Turnover de Funcionários: Insights e Previsões através de Modelos de Aprendizado de Máquina

Marcos A. Lommez C. R.
Pontifícia Universidade Católica
Belo Horizonte, Brasil
tonilommez@hotmail.com

Bernardo Marques Fernandes
Pontifícia Universidade Católica
Belo Horizonte, Brasil
bevgim@gmail.com

Frederico Malaquias
Pontifícia Universidade Católica
Belo Horizonte, Brasil
fmacaldeira@sga.pucminas.br

Pedro Soares
Pontifícia Universidade Católica
Belo Horizonte, Brasil
pedrogabrielbhz@gmail.com

Vitória de Lourdes
Pontifícia Universidade Católica
Belo Horizonte, Brasil
vitoria.lourdes@sga.pucminas.br

ABSTRACT

Turnover, the act of employees leaving a company, poses significant challenges for businesses across various sectors. Recognizing the elements that contribute to this behavior is essential for shaping more efficient retention strategies, which in turn can lead to cost savings and uphold operational continuity. In this study, we explore a dataset that captures employee attributes and their associated turnover outcomes. By employing a comparative analysis across multiple machine learning techniques, we aim to unearth key determinants influencing employee turnover. Through the optimization of these models, we aim to provide businesses with an insightful predictive tool, facilitating targeted interventions to curb turnover rates.

KEYWORDS

turnover, people analytics, classification, machine learning, hr analytics

ACM Reference Format:
Marcos A. Lommez C. R., Bernardo Marques Fernandes, Frederico Malaquias, Pedro Soares, and Vitória de Lourdes. 2023. Turnover de Funcionários: Insights e Previsões através de Modelos de Aprendizado de Máquina. In *Not Published: Internal Document, 2023, Belo Horizonte, Brasil*. ACM, New York, NY, USA, 9 pages.

Not Published '23, 2023, Belo Horizonte, Brasil

ACM ISBN N/A…$N/A

1 INTRODUÇÃO

O turnover de funcionários, definido por Anwar [2] como a taxa de funcionários na qual a empresa precisa repor em um dado período de tempo pelo numero total de funcionários, é dado como um dos maiores desafios financeiros e operacionais para empresas de todos os setores. A saída frequente de colaboradores pode resultar em perdas significativas: desde o investimento na formação e desenvolvimento destes profissionais até os impactos na produtividade e na continuidade de projetos. Além disso, o turnover pode levar a um aumento nos custos operacionais, especialmente em trabalhos de salario mínimo (Anwar e Shukur, 2015)[1]. Portanto, entender as motivações e causas do turnover não é apenas crucial para a saúde financeira da empresa, mas também para a preservação de um ambiente de trabalho coeso e eficiente.

No ambiente de negócios atual, onde as decisões são cada vez mais pautadas por análises data-driven, as técnicas de aprendizado de máquina emergem como soluções de ponta para prever e analisar o turnover (Barzilay, 2019)[6]. Modelos baseados nessas técnicas podem proporcionar insights valiosos, permitindo que empresas identifiquem, com antecedência, os sinais de possíveis saídas e, assim, implementem estratégias de retenção mais direcionadas e eficazes. Isso se torna ainda mais crucial em um mercado de trabalho altamente competitivo, onde a retenção de talentos é sinônimo de vantagem competitiva e inovação contínua.

Neste estudo, examinamos a eficácia de algoritmos de aprendizado de máquina em prever turnover, utilizando um dataset detalhado de classes de funcionários. Nosso foco é oferecer uma ferramenta preditiva econômica e eficiente às organizações.

2 DESCRIÇÃO DA BASE DE DADOS

A base de dados Employee Turnover [3] consiste em dados reais sobre funcionários de diversos setores e áreas, com foco no entendimento e previsão de turnover. Contando com um total de 1095 instâncias ela possui 16 classes distintas que

serão analisadas para classificar a classe target sendo esta a classe *EVENT*.

- `stag`: Tempo t quando ocorreu o turnover ou tempo de censura do estudo.
- `event`: Indica se o evento de sair da empresa ocorreu no tempo t.
- `gender`: Gênero do funcionário.
- `age`: Idade do funcionário.
- `industry`: Indústria de atuação do funcionário.
- `profession`: Profissão do funcionário.
- `traffic`: Canal pelo qual o funcionário ingressou na empresa.
- `coach`: Presença de um coach.
- `head_gender`: Gênero do supervisor.
- `greywage`: Se o empregador paga apenas uma pequena quantia de salário acima do salário mínimo (white).
- `way`: Forma que o funcionário vai ao trabalho.
- `extraversion`, `independ`, `selfcontrol`, `anxiety`, `novator`: Escalas segundo o teste Big5.

Os atributos são divididas entre 7 valores quantitativos contínuos (stag, age, extraversion, independ, selfcontrol, anxiety e novator), 4 valores qualitativos dicotômicos (event, gender, head_gender e greywage) e 5 valores qualitativos nominais (industry, profession, traffic, coach e way).

