Aplicação de Modelos de Classificação no Cenário Empregatício de Ciência de Dados

Iury Lima Rosal
Engenharia de Computação
Universidade Federal do Ceará (UFC)
Fortaleza, Brasil
iuryrosal@gmail.com

Vinícius Almeida de Castro
Engenharia de Computação
Universidade Federal do Ceará (UFC)
Fortaleza, Brasil
viniciusAC@alu.ufc.br

Luis Gustavo de Castro Sousa
Engenharia de Computação
Universidade Federal do Ceará (UFC)
Fortaleza, Brasil
gustavo.castro97@alu.ufc.br

***Resumo*—Neste artigo, realizamos a construção de uma modelagem de classificação usando modelos lineares e não-lineares para realizar a previsão do desejo ou não de mudança de emprego de funcionários da área de ciência de dados. No projeto, lidamos com situações de modelar dados categóricos e de desbalancemento de classes da variável target. A intenção é, além de fazer a modelagem preditiva, avaliar o impacto da modelagem, a partir do uso de modelos lineares e não-lineares.**

***Index Terms*—modelo de classificação, modelo preditivo, modelo linear de classificação, modelo não-linear de classificação**

I. Introdução

A modelagem preditiva é extremamente importante para a previsão de acontecimentos e estudo das relações que podem ocasionar nesses acontecimentos. O que torna ela uma ferramenta extremamente eficiente para tomadas de decisão e melhoria da qualidade de vida.

Modelos preditivos de classificação são modelos de aprendizagem de máquina supervisionados que realizam a tentativa de prever qual a classe que um determinado registro pertence em cima de um determinado conjunto de dados. Essa modelagem é muito comum na detecção de fraude, classificação de SPAM em e-mail, entre outros.

A escolha da base de dados se resume a tentar prever se um determinado cientista de dados pretende ou não mudar de emprego, sendo uma aplicação dentro da área de RH, por exemplo. Esse projeto ajuda na tomada de decisão dentro de entidades, que possuem como um de seus setores a área de ciência de dados, podendo prever rotatividade de funcionários.

II. Sobre a base de dados

A base de dados foi obtida via a plataforma do Kaggle [1]. Ela contém informações a respeito de funcionários da área de ciência de dados, tendo informações como gênero, nível de educação, entre outras informações. Detalhes a respeito do significado de cada variável (metadados) se encontram no próprio kaggle. [2] e na tabela I.

Métodos

Nas sessões a seguir (III-VI) serão mostrados, de forma mais detalhada, os métodos e procedimentos realizados na modelagem de classificação, juntamente com as visualizações geradas nesse processo.

Tabela I
Descrição das variáveis do problema

Variável	Descrição
enrollee_id	Identificador único do candidato
city	Código da cidade
city_development_index	Índice de desenvolvimento da cidade
gender	Gênero
relevent_experience	Experiência relevante do candidato
enrolled_university	Tipo de curso universitário matriculado, se houver
education_level	Nível de educação do candidato
major_discipline	Disciplina principal de educação do candidato
experience	Experiência total do candidato em anos
company_size	Nº de funcionários na empresa do atual empregador
company_type	Tipo de empresa do atual empregador
lastnewjob	Diferença, em anos, entre o atual trabalho e o anterior
training_hours	Horas completas de treinamento
target	Se deseja (1) ou não (0) mudar de emprego

III. Análise Exploratória dos dados

Antes de realizar o procedimento de modelagem é importante fazer um estudo exploratório das variáveis e suas relações em relação a variável target, que nesse caso seria estar procurando um novo emprego (1) ou não estar procurando um novo emprego (0).

Inicialmente, investigamos as variáveis numéricas, nesse caso, city_development_index, training_hours e a target. Na figura 1 se encontra a visualização em histogramas e na tabela II se encontra os valores das variáveis estatísticas.

