Exploração de modelos de aprendizado de máquina e seleção de atributos para Employee Attrition

Rudson Francicso da Silva Mendes
rudsonfsmendes@gmail.com
Universidade Federal de Santa Catarina
Florianópolis, SC, Brasil

João Victor Ribeiro de Jesus
joao.v.ribeiro@outlook.com
Universidade do Vale do Itajaí
Itajaí, SC, Brasil

RESUMO

Employee Attrition can be defined as the process by which employees leave a company, for both personal or professional reasons. In this work, several machine learning algorithms were sought to predict possible Attritions in order to assist companies in the management of their employees' working conditions. The features used were selected using specialized algorithms for detecting the relevance and correlations between them. The results obtained were generated through experiments using different input parameters. As a result, it was possible to achieve approximately an 94% accuracy rating and a 93% to the $f1$ factor.

KEYWORDS

Aprendizado de Máquina, Seleção de Recursos, *Employee Attrition*

1 INTRODUÇÃO

Employee Attrition pode ser definido como o processo natural, e de forma não prevista, pelo qual um conjunto de colaboradores deixam de fazer parte de uma determinada organização. O dicionário Barron Business, define o termo *attrition* como a redução normal e descontrolada da força de trabalho devido à aposentadoria, morte, doenças e realocação [1]. O *attrition* acarreta na redução de uma força de trabalho, como, por exemplo, colaboradores de uma empresa, sem que a gerência possa tomar alguma ação prévia acerca desse problema [2].

O *attrition* é uma parte inevitável de qualquer negócio. Chegará um momento em que algum colaborador precisará deixar sua empresa, seja por motivos pessoais ou profissionais. Entretanto, o *attrition* pode ser tão grande a ponto de reduzir a força de trabalho de uma empresa em um nível que coloque em risco o seu negócio, tornando este problema um motivo de preocupação. Como, por exemplo, o *attrition* em grupos minoritários de colaboradores pode vir a prejudicar a diversidade da organização. Assim como, o *attrition* entre líderes pode vir a criar uma lacuna significativa na liderança organizacional [3].

Caso uma organização tenha conhecimento dos porquês, motivos ou fatores que levam seus colaboradores a deixarem a organização, a mesma pode desenvolver políticas e estratégias para reter estes colaboradores. E então, diminuir o índice de *attrition* e consequentemente o risco de seu negócio. Muitos colaboradores tendem a realizar várias transições entre trabalhos durante sua carreira profissional [4].

A proposta do presente trabalho foi criar um algoritmo capaz de prever, com base nos atributos considerados mais relevantes, a saída de um colaborador de uma dada empresa. Para tal, foi realizado um estudo sobre os diferentes métodos de seleção de atributos, a fim de obter o melhor subconjunto desses. Para realizar a classificação preditiva, propriamente dita, sobre um *dataset* (conjunto de dados) com o melhor subconjunto de atributos, foi realizada uma exploração acerca de alguns algoritmos de aprendizado de máquina para determinar aquele que melhor resolve o problema em questão, como por exemplo, árvores de decisão, máquinas de vetor de suporte (SVM), algoritmos de clusterização, entre outros.

Na seção três deste trabalho são abordados os algoritmos utilizados para classificação dos dados. Dentre os algoritmos estudados foram Regressão Logística (LGR), Análise Discriminante Linear (LDA), K-vizinhos mais próximos (KNN) e Floresta de Decisão (RF). Todos os algoritmos citados na seção são de aprendizado de máquina.

As etapas de pré-processamento dos dados são discutidas na seção quatro. Primeiramente, o conjunto de dados é normalizado, para que os mesmos respeitem um intervalo único, com intuito de remover as distorções das diferentes escalas. Tendo em consideração que o problema em questão é uma classificação binária, o balanceamento dos dados é realizado de forma que, a quantidade de exemplos de cada uma das classes resultantes (*Yes* e *No*) seja igualada. O algoritmo utilizado para o balanceamento é chamado de ADASYN e é discutido no mesmo seção.

Alguns atributos do *dataset* não apresentam relevância alguma para a classificação correta dos dados e, por esta razão, foi selecionada uma quantidade fixa de atributos, não muito grande, por meio de algoritmos especializados em detectar a relevância e correlações entre esses atributos e a variável dependente. No seção cinco, são discutidos três métodos diferentes para a seleção de atributos.

