionalidades do conjunto de dados de treino. Para tanto, é aplicada a técnica de *PCA* (*Principal Component Analysis*). O algoritmo escolhido para gerar o modelo de classificação foi o *Random Forest Classifier*.

4.7.2.1. Parâmetros

O Quadro 8 informa os parâmetros usados para o algoritmo de classificação selecionado. O método de seleção de dados de treino e teste foi o *holdout*.

Algoritmo	Parâmetros
Random Forest Classifier	bootstrap=True, class_weight=None, criterion='gini', max_depth=None, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, n_estimators='warn', n_jobs=None, oob_score=False, random_state=None, verbose=0, warm_start=False

Quadro 8. Parâmetros utilizados no algoritmo *Random Forest Classifier*

5. Resultados

Nessa seção são apresentados e analisados os resultados do experimento feito nesse estudo.

5.1. Métricas do Modelo - Sem Redução da Dimensionalidade

Nesta subseção são informados os valores das principais métricas do modelo de classificação, gerado com *Random Forest Classifier*.

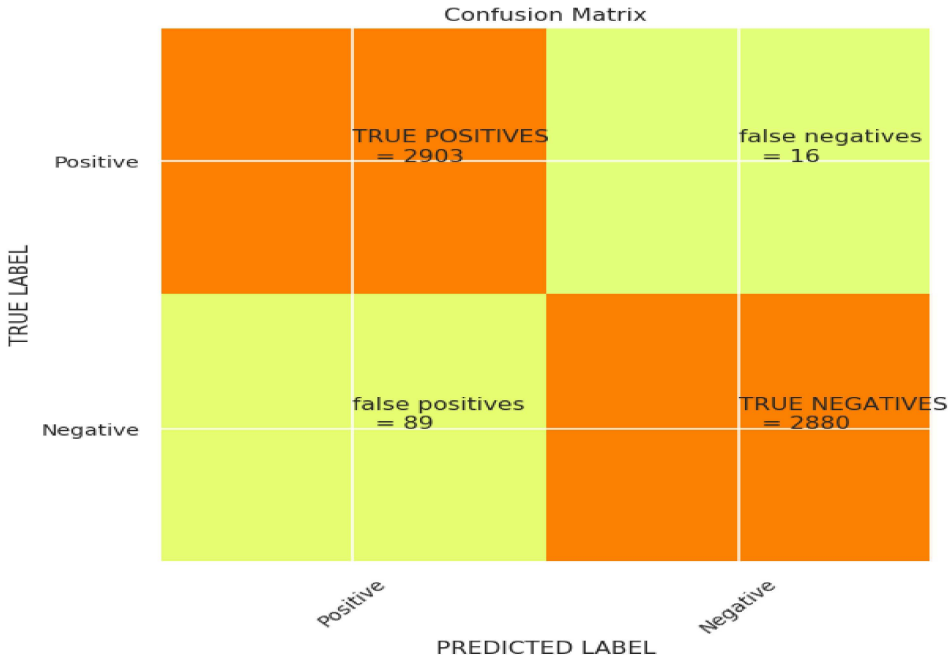

Figura 13. Matriz de confusão do modelo Gerado

Conforme Zeng (2019), a matriz de confusão possibilita medir a precisão, as taxas de erro e acerto do modelo gerado. A Figura 13 exibe a matriz de confusão gerada pelo modelo do experimento, onde a partir desta será extraída as demais métricas do classificador.

O Quadro 9 exibe os resultados gerados das métricas usadas para avaliação.

Model	Accuracy	Precision	Recall	Specificity	F1-Score
Random Forest Classifier	98,21%	97,02%	99,45%	97,00%	98,22%

Quadro 9. Métricas avaliadas do algoritmo *Random Forest Classifier*

Sabendo que a acurácia é a taxa de instâncias classificadas corretamente em relação ao total de instâncias preditas [Yousaf 2016], onde o valor de 98,21% indica que o classificador acertou a classe da maioria das instâncias avaliadas pelo modelo.