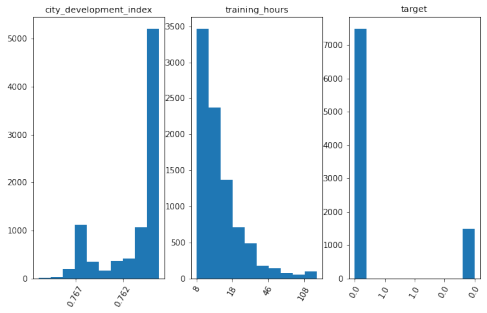

Figura 1. Histograma envolvendo apenas features numéricas

Tabela II
VALORES DAS VARIÁVEIS ESTATÍSTICAS PARA CADA FEATURE NUMÉRICA

Variável	média	desvio padrão	skewness
city_development_index	0.844570	0.116178	-1.269173
training_hours	65.074930	60.235087	1.849579
target	0.165606	0.371747	1.799443

Para investigar as variáveis categóricas, verificamos a frequência de cada categoria dentro de cada variável categórica por meio de um gráfico de barras, onde no eixo X temos as classes da variável categórica e o eixo Y exibe a frequência que a classe foi utilizada dentro da base de dados. A figura 2 exibe o resultado dessa visualização.

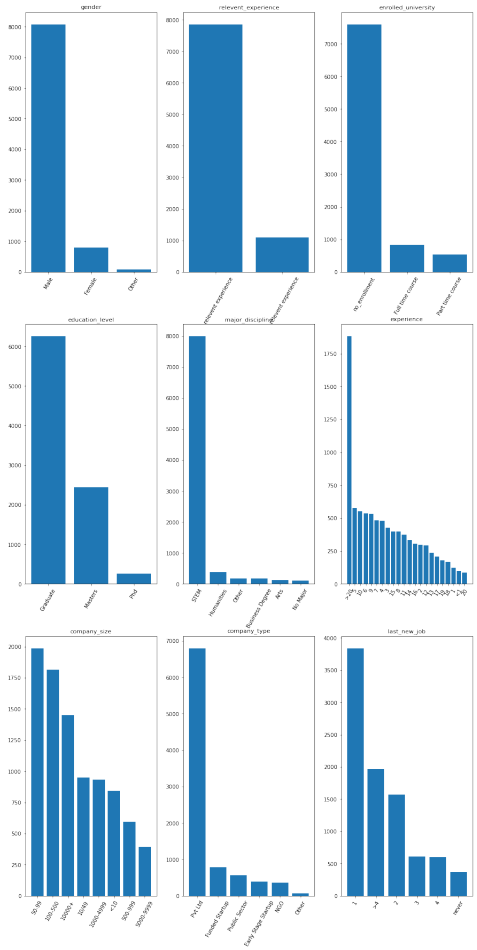

Figura 2. Gráficos de barras envolvendo features categóricas

Conclusões dessas análises, estão esboçadas na área de resultados na seção VII.

IV. PRÉ-PROCESSAMENTO DOS DADOS

Para aplicação na modelagem preditiva é necessário a realização de alguns procedimentos de pré-processamento de dados para garantirmos uma melhor performance do modelo, assim como torná-lo mais viável. Primeiramente, removemos as colunas 'enrollee_id' e 'city', pois a primeira remete a uma identificação, o que não possui nenhuma relação que impacte na decisão do funcionário, e a segunda coluna removida possui um grande número de classes, o que pode atrapalhar na predição pela grande variedade de classes.

Após a remoção dessas duas colunas, aplicaremos uma técnica de conversão de dados categóricos em dados numéricos, denominada One Hot Enconding [3], para possibilitar o uso deles na modelagem. Inicialmente, nesta técnica aplicamos a lógica do Label Enconding, onde as classes são substituídas por números. Por exemplo, 'Classe A' virará 0, 'Classe B' virará 1 e, assim por diante. Esse processo será aplicado em todas as features categóricas. Após isso, devemos aplicar o One Hot Enconding, pois deixar as classes apenas representadas por números pode fazer com que o modelo interprete que uma determinada classe seja mais importante ou de maior valor que as outras e isso pode enviesar a sua predição. Por exemplo, uma 'Classe A' recebe valor 0 no Label Enconder, enquanto uma 'Classe I' recebe valor 7. A modelagem pode interpretar a 'Classe I' como superior à 'Classe A', pois $7 > 0$. Logo, devemos aplicar o One Hot Enconding.

O One Hot Enconding funciona da seguinte forma: suponha que temos 1 feature categórica com m classes. Nesse processo, essa coluna será substituída por m colunas de valor binário, ou seja, colunas preenchidas por apenas valor 0 e 1. Para afirmar que um determinado registro pertence a uma classe, o valor 1 deve estar na coluna que representa essa classe e 0 nas demais classes. Essa lógica se aplica para todos os registros. Ou seja, agora em vez de termos uma única coluna com a informação da classe, temos um conjunto de colunas que a interseção delas dirá qual a classe o determinado registro pertence.