Na seção seis é explicado como foram realizados os experimentos e análise dos resultados obtidos a partir dos algoritmos de classificação sobre o melhor subconjunto de atributos encontrado. Por fim, as conclusões deste trabalho são apresentadas na seção sete.

2 TRABALHOS RELACIONADOS

Saídas voluntárias de colaboradores tem sido uma das principais preocupações que empresas possuem, isso devido as consequências que esses eventos podem ocasionar. Como por exemplo, colaboradores que possuem um papel vital na empresa, ao saírem, podem prejudicar no andamento da mesma, e geralmente são difíceis de serem repostos [5]. Pesquisadores vem estudando quais são os principais fatores que impactam na decisão de saída de colaboradores em empresas. Como, por exemplo, [6] e [7], que descobriram que o aumento de recompensas, como salário, contribuem fortemente tanto para a redução do *attrition* quanto para o aumento do desempenho dos colaboradores.

Em [8], os autores utilizaram diversos métodos de machine learning com o intuito de classificar o *attrition* de colaboradores.

Utilizaram o mesmo *dataset* que o presente trabalho, balanceando o mesmo com o algoritmo *ADASYN*, também utilizado no presente trabalho, e então obtiveram, por meio do t-test, um total de 12 atributos mais relevantes. Em seu estudo, conseguiram obter um valor para F1 de 0.92 utilizando uma árvore de decisão aleatória, considerando apenas 2 atributos e 0.90 com as 12 mais relevantes levantadas em seu estudo. Entretanto, há outros métodos de seleção de atributos relevantes que não foram explorados.

No estudo de [9] aplicado ao Swedbank, foram desenvolvidos modelos preditivos para o *attrition* de colaboradores. Em seu estudo, obtiveram 98, 6% de acurácia por meio de um modelo de floresta aleatória, inclusive se saindo melhor que modelos como máquinas de vetor de suporte (SVM) e uma rede neural multi-layer perceptron (MLP). Denotando, em seu trabalho, que o modelo de floresta aleatória resolve o problema com alta eficácia.

Em [10], os autores se propõem a prever o *attrition* de colaboradores utilizando um *dataset* que contém 1575 exemplos e 25 atributos. Em seu trabalho, são utilizados diversos algoritmos de mineração de dados, sendo estes: naive bayes, máquinas de vetor de suporte (SVM), regressão logística, árvores de decisão e florestas aleatórias. Os resultados de sua pesquisa resultaram em uma acurácia de 84, 12% por meio de um modelo de máquina de vetor de suporte.

3 MÉTODOS DE CLASSIFICAÇÃO

Uma série de modelos de classificação por meio de aprendizado de máquina foram utilizados para inferir sobre uma base de dados. Estes modelos de classificação serão brevemente abordados a seguir.

Regressão Logística (LGR) se trata de um modelo estatístico que faz parte dos modelos lineares genéricos. Este modelo, em sua forma básica, utiliza uma função logística para modelar uma variável dependente categórica, frequentemente binária [11, 12].

Análise Discriminante Linear (LDA) é uma generalização do método estatístico Discriminante Linear de Fisher. O método LDA tem por objetivo encontrar uma combinação linear de características que rotula ou separa duas ou mais classes de objetos ou eventos. A combinação resultante deste modelo pode ser utilizada como um classificador linear [13, 14].

K-vizinhos mais próximos (KNN) é um algoritmo de aprendizado de máquina utilizado, tanto para classificação, quanto para regressão. O KNN recebe um parâmetro *K*, que indica a quantidade de dados de treinamento que serão selecionados para a posterior classificação de novos dados. Cada novo exemplo é classificado baseado majoritariamente em seus K-vizinhos mais próximos [15].

Floresta de Decisão Aleatória (RF) é um método de aprendizado de máquina supervisionado. Este método constrói diversas árvores de decisão no momento de seu treinamento [16], podendo ser utilizado, tanto para classificação, quanto para regressão, que respectivamente são obtidas por meio da moda e média das árvores de decisão individuais [17]. O método é bastante utilizado por corrigir a tendência de overfitting que possa vir a aparecer em uma única árvore de decisão [18].