Precisão é a proporção de funcionários que o modelo classificou para o evento da evasão ocorrer em relação ao domínio de funcionários que realmente evadiram [Burkov, 2019]. A sensibilidade – *recall*, é a proporção de funcionários corretamente classificados para o evento da evasão ocorrer em relação ao domínio de funcionários preditos com o evento da evasão ocorrer [Alao, Adeyemo 2013]. Especificidade trata da proporção de funcionários corretamente classificados para o evento da evasão não ocorrer em relação ao domínio de funcionários preditos com o evento da evasão não ocorrer [Yousaf 2016]. Já, a *F1-Score* considera o balanço das taxas preditiva positiva (precisão) e verdadeiros positivos (sensibilidade) [Lu. 2014]. Considerando as métricas mencionadas – precisão, sensibilidade, especificidade e *F1-Score*, e seus respectivos valores para o modelo do experimento – 97,02%, 99,45%, 97%, 98,22%, é possível observar a boa capacidade de generalização do classificador.

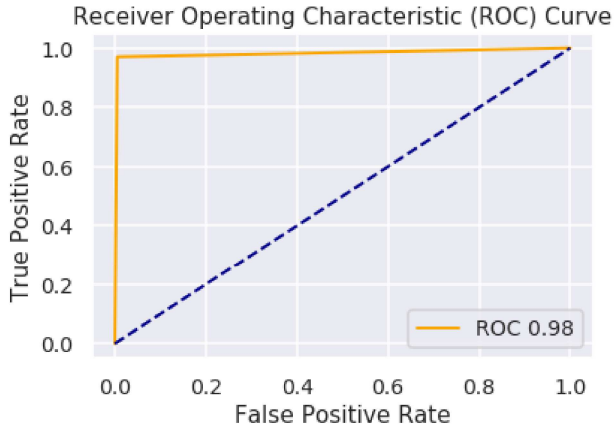

Figura 14. Curva ROC do modelo

Curva ROC tem como objetivo mensurar a probabilidade do modelo classificar um exemplo positivo aleatório mais do que um exemplo negative aleatório. [Yedida, Reddy, Vahi, Jana, Gv e Kulkarni 2018]. A Figura 14 indica que o modelo tem uma maior probabilidade de classificar uma instância dentro da sua classe esperada, independente do evento – a evasão do funcionário ocorrer ou não. O valor da AUC - area under the ROC curve, varia de 0,0 até 1,0, onde a linha pontilhada azul representa o valor 0,5. O valor do AUC do experimento gerado ficou em 0,98, representando maior taxa preditiva positiva do modelo.

5.1.2. Gerando o Score da Probabilidade do Evento Ocorrer

Sabendo que *Random Forest Classifier* é um modelo *ensemble* que cria *N* árvores de decisão formando uma floresta, e que cada árvore tem a sua própria decisão e escolhe uma classe, o *score* da probabilidade de cada evento ocorrer é o número de eventos para cada classe entre toda a floresta, dividido pelo número de árvores na floresta. Para tanto, no algoritmo em questão implementado na biblioteca *scikit-learn*, há um método chamado *predict_proba* que calcula o *score* da probabilidade de cada evento ocorrer, dentro das classes disponíveis.

Para o modelo treinado neste experimento, uma amostra das probabilidades de cada evento ocorrer pode ser visualizada na Figura 15 abaixo.

alvo_teste	alvo_previsoes	probabilidade_EVASÃO_NÃO_OCORRER	probabilidade_EVASÃO_OCORRER
1.0	1.0	0.0	1.0
1.0	1.0	0.0	1.0
1.0	1.0	0.0	1.0
1.0	1.0	0.0	1.0
0.0	0.0	1.0	0.0
1.0	1.0	0.0	1.0
0.0	0.0	1.0	0.0
1.0	1.0	0.0	1.0
1.0	1.0	0.0	1.0
0.0	0.0	0.9	0.1

Figura 15. Score da probabilidade dos eventos

A coluna “alvo_teste” representa a classe real do conjunto de dados de teste, enquanto, que “alvo_previsoes” foi o valor predito pelo modelo. Já, as colunas “probabilidade_EVASÃO_NÃO_OCORRER” e “probabilidade_EVASÃO_OCORRER”, indicam a força de cada evento ocorrer em uma escala de 0 até 1.

5.1.3. Árvore de Decisão Minimizada do Modelo com *Random Forest*

Os modelos baseados em árvores de decisão têm entre suas vantagens, a de fornecer uma fácil interpretação do modelo gerado. Na Figura 16 é demonstrada a árvore gerada com apenas 3 nodos, ou seja, há uma poda forçada para possibilitar exibir a imagem, pois a árvore original do modelo tem muitas ramificações.