Table 1: Os atributos da base de dados Employee Turnover

Classe	Tipo	Valores permitidos
stag	Numérico	float
event	Categórico	[0, 1]
gender	Categórico	[m, f]
age	Numérico	[0, 100]
industry	Categórico	[lista de indústrias]
profession	Categórico	[lista de profissões]
traffic	Categórico	[lista de fontes de tráfego]
coach	Categórico	[0, 1]
head_gender	Categórico	[m, f]
greywage	Caractere	[white, grey]
way	Categórico	[lista de formas]
extraversion	Numérico	[0, 10]
independ	Numérico	[0, 10]
selfcontrol	Numérico	[0, 10]
anxiety	Numérico	[0, 10]
novator	Numérico	[0, 10]

3 PRÉ-PROCESSAMENTO DE DADOS

Uma etapa fundamental em qualquer projeto de aprendizado de máquina é o processamento dos dados para que seja possível garantir que o modelo seja capaz de interpretá-lo, além de otimizar seu funcionamento evitando problemas como overfitting, treinamento e validação com os mesmos valores, under e oversampling, entre outros. Primeiro será necessário fazer uma análise exploratória para entender melhor as características da base de dados, para, em seguida, através do processamento, realizar o treinamento dos modelos.

Nulos e repetições

Em bases do mundo real, convencionalmente existem valores nulos e instâncias repetidas. De acordo com a tabela 2, não foram encontrados valor nulo em nenhuma instância. Além disso, fizemos uma análise dos tipos de dados presentes em nosso conjunto de dados. Temos um total de 7 atributos com dados quantitativos contínuos e 8 atributos com dados qualitativos nominais.

Table 2: Resumo dos atributos da Base de Dados

Classe	Null Count	Dtype
stag	0	float64
event	0	int64
gender	0	object
age	0	float64
industry	0	object
profession	0	object
traffic	0	object
coach	0	object
head_gender	0	object
greywage	0	object
way	0	object
extraversion	0	float64
independ	0	float64
selfcontrol	0	float64
anxiety	0	float64
novator	0	float64

Em contrapartida, foram identificadas 13 instâncias duplicadas, o que não impacta negativamente a qualidade dos dados em caso de simples remoção da tabela. Os dados repetidos foram retirados da tabela, levando a um total de 1082 instâncias.

Detecção de Outliers

Essencialmente, em qualquer conjunto de dados, as observações que desviam de maneira significativa do padrão geral são frequentemente consideradas outliers ou anomalias. Esses pontos de dados extremos têm o potencial de impactar negativamente o treinamento de modelos de aprendizado de máquina. Isso ocorre porque eles podem introduzir variâncias não representativas e distorcer a interpretação estatística geral dos dados, levando, em muitos casos, a predições imprecisas ou tendenciosas.

Para avaliar e manejar esses outliers, a análise de quartil é uma técnica valiosa. Ela divide um conjunto de dados em quatro partes, facilitando a identificação de valores atípicos. O quartil pode ser calculado a partir de [4] :

$$Q_k = \frac{k(n+1)}{4} \tag{1}$$

onde:

Q_k é o valor do k-ésimo quartil

k é o número do quartil

n é o número total de observações

De acordo com a tabela 2, fornecida através da aplicação dos quartis na base de dados, podemos analisar que todos exceto St. (stag) respeitam uma proporção entre os quartis. Na mesma tabela podemos ver a variação entre Ag. (Age), Ex. (Extraversion), In. (Independ), S.C. (Selfcontrol), An. (Anxiety) e No. (novator).

Table 3: Quartis, minimo e maximo

	St.	Ag.	Ex.	In.	S.C.	An.	No.
Min	0.3	18	1	1	1	1	1
Q_1	11.7	26	4	4	4	4	4
Q_2	24.4	30	5	5	5	5	6
Q_3	51.6	36	7	6	7	7	7
Max	179.4	58	10	10	10	10	10

Esta prova pode ser retirada ao se aplicar a métrica de z-score, que descreve a posição de uma observação dentro de uma distribuição. Especificamente, ela indica quantos desvios padrão uma observação está em relação à média da distribuição. O z-score é calculado usando a seguinte fórmula [4] :

$$z = \frac{x - \mu}{\sigma} \tag{2}$$

onde:

x é o valor da observação;

μ é a média da distribuição)

σ é o desvio padrão da distribuição.