Estando entre os modelos de redes bayesianas mais simples, os classificadores Naive Bayes são uma família de classificadores probabilísticos que baseiam-se em aplicar o teorema de Bayes, assumindo um certo nível elevado de independência entre as atributos [19]. Entretanto, para atingir níveis maiores de acurácia, pode-se utilizar estimativa de densidade kernel [20, 21]. Neste trabalho são utilizadas as redes bayesianas gaussiana, multinomial e de Bernoulli.

4 PRE-PROCESSAMENTO

Nesta seção serão abordados os procedimentos utilizados para a transformação e preparação dos dados para efetivar a classificação e seleção de atributos. Primeiramente, será apresentado o *dataset* utilizado. Após isso, serão listadas as ferramentas utilizadas neste trabalho, assim como suas respectivas versões. Por último, os métodos adotados para a normalização e balanceamento dos dados.

4.1 Dataset

Neste estudo foi utilizado um conjunto de dados de acesso público, obtido pelo IBM Watson Analytics. Este conjunto contém dados artificiais projetados por cientistas de dados da IBM, entre eles, os dados utilizados foram os 1470 registros de colaboradores com 35 atributos. Destes registros, um total de 1233 são da categoria "*No*" para *Attrition*, enquanto os 237 restantes fazem parte dos que possuem a categoria "*Yes*" para *Attrition*.

De um total de 35 atributos, os atributos *EmployeeCount*, *Over18* e *EmployeeID* foram removidos devido ao fato de apresentarem o mesmo valor para todos os 1470 colaboradores. Além disso, todos os valores textuais foram convertidos para valores numéricos, como, por exemplo, para o atributo *Department*, os valores *Sales*, *Research & Development* e *Human Resources*, foram convertidos respectivamente para 1, 2 e 3.

Cada um dos registros do *dataset*, isto é, cada colaborador possui 30 atributos, os quais foram utilizados neste trabalho, que são:

- *Age*: idade;
- *Attrition*: desligamento do colaborador;
- *BusinessTravel*: frequência de viagens a trabalho;
- *DailyRate*: nível de salário;
- *Department*: departamento;
- *DistanceFromHome*: distância da residência e sede da organização;
- *Education*: nível de escolaridade;
- *EducationField*: campo de estudo;
- *EnvironmentSatisfaction*: satisfação com o ambiente de trabalho;
- *Gender*: gênero;
- *HourlyRate*: hora salarial;
- *JobInvolvement*: engajamento no trabalho;
- *JobLevel*: nível hierárquico;
- *JobRole*: função;
- *JobSatisfaction*: satisfação com a função;
- *MaritalStatus*: estado cívil;
- *MonthlyIncome*: salário mensal;
- *MonthlyRate*: ganho médio mensal;
- *NumCompaniesWorked*: quantidade de organizações em que já trabalhou;
- *OverTime*: realiza horas extras;
- *PercentSalaryHike*: percentual acrescentado ao salário;
- *PerformanceRating*: avaliação de performance;
- *RelationsSatisfaction*: satisfação com relações entre colegas de trabalho;
- *StandardHours*: quantidade média de horas trabalhadas;

- *StockOptionsLevel*: nível opção de compra de ações;
- *TotalWorkingYears*: quantidade total de anos já trabalhados;
- *TrainingTimesLastYear*: horas utilizadas em treinamento no último ano;
- *WorkLifeBalance*: equilíbrio entre vida pessoal e trabalho;
- *YearsAtCompany*: quantidade de anos em que atua na organização atual;
- *YearsInCurrentRole*: quantidade de anos em que atua no cargo atual;
- *YearsSinceLastPromotion*: quantidade de anos desde sua última promoção; e
- *YearsWithCurrentManager* quantidade de anos em que está sob liderança de seu atual líder.

4.2 Ferramentas

A linguagem Python em sua versão 3.7 foi utilizada, junto com as bibliotecas mais famosas de cunho analítico e de inteligência artificial onde, na **Tabela 1** abaixo, são listadas juntamente com suas respectivas versões.

Tabela 1: Bibliotecas utilizadas na implementaçao do trabalho, com suas respectivas versões.