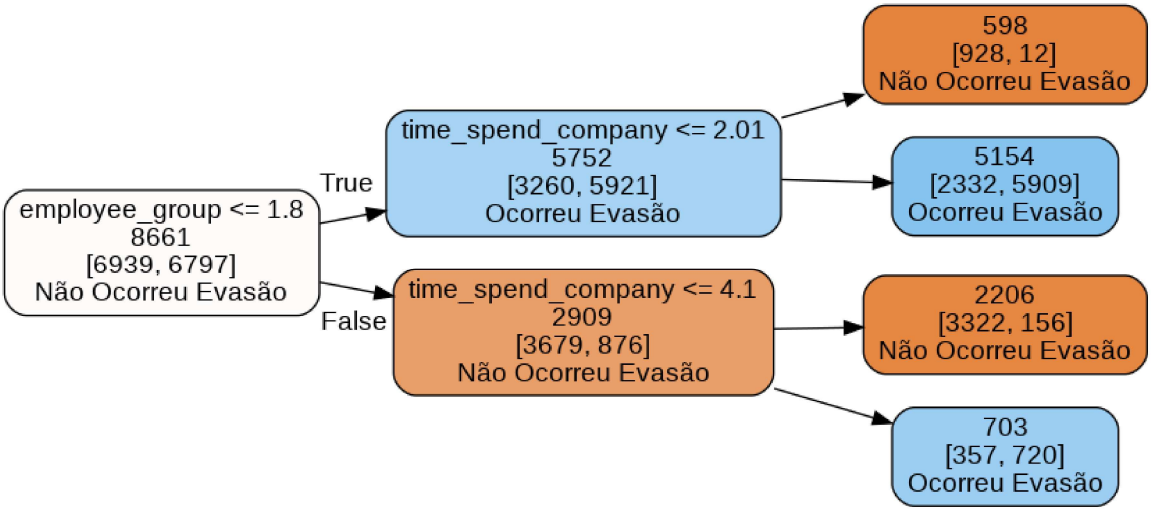

Figura 16. Modelo *Random Forest Classifier* com 3 nodos

Apesar da árvore do modelo original da Figura 16 estar limitada a 3 nodos, ela sintetiza o trabalho do classificador da seguinte forma: o nodo raiz, contido no primeiro retângulo de cor branca à esquerda, representa a amostra inteira de dados, com cerca de 8661

instâncias. Demais nodos com cores azul e laranja, são considerados nodos folhas. Há duas possibilidades para o funcionário ter a sua evasão detectada pelo classificador:

- se o grupo ao qual ele participa for menor ou igual a 1,8 e o tempo que ele leva em deslocamento para chegar na empresa for menor ou igual a 2,01 horas. Do total das observações, 5154 fazem parte deste grupo de evasões.
- se o grupo ao qual ele participa for maior que 1,8 e o tempo que ele leva em deslocamento para chegar na empresa for maior que 2,01 horas e menor ou igual a 4,1 horas. Do total das observações, 703 fazem parte deste grupo de evasões.

Qualquer outra condição diferente das citadas anteriormente levam ao classificador indicar que o funcionário não irá evadir, representado cerca de 2804 das observações. Quanto maior o ganho de informação obtido por uma das variáveis preditoras, mais próximo ela estará da raiz da árvore de decisão.

5.2. Importância das Variáveis Preditoras - Sem Redução da Dimensionalidade

Nessa subseção é descrito o processo de identificação das variáveis preditoras e seus respectivos graus de importância.

5.2.1. Seleção de Características do Modelo Gerado

Uma árvore de decisão contém *N* nodos que são testados, durante fase de treinamento, sendo possível calcular o grau de impureza (entropia) ponderada contida nesta, computando o ganho de informação em cada um dos atributos. Os nós que apresentam o menor valor de entropia, são os que tem a maior importância para o modelo gerado. Se tratando de uma floresta de árvores de decisão, é considerada a média do grau de impureza das árvores consideradas na construção do modelo para computar e medida de ganho de informação dos atributos.

No algoritmo *Random Forest Classifier*, na biblioteca *scikit-learn*, há um método chamado *feature_importances_*, que computa o ganho de informação considerando o grau de impureza dos atributos no modelo gerado. Para o modelo treinado neste experimento, a importância das variáveis pode ser visualizada na Figura 17.