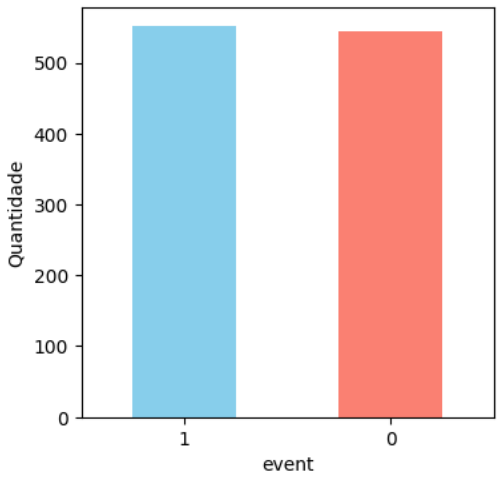
Usando um z-score aplicado a um limiar de corte 3, chegamos a um total de 21 instâncias a serem retiradas da base, o que modifica consideravelmente os valores de 'stag' para um mínimo de 0.3, quartis de 11.5, 24.0, 50.1 e um máximo de 139.0. Comprovando que a retirada dos outliers tiveram pouco impacto sobre os demais quartis e ainda sim diminuindo seu range máximo sem afetar consideravelmente a representatividade dos dados.

Observamos que a remoção dos outliers resultou em uma distribuição mais consistente dos dados.

Balanceamento

Uma análise de balanceamento dos dados se faz necessária para que o modelo não seja tendencioso ao votar na classe majoritária, o que resultaria em erros consideráveis em sua validação. A distribuição dos valores '0' e '1' na classe alvo é apresentada no gráfico 1, onde se torna possível afirmar que a base possui uma alta representatividade para ambos os rótulos.

Figure 1: Distribuição do atributo "event"



Transformação de Dados

Após a análise de remoção de instâncias inválidas para o treinamento, deve ser feito o tratamento de tipagem dos atributos para que seja possível treinar o modelo. Para isso, cada classe deve ser analisada individualmente para uma resposta adequada.

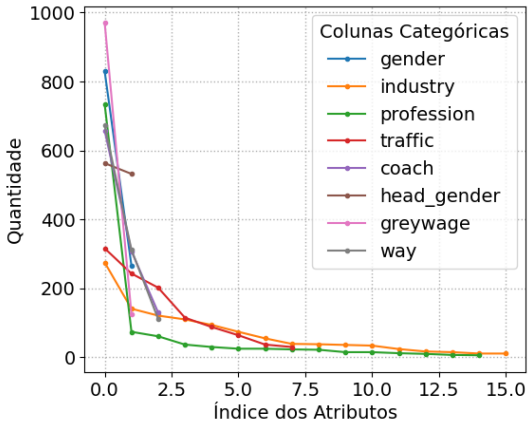
Qualitativos

Essencialmente, atributos que qualificam algo podem ser separados entre ordinais ou nominais e, para cada um destes, é necessário uma estratégia para prepará-los. Para tratar atributos dicotômicas, uma estratégia simples de convertê-las em valores binários 0 e 1 basta. Por outro lado, aos demais, se torna necessário um estudo de caso para tomada de decisão como mostrado a seguir.

Ao analisar a quantidade de possíveis instâncias por coluna através da figura 2, concluímos que:

- Existem 3 colunas binarizáveis (gender, head_gender e greywage) fora a target (event).

Figure 2: Distribuição do atributo categóricas

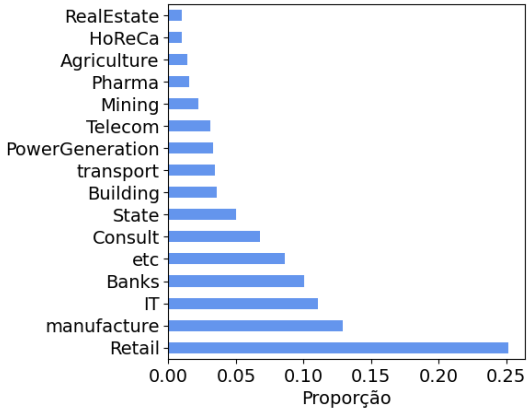


- Existem 2 colunas com um número pequeno o suficiente de categorias para aplicar Encoding (coach e way).
- Existem 3 colunas que requerem uma estratégia adicional para o tratamento.

Classe 'Industry'

A classe 'industry' é particularmente diversificada e representa vários setores distintos. Em uma inspeção preliminar, conforme ilustrado na Figura 3, é evidente que cada categoria possui uma distribuição significativa. Tal distribuição heterogênea pode ter implicações na modelagem e na precisão do modelo preditivo.

Figure 3: Distribuição da classe "industry"



Dada a diversidade e a relevância potencial da classe 'industry', foi escolhida uma abordagem de codificação baseada na relação com o turnover. Em vez de usar uma codificação one-hot, que poderia expandir excessivamente o espaço de recursos, optamos por uma técnica de codificação mais informativa. A estratégia consiste em substituir o valor da 'industry' pela porcentagem de casos de turnover associados a cada categoria específica. Isso não apenas reduz a dimensionalidade do dataset, mas também infunde o modelo com informações potencialmente mais significativas sobre a relação entre a indústria e o turnover.