Biblioteca	Versão
Pandas	1.0.3
Numpy	1.18.4
Seaborn	0.10.1
Imblearn	0.4.3
Sklearn	0.22.2.post1
Matplotlib	3.2.1

4.3 Normalização

Com intuito de remover as distorções das diferentes escalas que cada *feature* pode apresentar, uma normalização sobre o conjunto de dados foi realizada. A normalização teve como objetivo deixar os dados em uma única escala, como por exemplo, de -1 à 1. Devido a inviabilidade do uso do método Multinomial Naive Bayes para processos que utilizem números negativos, o seguinte método de normalização foi utilizado

$$f(X_i) = \frac{max(X) - X_i}{max(X) - min(X)} \quad (1)$$

sendo X_i uma amostra de uma *feature* do *dataset* e X todas amostras da mesma *feature*.

4.4 Balanceamento de Dados

No *dataset* há um desbalanceamento de classes, sendo 1233 para a classe *No* e 247 para *Yes*. Este comportamento pode causar um enviesamento para os modelos de classificação. Portanto, para contornar esse comportamento, foi utilizado um método de balanceamento dos dados, em outras palavras, foram adicionados exemplos artificiais de modo a tornar o conjunto de dados balanceado, compostos por, em média, 50% de cada classe.

Utilizando o método de *oversampling* foi possível adicionar exemplos artificiais de determinadas classes desejadas. Para este trabalho, o algoritmo *ADASYN* foi utilizado, adicionando exemplos apenas para a classe minoritária, isto é, para a classe com a menor quantidade de exemplos. Este algoritmo utiliza o parâmetro k para gerar exemplos artificiais com base em K-exemplos.

Utilizando $k = 5$ o número de exemplos da classe *Yes* aumentou de 247 para 1216, enquanto a classe *No* permaneceu com 1233. Então, o novo conjunto de dados é composto em 50, 35% de exemplos da classe *Yes* enquanto 49, 65% da classe *No*.

5 SELEÇÃO DE FEATURES

Datasets podem apresentar um número grande de atributos. Algumas dessas atributos são consideradas como ruídos e não apresentam influência alguma para os algoritmos de aprendizado de máquina. Além disso, quanto maior for a quantidade de atributos utilizados, maior será a complexidade envolvida, afetando diretamente, tanto a performance, quanto o tempo de treinamento dos algoritmos.

Para determinar qual a melhor quantidade de atributos a ser utilizada para a classificação dos dados, foi realizada uma classificação iterativa. Partindo de um conjunto F de atributos contendo apenas uma *feature*, a cada nova iteração é adicionado a este conjunto, a *feature* f que possui maior relevância, tal que $f \ni F$. Por exemplo, se $F = OverTime$, de acordo com T-test, a próxima *feature* a ser incluída é *MaritalStatus*, definindo então $F = OverTime, MaritalStatus$. O classificador empregado nesta etapa é o *Random Forest Classifier* (RFC), sendo atribuído um número de estimadores igual a 100.

Há diferentes métodos que podem ser utilizados para avaliar e ranquear os atributos de um *dataset*. Neste trabalho foram testados três métodos de *feature selection*: K-best, *Recursive feature elimination* e t-test.

5.1 K-best Feature Selection

Por meio do teste chi2, é possível quantificar o quanto duas variáveis categóricas são dependentes e que possuem maior influência sobre a classificação. Combinada com o método de *feature selection* K-best, é possível encontrar as K atributos que possuem os maiores valores para o teste chi2. Testando para valores de K de 1 até a quantidade total de atributos, é obtido um ranqueamento de atributos ilustrado na **Tabela 2**.

5.2 Recursive Feature Elimination

Recursive Feature Elimination (RFE) é um método de *feature selection*, que iterativamente remove os atributos mais fracos até que um número especificado de atributos é alcançado, como por exemplo, um parâmetro K. Recursivamente, este método elimina um número pequeno definido por step, de atributos a cada iteração, acabando por eliminar dependências e colinearidades que podem vir a existir em um modelo.

RFE precisa de um parâmetro K que define quantos atributos devem ser mantidos. Entretanto, não é de conhecimento geral, ou algo concreto que aponte quantos atributos devem, de fato, serem mantidos. Para encontrar o melhor número de atributos, isto é, os atributos que resultam em um melhor valor de classificação,

Tabela 2: Ranqueamento das atributos por meio do K-best feature selection.