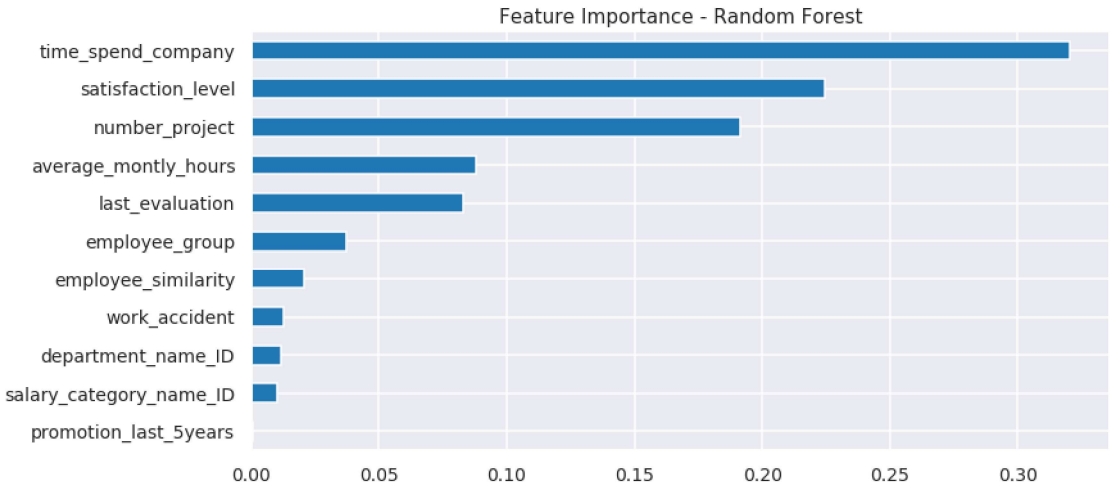

Figura 17. Importância das variáveis do modelo.

A variável que tem o maior ganho de informação a respeito do evento alvo – evasão do funcionário, é a *time_spend_company*, que representa a quantidade de horas que o funcionário leva em deslocamento para a empresa. Considerando que na base de dados do experimento há pessoas que levam até 4 horas em deslocamento, há de se suspeitar que tais funcionários possam estar insatisfeitos com essa condição, podendo pesar em uma eventual evasão do emprego.

Demais variáveis que impactam significativamente no evento alvo são *satisfaction_level* e *number_project*. A primeira diz respeito ao nível de satisfação do funcionário em relação ao emprego e a segunda sobre a quantidade de projetos em atuação. Funcionários com baixo nível de satisfação tem maior probabilidade de evadir o emprego? E se a carga de projetos for acima da média da amostragem, há uma forte tendência de evasão? Essas são questões que podem ser respondidas, a partir de uma análise exploratória combinada com teste de hipóteses, porém, não serão abordadas neste estudo por não fazerem parte do escopo da pesquisa proposta.

5.2.2. Características e Respectivos Pesos para a Previsão do Evento de Uma Instância

Nesta subseção é descrito o processo de identificação das características para a predição de uma determinada instância do conjunto de dados e seus respectivos graus de importância.

5.2.2.1. LIME

Local Interpretable Model-Agnostic Explanations – LIME, indica quais características e os seus respectivos graus de importância que contribuem para a predição na classificação do evento da instância avaliada, visando a interpretação de modelos de aprendizado de máquina *black box* – modelos que não permitem fácil explicação de suas previsões [Ribeiro, Singh e Guestrin 2016].

Conforme L. Hulstaert (2018), para gerar uma explicação de um modelo *black box*, *LIME* treina modelos interpretáveis - lineares com forte regularização, como por exemplo árvores de decisão, com um conjunto de dados gerado através de pequenas variações nas características fornecidas, a partir da instância original, que está em avaliação. Tal modelo fornece apenas uma boa aproximação local, uma vez que é avaliado pequenas variações de uma única instância.

Com isso, aproximando a *black box* localmente da vizinhança da amostra de dados, é simplificada a tarefa da explicação do evento de uma única instância. Apesar disso, existem armadilhas onde aplicar *LIME* pode se tornar uma desvantagem. Sabendo que apenas modelos lineares são usados para aproximar o comportamento local, ao expandirmos muito a região amostral é possível que haja uma perda no poder de explicação em relação ao comportamento do modelo original, onde as amostras são menores e mais lineares em torno da instância avaliada. Isso se dá devido a não linearidade das regiões locais do conjunto de dados, exigindo modelos complexos e não interpretáveis.