Assim sendo, a relação de target para a classe 'industry' é calculada com o uso do mean encoding [4] :

$$\text{MeanEncoding}_{\text{categoria}} = \frac{\sum_{i \in \text{categoria}} \text{Target}_i}{\text{Contagem}_{\text{categoria}}} \tag{3}$$

onde:

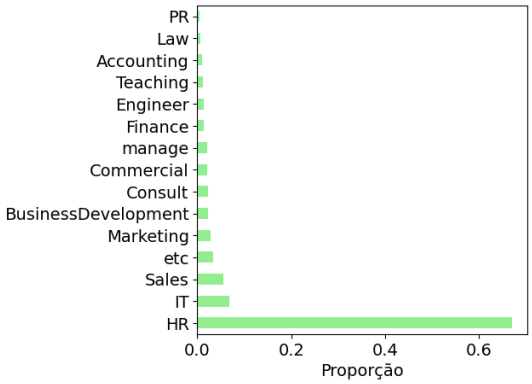
- $\sum_{i \in \text{categoria}} \text{Target}_i$: soma dos valores do target para todas as observações na categoria.
- $\text{Contagem}_{\text{categoria}}$: número total de observações naquela categoria.

Esta abordagem é especialmente útil a partir do momento em que o modelo seja capaz de retirar insights importantes pela captura e incorporação de informações úteis sobre a relação entre cada categoria e o target. Evita a criação de múltiplas colunas, levando a uma redução de dimensionalidade considerável comparada a métodos como One Hot Encoding. Por fim, pode ajudar na prevenção de overfitting, ajudando o modelo a se tornar mais robusto.

Classe 'Profession'

A classe 'profession' representada na figura 4 possui uma grande quantidade de instâncias agrupadas em uma única classe de maneira desbalanceada. Para tratá-la, categorias com uma quantidade de instâncias menor que "Etc" foram agrupadas nesta mesma classe, devido a possuírem uma quantidade média menor do que 5% em relação ao conjunto total.

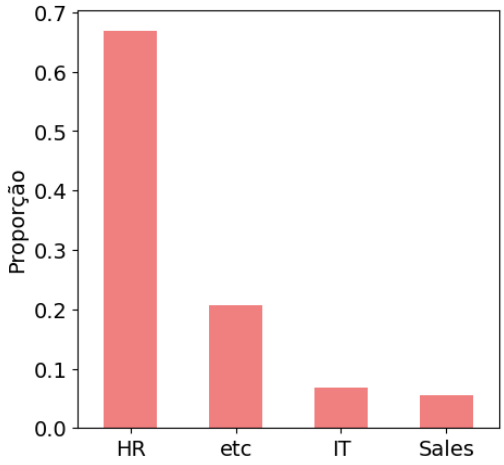
Figure 4: Distribuição da classe "profession"



Após agrupadas, as colunas se tornaram uma quantidade pequena o suficiente de rótulos, como mostra na figura 5,

consequentemente, se torna viável seguir por estratégias como o One Hot Encoding que foi a escolhida.

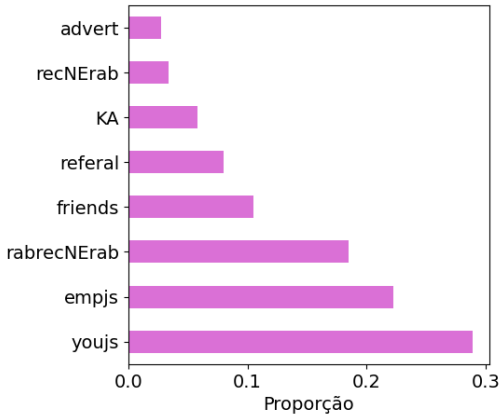
Figure 5: Distribuição da classe "profession_grouped"



Classe 'Traffic'

A classe 'traffic' representada na figura 6 apresenta um problema semelhante ao da classe 'industry', pois não possui um desequilíbrio muito grande. Portanto, é sugerido seguir o mesmo caminho de realizar um mean encoding.

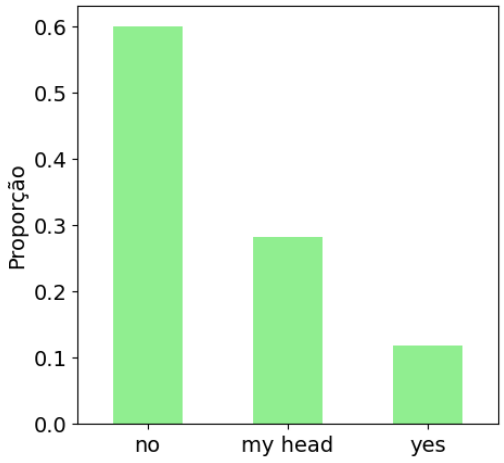
Figure 6: Distribuição do classe "Traffic"



Classe 'Coach'

A classe 'coach' pode ser considerada uma classe binária, mas, para a construção da base de dados, coach foi separado entre um agente externo (yes) ou o próprio treinador do funcionário (my head). Assim, realizamos um encoding tratando "my head" como um supertipo, mas sem considerar a ordem entre os dois. Para o mapeamento proposto temos:

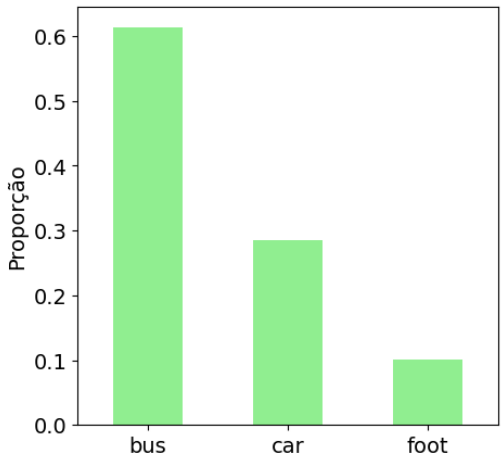
Figure 7: Distribuição do classe "Coach"



- no ⟶ 0
- my head ⟶ 1
- yes ⟶ 2

Classe 'Way'

Figure 8: Distribuição da classe "Way"



Os valores da classe 'way' não são possíveis de se relacionar diretamente, pois não possuem uma cardinalidade. Portanto, usamos uma estratégia de separá-los em três classes com One Hot Encoding, já que existem apenas 3 tipos de instâncias.

Quantitativos

Assim como a necessidade de detecção de outliers para se evitar que valores em uma grande escala possam influenciar o modelo, também se torna necessário normalizá-los para

garantir que o modelo preditor usado não dê mais atenção para uma classe do que outra dependendo de qual métrica for usada, principalmente para os modelos baseados em regressão com gradiente descendente, que são especialmente sensíveis a ordem de grandeza.

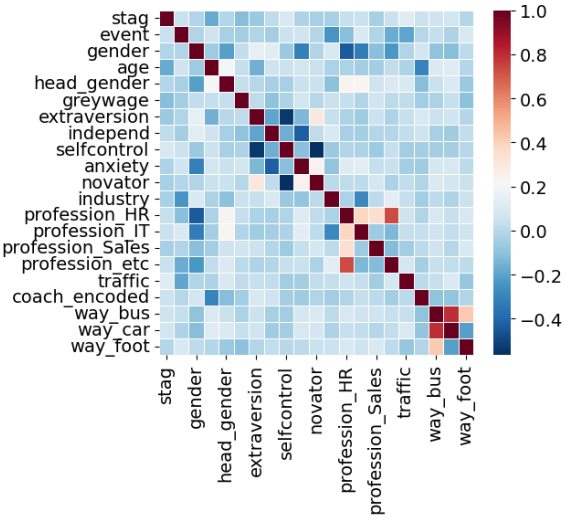
Assim como demonstrado na tabela 3, a única instancia que de fato destoa da ordem dos outros é a classe 'stag', por outro lado, outras classes numéricas vão para uma distância de 1 ate 10, enquanto diversas outras colunas que sofreram encoding possuem um padrão entre 0 e 1, para isso, todas as colunas que possuem um mínimo menor que 0 e um máximo maior que 1 serão aplicadas a fórmula de normalização [4] :

$$x_{\text{normalizado}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{4}$$

4 ANÁLISE EXPLORATÓRIA DE DADOS

É necessário analisar a correlação dos dados da tabela para entender se determinados valores são redundantes, o que podem levar a problemas que interferem no caráter estatístico dos modelos levando ao erro no aprendizado. Através da matriz de correlação em 9 entre os elementos podemos ver se o crescimento das classes segue direta ou inversamente o crescimento de determinada classe alvo, seguindo uma escala que começa de 1 se for diretamente proporcional ou até -1 se for inversamente proporcional.

Figure 9: Matriz de Correlação



A matriz de correlação foi plotada usando um mapa de calor. É possível identificar que existe correlação consideravelmente alta entre os valores de "way_car" e "way_bus", assim como entre "profession_etc" e "profession_HR". No caso de "way_car". Ambos os dois casos naturalmente possuem uma correlação devido ao fato de serem mutuamente excludentes durante seu processo de encoding, mas ainda assim seus valores podem interferir na avaliação final do modelo de maneira positiva ou negativa, assim faz-se necessário uma analise mais detalhada da correlação dos atributos que é feita a seguir, utilizando a correlação diretamente com a classe alvo:

Table 4: Correlação de classes em relação a Event

Classe	Correlação
way_foot	0.089897
profession_IT	0.089037
anxiety	0.060485
age	0.053539
selfcontrol	0.047828
stag	0.044558
way_bus	0.033207
coach_encoded	-0.008334
novator	-0.010980
gender	-0.016399
extraversion	-0.016989
profession_Sales	-0.018356
way_car	-0.024290
head_gender	-0.042759
greywage	-0.046527
independ	-0.051918
profession_HR	-0.108105
profession_etc	-0.170202
traffic	-0.196490
industry	-0.247072

Levando ambas as tabelas em conta e testes que foram realizados no modelo final, conclui-se que a pontuação geral do modelo cresce se retirarmos as classes gender, profession_IT, novator e coach_encoded, por isso estas foram retiradas da nossa base de dados, outros valores mesmo possuindo correlação levaram a uma piora geral no modelo por isso foram mantidos.