Rank	Feature
1	OverTime
2	MaritalStatus
3	JobSatisfaction
4	StockOptionLevel
5	EnvironmentSatisfaction
6	Gender
7	TotalWorkingYears
8	YearsWithCurrManager
9	NumCompaniesWorked
10	EducationField
11	YearsInCurrentRole
12	YearsAtCompany
13	PercentSalaryHike
14	JobRole
15	RelationshipSatisfaction
16	PerformanceRating
17	YearsSinceLastPromotion
18	HourlyRate
19	Department
20	DistanceFromHome
21	JobInvolvement
22	Age
23	MonthlyRate
24	TrainingTimesLastYear
25	JobLevel
26	MonthlyIncome
27	BusinessTravel
28	DailyRate
29	Education
30	WorkLifeBalance

uma validação cruzada pode ser realizada. Junto com RFE, essa validação cruzada separa em diferentes subconjuntos de atributos, quantificando-os e selecionando os melhores atributos.

Utilizando um estimador RFC, validação cruzada com 10 subconjuntos, valor 1 para step e métrica

$$F1 = \frac{2 * precision * recall}{precision + recall} \tag{2}$$

para seleção dos atributos, foi obtido um ranqueamento mostrado pela **Tabela 3**.

5.3 T-test feature selection

O método t-test calcula a média e desvio padrão para classes binárias, o que é o caso do presente trabalho. Este método é definido por

$$t(X_i) = \frac{|\mu_{i1} - \mu_{i2}|}{\sqrt{\frac{\alpha_{i1}^2}{n_1} + \frac{\alpha_{i2}^2}{n_2}}} \tag{3}$$

Tabela 3: Ranqueamento das atributos por meio do RFECV.

Rank	Feature
1	Age
1	YearsInCurrentRole
1	YearsAtCompany
1	WorkLifeBalance
1	TrainingTimesLastYear
1	TotalWorkingYears
1	StockOptionLevel
1	RelationshipSatisfaction
1	PercentSalaryHike
1	OverTime
1	NumCompaniesWorked
1	MonthlyRate
1	MonthlyIncome
1	YearsSinceLastPromotion
1	MaritalStatus
1	JobRole
1	JobLevel
1	JobInvolvement
1	HourlyRate
1	EnvironmentSatisfaction
1	EducationField
1	DistanceFromHome
1	Department
1	DailyRate
1	BusinessTravel
1	JobSatisfaction
1	YearsWithCurrManager
2	Education
3	Gender
4	PerformanceRating

onde μ_{ij} representa a média dos exemplos da *feature* i e classe j; α_{ij} é o desvio padrão dos exemplos da *feature* i e classe j; e n_j é o total de exemplos que pertencem a classe j. Executando este método para o conjunto de dados balanceado e normalizado, foi obtido o ranqueamento apresentado **Tabela 4**.

6 EXPERIMENTAÇÃO E RESULTADOS

Foram realizadas 30 classificações para cada um dos 3 ranks de *feature* apresentados anteriormente, com intuito de definir qual o conjunto de atributos a ser utilizado. Por meio de um modelo de floresta aleatória, é realizada uma classificação utilizando a *feature* com maior rank, em seguida é adicionada a *feature* com o segundo maior rank para a próxima classificação, posterior a essa adicionou-se a com o terceiro maior rank, e assim por diante até finalizar as 30 classificações. O resultado deste processo, pode ser visto na Tabela 5.

Tabela 4: Ranqueamento dos atributos por meio do método T-test.

Rank	Feature
1	OverTime
2	MaritalStatus
3	StockOptionLevel
4	YearsWithCurrManager
5	YearsInCurrentRole
6	JobSatisfaction
7	YearsAtCompany
8	TotalWorkingYears
9	EnvironmentSatisfaction
10	YearsSinceLastPromotion
11	EducationField
12	NumCompaniesWorked
13	PercentSalaryHike
14	JobRole
15	Gender
16	PerformanceRating
17	Department
18	JobInvolvement
19	HourlyRate
20	TrainingTimesLastYear
21	DistanceFromHome
22	Age
23	JobLevel
24	RelationshipSatisfaction
25	MonthlyRate
26	BusinessTravel
27	MonthlyIncome
28	DailyRate
29	Education
30	WorkLifeBalance

São utilizados 11 modelos para experimentação, 4 destes modelos são KNN com diferentes parâmetros para k. Devido aos testes realizados anteriormente com métodos de *feature selection* apontarem que é o conjunto que apresenta o maior índice para $f1$. Então, foi utilizado um conjunto com as 23 atributos mais relevantes apontados por meio do RFECV. Para garantir a reprodutibilidade, durante todas as etapas deste trabalho, como por exemplo, balanceamento de *dataset*, *feature selection* e esta etapa de experimentação, foi definida uma semente de pseudo aleatoriedade com valor 7.