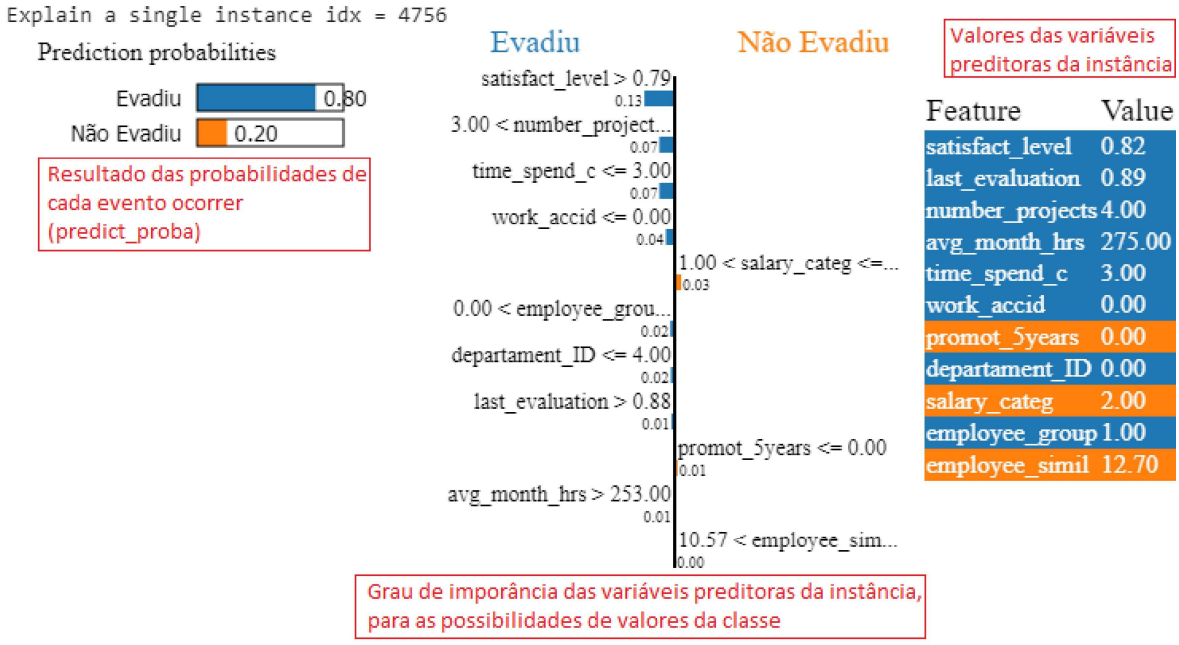

Figura 18. Importância das variáveis para uma instância do modelo.

A Figura 18 representa a saída da predição do funcionário com índice número 4756, onde no canto direito são apresentados os valores de entrada para cada uma das variáveis preditoras da instância. No canto superior esquerdo é apresentado o resultado das probabilidades da predição (*predict_proba*) – 80% de chances de ocorrer a evasão e 20% de chances de não ocorrer a evasão.

No meio dessa figura é apresentado o grau de importância de cada uma das variáveis preditoras de entrada em relação as possibilidades de valores da classe. Nota-se que as variáveis que mais impactaram para a probabilidade da evasão ocorrer para a instância avaliada são *satisfaction_level*, *number_project* e *time_spend_company*.

5.3. Teste T para Validação da Hipótese

Nesta subseção é descrito o resultado do teste de hipótese levantado nos objetivos específicos deste artigo.

5.3.1. O modelo gerado sem redução da dimensionalidade é mais eficiente que o modelo com redução da dimensionalidade?

No Quadro 10 são apresentadas as métricas e seus respectivos valores utilizados para a avaliação dos modelos gerados, considerando os dois cenários propostos: sem e com redução da dimensionalidade, onde no modelo com PCA (*Principal Component Analysis*) são considerados 6 componentes principais, que explicam um total de 72% da variância dos dados em relação ao conjunto de dados original.

Métricas	Modelo sem redução da dimensionalidade	Modelo com redução da dimensionalidade (6 componentes principais)
Acurácia	0,95482337	0,953634511

Precisão	0,94438861	0,941
Sensibilidade	0,965741692	0,967112025
Especificidade	0,944088919	0,940383968
F1 Score	0,954945799	0,953877344
Média da taxa de falsos positivos (ROC)	0,344752769	0,344295992
Média da taxa de verdadeiros positivos (ROC)	0,64802964	0,646794656
Média dos valores	0,822395828	0,821014071
Desvio Padrão	0,239406	0,239041
Quantidade de amostras	7	7

Quadro 10. Métricas e medidas sumarizadas dos modelos de classificação - sem e com redução da dimensionalidade

Tendo conhecimento das métricas e parâmetros apresentados, abaixo é apresentada as etapas para responder a hipótese levantada desta subseção:

1º) Formulação das hipóteses para o teste unilateral à esquerda:

- H0 (hipótese nula): a média dos valores das métricas do modelo com redução ser maior ou igual a 0,822395828.
- H1 (hipótese alternativa): a média dos valores das métricas do modelo com redução ser menor que 0,822395828.