Métodologia de validação dos modelos

O método de validação escolhido para avaliar os modelos foi o Cross-Validation com 10 folds. Esta técnica divide o conjunto de dados em 'n' partes iguais, onde cada parte é usada como um conjunto de teste em iterações separadas, enquanto as outras 'n-1' partes compõem o conjunto de treinamento. Além disso, integramos este processo com o algoritmo de Grid-Search, que se baseia no Cross-Validation para avaliar os modelos sob diferentes configurações de hiper-parâmetros. Essa combinação não só ajuda a identificar a melhor configuração para cada modelo, e uma validação eficiente mas

Table 5: Métricas de performance de modelos

Modelo	Acurácia	Precisão	Recall	F1
Decision Tree	0.60	0.62	0.54	0.58
Random Forest	0.63	0.64	0.58	0.61
Gradient Boosting	0.59	0.58	0.64	0.60
Ada Boost	0.61	0.61	0.61	0.61
K-nearest	0.60	0.61	0.58	0.59
Naive Bayes	0.63	0.63	0.62	0.62
Logistic Regression	0.62	0.62	0.62	0.62
SVM	0.62	0.60	0.67	0.63
Neural Network	0.63	0.63	0.61	0.62

também uma medida mais justa e realista para a técnica implementada, garantindo assim resultados mais confiáveis e representativos.

5 APRENDIZADO DE MÁQUINA

O objetivo deste estudo é analisar diversos modelos para fornecer insights sobre o turnover. Para uma interpretação clara, focamos em quatro modelos baseados em arvore, sendo três modelos da família Ensemble: **Árvore de Decisão**, **Random Forest**, **Gradient Boosting** e **Ada Boost**. Além destes, avaliamos a performance de **Naive Bayes**, **Regressão Logística**, **Support Vector Machine (SVM)**, **K-Nearest Neighbors** e **Rede Neural**, sendo a última reconhecida por sua alta capacidade de generalização, comparável aos métodos Ensemble.

Modelos ensemble e baseados em redes neurais são especialmente aptos para dados que exigem correlações entre colunas, como é o caso da análise de comportamento humano no contexto de turnover. Dada essa complexidade, espera-se que tais modelos apresentem desempenho superior, com exceção do **Naive Bayes** que, apesar de sua limitação em analisar correlações, possui um caráter estatístico que pode garantir um bom rendimento.

Para modelos de regressão e de saída numérica como a **Rede Neural**, empregamos uma estratégia de limiarização através da função Sigmoid, dada por [4]:

$$S(x) = \frac{1}{1 + e^{-x}}$$

Ajustes de hiper parâmetros

Antes do treinamento de cada modelo faz-se necessário uma analise de quais parâmetros e configurações serão necessários para melhor adaptar os modelos ao problema. Para isso foi utilizado a estrategia do Grid Search que envolve uma testagem de combinações entre uma lista fornecida de parâmetros possíveis e avaliando através de uma validação crusada, que neste trabalho foi utilizado o valor de 10 combinações do cross-validation:

- `Decision Tree`: 'criterion': 'gini', 'max_depth': 10, 'min_samples_leaf': 6, 'min_samples_split': 5.
- `Random Forest`: 'criterion': 'entropy', 'max_depth': 40, 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 50.
- `Gradient Boosting`: 'criterion': 'friedman_mse', 'learning_rate': 1, 'loss': 'log_loss', 'n_estimators': 100.
- `Ada Boost`: 'algorithm': 'SAMME.R', 'learning_rate': 1, 'n_estimators': 50.
- `K-Nearest Neighbors`: 'algorithm': 'auto', 'leaf_size': 20, 'n_neighbors': 8, 'p': 1, 'weights': 'distance'.
- `Naive Bayes`: 'var_smoothing': 1e-09.
- `Regressão Logística`: 'C': 100, 'max_iter': 50, 'solver': 'lbfgs'.
- `Support Vector Machine`: 'C': 10, 'class_weight': None, 'coef0': 0.0, 'gamma': 'auto', 'kernel': 'sigmoid'.
- `Neural Network MLP`: 'activation': 'tanh', 'alpha': 0.0001, 'hidden_layer_sizes': (50, 100, 50), 'learning_rate': 'constant', 'solver': 'adam'.

6 ANÁLISE DOS RESULTADOS

Assim como pode ser analisado na tabela 5, dentre os modelos propostos, os métodos **Neural Network**, **SVM**, **Logistic Regression**, **Random Forest** e **Naive Bayes** se destacaram, alcançando acurácias que variaram de 62% a 63%.