Para cada modelo utilizado, foi empregada uma validação cruzada, cujo parâmetro de *batches* é de 10. Por meio da validação cruzada, é possível diminuir drasticamente a tendência e enviesamento do *dataset* quando separado em conjuntos de treino, teste e validação. Isto é, devido a validação cruzada treinar com 9 *batches* dos dados e sempre utilizar o remanescente para teste, esse processo se repete 10 vezes, até que todos os *batches* sejam utilizados para teste.

Com base na Tabela 6, nota-se os resultados obtidos por meio dos métodos através da validação cruzada. O método que melhor obteve desempenho é o RFC, apresentando os maiores valores para as métricas acurácia, recall e f1. Enquanto, os algoritmos KNN se saem melhores em relação à precisão, apresentando índices elevados, chegando a até 100% quando utilizado um valor de $k = 1$. Importante ressaltar que o modelo de máquina de vetores de suporte de função de base radial (SVG), atingiu valores maiores que 85% para todas as métricas, inclusive sua precisão se mostrou cerca de 2% maior que RFC.

Tabela 5: Valores de $f1$ para diferentes quantidades de atributos por meio de diferentes métodos. Os valores em negrito são os melhores para cada método de *feature selection*. Os melhores resultados de cada métrica estão detacados em negrito.

Quantidade de atributos	T-Test	K-best	RFECV
1	0,6443	0,6443	0,8708
2	0,7671	0,7671	0,8234
3	0,7952	0,8374	0,8125
4	0,8351	0,8503	0,8583
5	0,8377	0,8545	0,8605
6	0,8594	0,8461	0,8892
7	0,8568	0,8700	0,8834
8	0,8847	0,8803	0,8915
9	0,8795	0,8823	0,8947
10	0,8883	0,8928	0,9020
11	0,8846	0,8967	0,8999
12	0,8992	0,8964	0,9051
13	0,9158	0,9074	0,9079
14	0,9229	0,9059	0,9128
15	0,9141	0,9206	0,9042
16	0,9086	0,9118	0,9133
17	0,9111	0,9214	0,9086
18	0,9179	0,9137	0,9154
19	0,9115	0,9214	0,9091
20	0,9188	0,9155	0,9125
21	0,9103	0,9198	0,9186
22	0,9253	0,9264	0,9292
23	0,9196	0,9147	**0,9303**
24	0,9164	0,9262	0,9247
25	0,9183	0,9183	0,9223
26	0,9154	0,9133	0,9204
27	0,9245	0,9245	0,9189
28	0,9202	0,9202	0,9140
29	0,9208	0,9208	0,9249
30	**0,9299**	**0,9299**	0,9299

7 CONCLUSÃO

Quanto maior o índice de *attrition* em uma organização, maior é o risco em que a mesma está sujeita. Principalmente, as empresas que investem em seus colaboradores, a perda daqueles que apresentam alta performance é muito prejudicial. Encontrar colaboradores com níveis de performance similares a estes que saíram pode ser considerada uma tarefa difícil e custosa tanto em termos de dinheiro quanto tempo.

Tabela 6: Lista ordenada pela acurácia dos resultados obtidos utilizando os 23 melhores atributos do RFECV. Os melhores valores para cada métrica estão destacados em negrito.

Modelo	Acurácia	Precisão	Recall	F1
RFC	**0.936096**	0.908448	**0.970823**	**0.938531**
SVG	0.869296	0.912636	0.818345	0.862656
DTC	0.844078	0.859032	0.824823	0.841505
KNN-1	0.841193	**1.000000**	0.683687	0.811466
KNN-3	0.801304	0.997181	0.605862	0.753014
KNN-5	0.786244	0.994730	0.577439	0.729638
LGR	0.702389	0.712911	0.682127	0.696595
SVL	0.705638	0.727022	0.664274	0.693652
LDA	0.699939	0.713143	0.674003	0.692394
MNB	0.637624	0.615286	0.745384	0.673923
BNB	0.691377	0.732963	0.608196	0.662827
GNB	0.687287	0.755345	0.560392	0.642760
KNN-10	0.725180	0.984699	0.459920	0.626526
SVS	0.460096	0.479135	0.866988	0.617164

O principal objetivo deste estudo foi a utilização de modelos de aprendizado de máquina para classificar o *attrition* de colaboradores de uma organização, tendo em base seus atributos. O resultado disso auxiliará as organizações a estarem menos sujeitas a esse problema de *attrition*. Isto, devido ao suporte nas tomadas de decisões e predições para que a gerência da organização possa tomar ações mais rápidas e assertivas.