2º) Nível de significância do teste: 0,05.

3º) Amostra aleatória, onde:

- n (quantidade de amostras) = 7
- z (a média dos valores das métricas do modelo com redução) = 0,821014071
- s (desvio padrão) = 0,239041

4º) Cálculo da estatística amostral do teste t: tcal = $(0{,}821014071 - 0{,}822395828) \,/\, \left(0{,}239041 - \sqrt{7}\right) = 0{,}000574$

5º) Determinando a região de rejeição da hipótese nula do teste: pela tabela *t student*, temos tcrit = 1,943. Como a região de rejeição de H0 ocorre para valores de |tcal| >= |tcrit|, |tcal| calculado não é >= 1,943, considerando as 7 amostras e o nível de significância de 5%.

6º) Conforme o item anterior, o valor da estatística do teste calculada é menor que o valor crítico, logo aceita-se H0 pois não há evidências para afirmar H1.

5.4. Métricas do Modelo Gerado X Competição do Kaggle

Essa subseção trata de comparar a acurácia do modelo do experimento gerado neste artigo com os primeiros colocados para o mesmo desafio no portal *Kaggle*, exibidos na Figura 19. Ambos trabalham no mesmo conjunto de dados e tratam da resolução do seguinte problema: um modelo preditivo que vise identificar se o empregado irá ou não evadir o emprego. A acurácia do experimento deste artigo, sem redução da dimensionalidade ficou em torno de 98,21%. Atualmente, o modelo do experimento desenvolvido estaria entre os *top* 10 do desafio proposto pelo portal *Kaggle*. Entretanto, se optou por não compartilhar publicamente tal experimento, nesse momento.

#	△pub	Team Name	Kernel	Team Members	Score
1	▲4	WillRyanandJackLatham			0.98760
2	▼1	Allison & Madison			0.98560
3	▲4	Arvan_Smith_Teggi			0.98520
4	▼2	Kali Riggins			0.98520
5	▼2	scottskowronskiandverapalad...			0.98520
6	▼2	ErikRomerandGabrielleHusted			0.98520
7	▼1	AguiarCoulthard2			0.98520
8	▲1	Matt Rubin			0.98200
9	▲1	Joel & Alison			0.98040
10	▼2	Jordan Whitaker			0.98000

Figura 19. Top 10 desafio Kaggle: *Employee Churn Prediction*

6. Conclusão

O estudo visa responder a seguinte questão de pesquisa: como prever e identificar as principais causas da evasão dos empregados de uma corporação, por meio de aplicação de técnicas de aprendizado de máquina? O objetivo principal é prever a evasão dos funcionários, compreendendo os motivos de tal evento. Para tanto, possui como objetivos específicos: (i) Enriquecimento do conjunto de dados com a criação de novas informações; (ii) Gerar um score com a probabilidade tanto da evasão ocorrer quanto de não ocorrer, para cada funcionário avaliado; (iii) Determinar, através de inferência estatística, se o modelo gerado originalmente (sem redução dimensionalidade) é tão eficiente quanto um modelo gerado com redução da dimensionalidade, através de *Principal Component Analysis* (PCA).

Seguindo uma metodologia orientada por diretrizes direcionada para projetos de mineração de dados - *CRISP-DM*, associada às técnicas de ciência de dados, possibilitou-se encontrar tais respostas. A predição da evasão de empregados pôde ser obtida aplicando algoritmos de classificação, onde o desempenho do algoritmo selecionado pós-validação cruzada - *Random Forest Classifier*, foi satisfatória. No cenário em que não foi aplicada redução da dimensionalidade, o modelo obteve uma acurácia de 98,21%, representando que a cada 1000 funcionários avaliados pelo modelo, 980 são classificados corretamente e apenas 20 incorretamente. A taxa de precisão de 97,02% indica uma proporção relevante de instâncias classificadas como positivas em relação ao domínio de instâncias, que realmente são positivas, assim como a sensibilidade de 99,45% revela a proporção massiva de instâncias corretamente classificadas como positivas em relação ao domínio de instâncias preditas como positivas.