Modelos baseados mais puramente em tecnicas estatística como o **Naive Bayes** e a **Regressão Logística** tiveram bom desempenho. Por outro lado, o modelo baseado em arvore de decisão como a propria **Decision Tree** e suas variações como o **Gradiente Boosting** embora tiveram desempenho muito proximo mas chegaram a uma pontuação levemente abaixa dos demais.

Recall vs Precisão

Priorizar a precisão está relacionado à ideia de reduzir o número de "alarmes falsos", o que é especialmente útil se a empresa planeja tomar medidas caras ou significativas para intervir, como oferecer bônus ou promoções para reter talentos.

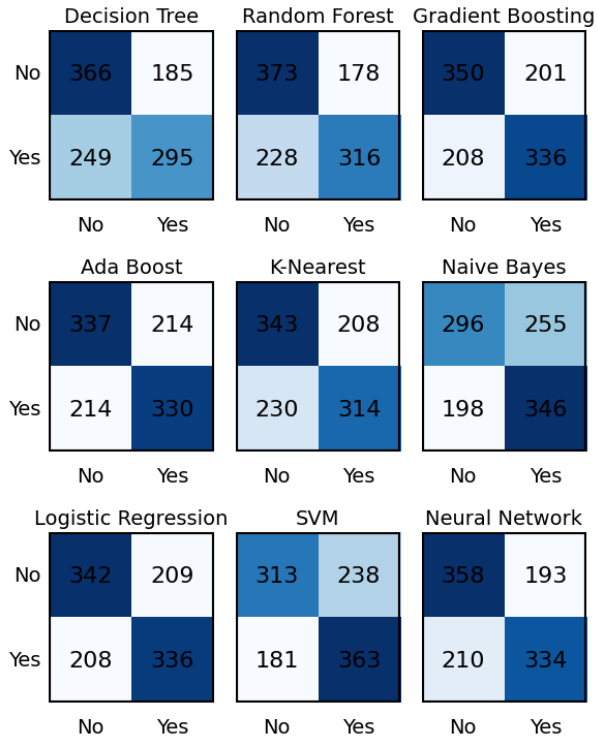
Priorizar o recall se torna importante se a empresa quer evitar a perda de talentos, por exemplo. Ter um recall baixo significa que muitos funcionários que realmente saíram não foram identificados pelo modelo, o que é crítico em setores ou funções onde a perda de talentos é muito custosa para a organização, como em casos altamente especializados.

7 DESEMPENHO DOS ALGORITMOS

A matriz de confusão na figura 10 mostra a proporção de previsões para cada classe, ajudando a avaliar o modelo, a

precisão indica a acurácia nas previsões, enquanto o recall mostra a sensibilidade do modelo em identificar casos positivos. A matriz é essencial para entender o comportamento do modelo em relação a classes específicas e ajustar seu desempenho, se necessário.

Figure 10: Matriz de Confusão



Nela podemos perceber que existe uma tendência de acerto ligeiramente maior nos modelos baseados em arvore em acertar os valores verdadeiros daqueles funcionários que não irão sair do trabalho. Por outro lado modelos mais baseados em formulas do que decisão tendem a acertar mais valores referentes a quem realmente ira sair do trabalho.

8 ANÁLISE COMPARATIVA DOS MODELOS ATRAVÉS DE TESTES T PAREADOS

A análise estatística dos modelos foi realizada através de testes T pareados, cujos resultados são apresentados na Tabela 6 de maneira que o nome dos modelos esta abreviado de maneira a manter a primeira letra de cada palavra do nome como por exemplo 'Decision Tree' se torna 'DT', seguindo a mesma ordem dos modelos apresentadas em outras tabelas e gráficos no presente artigo como a Tabela 5. Além disso os valores são apresentados omitindo a existência de valor zero a esquerda da virgula, focando exclusivamente nas casas decimais Valores p inferiores a 0,05 indicam diferenças estatisticamente significativas entre os desempenhos dos modelos

Table 6: Valores p dos testes T pareados entre os modelos

	DT	RF	GB	AB	KN	NB	LR	SV	NN
DT		.06	.40	.27	.67	.86	.24	.25	.23
RF	.06		.24	.22	.18	.21	.45	.30	.52
GB	.40	.24		.91	.76	.69	.68	.80	.63
AB	.27	.22	.91		.66	.62	.72	.86	.67
KN	.67	.18	.76	.66		.89	.51	.59	.48
NB	.86	.21	.69	.62	.89		.49	.56	.47
LR	.24	.45	.68	.72	.51	.49		.84	.93
SV	.25	.30	.80	.86	.59	.56	.84		.78
NN	.23	.52	.63	.67	.48	.47	.93	.78	

comparados. Por exemplo, a comparação entre Decision Tree (DT) e Random Forest (RF) apresentou um valor p de 0,061, sugerindo uma diferença marginalmente significativa. Em contraste, valores p elevados, como 0,855 entre DT e Naive Bayes (NB), indicam desempenhos estatisticamente similares entre os modelos.