Por meio desse estudo, foi possível atestar que utilizando modelos de aprendizado de máquina, atingiu-se uma acurácia de classificação de *Employee Attrition* de até aproximadamente 94%. Além disso, o subconjunto de atributos que maximizou os resultados da classificação possui um total de 23 atributos. Este subconjunto foi obtido por meio do método RFECV. Um fato interessante descoberto foi que, utilizando apenas a *feature Age*, foi possível obter um índice de $f1$ de aproximadamente 87% para classificação.

Neste estudo os diferentes subconjuntos de atributos explorados foram elaborados de forma sequencial, respeitando a ordem de importância que os diferentes métodos de seleção definem. Entretanto, determinadas combinações de atributos, não respeitando a ordem de importância, podem vir a aumentar os resultados obtidos. Para isso, a aplicação de algoritmos aproximados podem realizar essas combinações de uma forma eficaz e rápida, por exemplo, com heurísticas. Além disso, neste trabalho não foi explorado o uso de redes neurais artificiais, apesar de existirem trabalhos similares que apresentam ótimos resultados para esse mesmo problema.

REFERÊNCIAS

[1] Douglas Downing. *Dictionary of Computer and Internet Terms (Barron's Business Dictionaries)*. 1996.
[2] Atanu Adhikari. Factors affecting employee attrition: a multiple regression approach. *IUP Journal of Management Research*, 8(5):38, 2009.
[3] K Lavanya Latha. A study on employee attrition and retention in manufacturing industries. *BVIMSR's Journal of Management Research (BJMR)*, 5(1):1–23, 2013.
[4] Nathan Bennett, Terry C Blum, Rebecca G Long, and Paul M Roman. A firm-level analysis of employee attrition. *Group & Organization Management*, 18(4):482–499, 1993.
[5] Sarabjeet Kaur and Ms Ritu Vijay. Job satisfaction-a major factor behind attrition of retention in retail industry. *Imperial Journal of Interdisciplinary Research*, 2(8): 993–996, 2016.
[6] Donald G Gardner, Linn Van Dyne, and Jon L Pierce. The effects of pay level on organization-based self-esteem and performance: A field study. *Journal of occupational and organizational psychology*, 77(3):307–322, 2004.
[7] Elisa Moncarz, Jinlin Zhao, and Christine Kay. An exploratory study of us lodging properties' organizational practices on employee turnover and retention. *International Journal of Contemporary Hospitality Management*, 2009.
[8] Sarah S Alduayj and Kashif Rajpoot. Predicting employee attrition using machine learning. In *2018 International Conference on Innovations in Information Technology (IIT)*, pages 93–98. IEEE, 2018.
[9] Mari Maisuradze. Predictive alaysis on the example of employee turnover. *Tallinn University Of Technology*, 2017.
[10] V Vijaya Saradhi and Girish Keshav Palshikar. Employee churn prediction. *Expert Systems with Applications*, 38(3):1999–2006, 2011.
[11] John Ashworth Nelder and Robert WM Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3):370–384, 1972.
[12] Daryl Pregibon et al. Logistic regression diagnostics. *The Annals of Statistics*, 9 (4):705–724, 1981.
[13] Chengjun Liu and Harry Wechsler. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *IEEE Transactions on Image processing*, 11(4):467–476, 2002.
[14] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.
[15] Simon Rogers and Mark Girolami. *A first course in machine learning*. CRC Press, 2016.
[16] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.
[17] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
[18] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8):832–844, 1998.
[19] Chris Pal and Andrew McCallum. Cc prediction with graphical models. In *CEAS*, 2006.
[20] S Madeh Piryonesi and Tamer E El-Diraby. Role of data analytics in infrastructure asset management: Overcoming data size and quality problems. *Journal of Transportation Engineering, Part B: Pavements*, 146(2):04020022, 2020.
[21] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.