Já, a especificidade em torno 97% indica a boa taxa da eficiência do modelo em reconhecer amostras negativas. Não menos importante, a métrica *F1-Score* obteve uma taxa de 98,22%, indicando um ótimo balanço das taxas preditiva positiva (precisão) e verdadeiros positivos (sensibilidade). As métricas apresentadas e seus respectivos valores atestam que o

modelo tem boa capacidade de generalização para o conjunto de dados trabalhado, gerando resposta satisfatória para a predição do evento alvo - evasão do funcionário. Logo, através de algoritmos de aprendizado de máquina voltados para problemas de classificação, é possível prever e identificar as principais causas da evasão dos funcionários de uma corporação.

A previsão da evasão de um funcionário e a explicação deste evento indicando quais características e os seus respectivos graus de importância que contribuíram para a classificação, assim como a apresentação das probabilidades de chances do evento ocorrer em cada possibilidade de classe, são os principais diferenciais em relação à pesquisa aplicada sobre o tema. Tendo em vista que o enriquecimento do conjunto de dados é importante para o aprendizado de máquina, o experimento teve dois incrementos de características: *EmployeeGroup* - grupo ao qual pertence determinada instância, e *EmployeeSimilarity* - grau de distância em que está a instância em relação ao seu próprio *cluster*. Tais informações foram obtidas após aplicação do algoritmo *Kmeans*, que visa agrupar instâncias com características semelhantes. A aplicabilidade do experimento seguindo as definições de um *framework* - *CRISP-DM*, é uma boa prática de condução para projetos de dados.

A proposta inicial deste projeto de pesquisa era aplicar os conhecimentos abordados em grandes conjuntos de dados de corporações, que mantêm um fluxo ativo de empregados. Porém, devido às dificuldades de conseguir empresas aptas a fornecerem dados sensíveis sobre seus colaboradores, assim como na busca de dados de qualidade na área de recursos humano, ficou concebido que o trabalho seria orientado a conjuntos de dados públicos.

Portanto, diante dessa limitação, se sugere como trabalho futuro a aplicação em grande volume de dados corporativos. Além de uma variação deste trabalho, determinando se, com base no perfil de cada funcionário, estes detêm um padrão de empregado apropriado às atribuições do cargo, considerando este padrão algo determinado por um profissional especialista na área ou pela detecção de padrões de características identificados por algoritmos de *clustering*.

Seguem outras sugestões de pesquisa para o tema: prever o período que determinado funcionário irá permanecer na empresa ou ainda prever e projetar as suas promoções de cargo dentro da corporação, baseado nas suas características pessoais, comportamentais e profissionais.

Tais cenários sugeridos podem apoiar especialistas de recursos humanos no entendimento de quais características e comportamentos faltam para que determinados funcionários atinjam de forma plena as atribuições exigidas ao cargo, consigam projetar futuras evasões e promoções de função na corporação. Com o conhecimento obtido, as empresas podem desenvolver estratégias visando melhorar panoramas futuros.

Referências

H. Jantan, A. R. Hamdan, Z. A. Othman. Data Mining Classification Techniques for Human Talent Forecasting. Malásia, 2011.

Z. Ö. K. Lu. Employee Turnover Prediction Using Machine Learning Based Methods. Dissertação de Mestrado, Middle East Technical University, Turquia, 2014.

A. M. E. Sikaroudi, RouzbehGhousi, A. EsmaieeliSikaroudi. A Data Mining Approach To Employee Turnover Prediction (Case Study: Arak Automotive Parts Manufacturing). Journal of Industrial and Systems Engineering, vol. 8, Nro. 4, páginas 106-121, 2015.

M. A. Valle, G. A. Ruz. Turnover Prediction in a Call Center: Behavioral Evidence of Loss Aversion using Random Forest and Naïve Bayes Algorithms. Chile, 2015.

R. Veldhuis. Predicting Individual Employee Turnover Using Machine Learning Techniques. 2017.

H. Yousaf. Analysing Which Factors Are Of Influence In Predicting The Employee Turnover. Vrije Universiteit, Holanda, 2016.

Zhao Y., Hryniewicki M.K., Cheng F., Fu B., Zhu X. Employee Turnover Prediction with Machine Learning: A Reliable Approach. Intelligent Systems and Applications. IntelliSys - Advances in Intelligent Systems and Computing, vol. 869. Springer, 2018.

D. S. Sisodia, S. Vishwakarma, A. Pujahari. Evaluation of Machine Learning Models for Employee Churn Prediction. International Conference on Inventive Computing and Informatics (ICICI), Índia, 2018.