Destaca-se a similaridade entre Gradient Boosting (GB) e Ada Boost (AB), com um valor p extremamente alto de 0,907, indicando desempenhos quase idênticos. Da mesma forma, Regressão Logística (LR) e Neural Network (NN) mostraram desempenhos comparáveis, com um valor p de 0,932. Esses resultados demonstram a importância de escolher modelos não apenas com base em métricas de desempenho, mas também considerando a natureza dos dados e o contexto de aplicação, visto que modelos diferentes podem oferecer resultados semelhantes em determinadas situações.

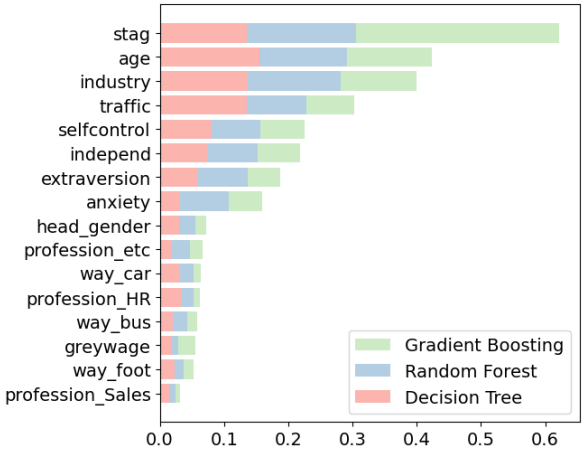
9 INTERPRETABILIDADE DOS MODELOS GERADOS

Modelos baseados em árvore, como Random Forest e Gradient Boosting, são conhecidos não só pela capacidade preditiva, mas também por sua habilidade interpretativa. Eles permitem quantificar a importância das classes, onde aqueles frequentemente localizados nos nós raiz têm maior relevância. Para uma compreensão mais profunda a figura 11 apresenta os dados retirados dos modelos baseados em arvore treinados ordenados pela sua relevância na classificação.

Com base nas importâncias das características, podemos fazer as seguintes conclusões:

- O tempo de permanência do funcionário é o fator mais importante na determinação se ele permanecerá na empresa ou não.
- A indústria em que o funcionário trabalha também desempenha um papel significativo, indicando que o clima e a cultura empresarial são fortes influenciadores da retenção de funcionários, levando a necessidade de

Figure 11: Importância dos atributos



trabalhos futuros dedicados exclusivamente a analise desses setores empresariais e suas correlações.

- Características pessoais, como autocontrole e ansiedade também são importantes, provavelmente relacionadas à resiliência do indivíduo.
- O método de contratação pode indicar a existência de fatores culturais específicos associados à plataforma de recrutamento.
- A idade é um fator crucial na determinação da resiliência de um funcionário em uma vaga.

Essas descobertas podem ser valiosas para a tomada de decisões e estratégias de retenção de funcionários, permitindo que as empresas ajam de maneira mais eficaz na gestão de seu pessoal.

10 CÓDIGO

O código para o presente artigo pode ser encontrado em um repositório Github a partir do endereço https://github.com/ToniLommez/EmployeeTurnover_IA [5]

11 TRABALHOS FUTUROS

Apesar dos avanços e insights gerados neste estudo, há várias direções a serem exploradas em trabalhos futuros:

Integração com Outros Dados

Incorporação de outras fontes de dados, como avaliações de desempenho dos funcionários ou feedbacks de satisfação, para enriquecer a análise e melhorar a precisão dos modelos.

Análise temporal

Pode-se realizar a análise de tempo de vida baseado em séries temporais para determinar de quando um funcionário ira sair, além de fornecer mais informação sobre as características mais relevantes para essa decisão.

REFERENCES

[1] G. Anwar and I. Shukur. 2015. The Impact of Training and Development on Job Satisfaction: A Case Study of Private Banks in Erbil. *International Journal of Social Sciences & Educational Studies* 2, 1 (2015), 65.

[2] K. Anwar. 2017. Analyzing the conceptual model of service quality and its relationship with guests'satisfaction: a study of hotels in erbil. *The International Journal of Accounting and Business Society* 25, 2 (2017), 1–16.

[3] Edward Babushkin. 2017. *Employee Turnover.* https://www.kaggle.com/datasets/davinwijaya/employee-turnover Base de dados disponível em Kaggle.

[4] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2022. *An introduction to statistical learning: With applications in R.* Springer.

[5] Marcos A. Lommez C. R. 2023. EmployeeTurnover_IA. github.com/ToniLommez/EmployeeTurnover_IA Codigo Jupyter de treinamento do modelo.

[6] Arianne Renan Barzilay. 2019. Data Analytics at Work: A View From Israel on Employee Privacy and Equality in the Age of Data-Driven Employment Management. *COMP. LABOR LAW & POL'Y JOURNAL* 40 (June 2019), 421. Available at SSRN: https://ssrn.com/abstract=3426614 or http://dx.doi.org/10.2139/ssrn.3426614.