R. Yedida, R. Reddy, R. Vahi, R. J. Jana, A. Gv, D. Kulkarni. Employee Attrition Prediction. 2018.

G. Ying, Using Decision Tree to Analyze the Turnover of Employees. Uppsala University, Suécia, 2017.

D. Alao, A. B. Adeyemo, Analyzing Employee Attrition Using Decision Tree Algorithms. Computing, Information Systems & Development Informatics, Vol. 4, Nr. 1, Nigéria, 2013.

A. Burkov. The Hundred-Page Machine Learning Book. Andriy Burkov, Canadá, 2019.

K. Facelli, A. C. Lorena, J. Gama, A. C. P. L. F. Carvalho. Inteligência Artificial - Uma Abordagem de Aprendizado de Máquina. LTC, Brasil, 2017.

D. J. Sweeney, T. A. Williams, D. R. Anderson. Estatística Aplicada à Administração e Economia - tradução da 6ª edição norte-americana. Cengage Learning, EUA, 2014.

De Coninck, J. B., Johnson, J. T. (2009). The effects of perceived supervisor support, perceived organizational support, and organizational justice on turnover among salespeople. Journal of Personal Selling & Sales Management, 29(4), 333-350.

Hancock, J. I., Allen, D. G., Bosco, F. A., McDaniel, K. R., & Pierce, C. A. (2013). Meta-analytic review of employee turnover as a predictor of firm performance. Journal of Management, 39(3), 573-603.

Russel, S.; Norvig, P. Inteligência Artificial. 3. ed. Rio de Janeiro, Brasil, 2013.

Chakrabarty, N., Kundu, T., Dandapat, S., Sarkar, A., Kole, D., K. Flight Arrival Delay Prediction Using Gradient Boosting Classifier. 2018.

Tharwat, A., Gaber T., Ibrahim, A., Hassanien, A., E. Linear discriminant analysis: A detailed tutorial. Egito, 2017.

Pednekar, A., M. Optimal initialization of K-means using Particle Swarm Optimization. 2019.

Bholowalia, P., Kumar, A. EBK-Means: A Clustering Technique based on Elbow Method and K-Means in WSN. 2014.

Øyvind, G., Myklebust, H., Hallén, J., Federolf, P. Technique Analysis in Elite Athletes Using Principal Component Analysis, Journal of Sports Sciences. 2017.

Aït-Sahalia, Y. Xiu, D.. Principal Component Analysis of High Frequency Data, Journal of the American Statistical Association. 2017.

Rousson, V., Gasser, T. Simple Component Analysis. Appl. Statist, vol. 53, edição 4, páginas 539–555, 2004.

Kohavi, R.; Provost, F.. Glossary of terms. Machine Learning, vol. 30, páginas 271-274, 1998.

Zeng, G. On the Confusion Matrix in Credit Scoring and Its Analytical Properties, Communications in Statistics - Theory and Methods. 2019.

Starczewski, A., Krzy˙zak, A. Performance Evaluation of the Silhouette Index. Polônia, 2015.

Shafique, U., Qaiser, H. A Comparative Study of Data Mining Process Models (KDD, CRISP-DM and SEMMA). International Journal of Innovation and Scientific Research, Vol. 12, Nro. 1, páginas 217-222, 2014.

Dutta, D., Barai, S., V., Prediction of Compressive Strength of Concrete: Machine Learning Approaches. In: Rao A., Ramanjaneyulu K. (eds) Recent Advances in Structural Engineering, Volume 1. Lecture Notes in Civil Engineering, Vol. 11. Springer, Singapore. 2019.

Li Y., Cui. W. Identifying the mislabeled training samples of ECG signals using machine learning. Biomedical Signal Processing and Control, Vol. 47. Elsevier, páginas 168-176. China.

P. S. Kuyven. Métodos Estatísticos Aplicados ao Processo Decisório. Coleção EAD – Editora Unisinos. São Leopoldo, RS, Brasil, 2010.

M. T. Ribeiro, S. Singh, C. Guestrin. “Why Should I Trust You?” Explaining the Predictions of Any Classifier. 2016.

Hui E.G.M. Learn R for Applied Statistics. Apress, Berkeley, CA. 2019.

L. Hulstaert. Understanding Model Predictions With LIME. https://towardsdatascience.com/understanding-model-predictions-with-lime-a582fdff3a3b, acessado em Agosto de 2019. Publicação: 2018.

Google Colab. https://colab.research.google.com/. Agosto de 2019.