Tendo as variáveis categóricas como numéricas, podemos realizar uma análise para verificar se priorizamos alguma variável. Para isso, realizamos uma análise bivariada, utilizando a matriz de correlação. Essa matriz contém as correlações de Pearson [8], que é uma associação estatística que se refere a quão próximas duas variáveis estão de ter uma relação linear entre si. O coeficiente de correlação de Pearson (r) pode ser descrito na fórmula matemática abaixo:

$$r = \frac{\sum(x - m_x)(y - m_y)}{\sqrt{\sum(x - m_x)^2(y - m_y)^2}}$$

O valor desse coeficiente varia entre -1 e 1. Uma correlação pode ser positiva ($r > 0$), o que significa que ambas as variáveis se movem na mesma direção, ou são negativas ($r < 0$), o que significa que se movem em direções opostas, ou seja, quando o valor de uma variável aumenta, o valor da outra variável diminui. A correlação também pode ser nula ou zero ($r = 0$), o que significa que as variáveis não estão relacionadas. A figura 3 exibe o resultado dessa visualização.

Observando a coluna que envolve a variável target, a correlação com as outras variáveis é bem baixa, variando entre -0.1 e 0.2, para a maioria das variáveis. Portanto, decidimos utilizar todas as variáveis como preditoras, já que não existem

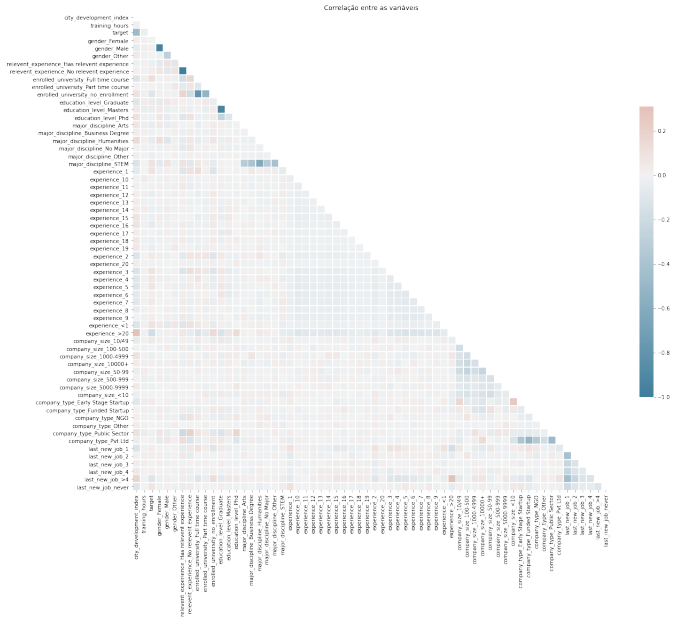

Figura 3. Matriz de correlação em mapa de calor (heatmap)

variáveis com um alto valor de correlação com a variável target, tanto num sentido positivo como negativo.

Após converter todos os dados categóricos em dados numéricos, realizamos o processo de Undersampling [4], para balancear as classes da variável target. Para o balanceamento, reduzimos a frequência de dados da classe 0 até ficar com a mesma frequência dos dados da classe 1. Esse balanceamento é importante para evitar enviesamento na predição, ou seja, o modelo teria dificuldade de diferenciar as classes da variável target caso as classes estejam desbalanceadas. Nas figuras 4 e 5 mostram as frequências das classes, na variável target, antes e depois desse processo, respectivamente.

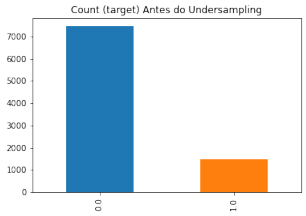

Figura 4. Desbalanceamento da classe target

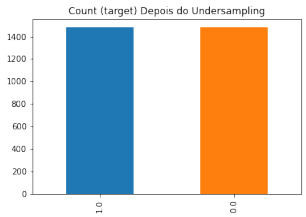

Figura 5. Balanceamento das classes feita via procedimento de undersampling

Por fim, fazemos a separação dos preditores e da variável target. Após isso, aplicamos uma padronização dos dados preditores para garantir que todos estejam no mesmo intervalo, evitando que o modelo dê um maior peso em determinados preditores em comparação aos demais. Esse processo é feito pela fórmula abaixo:

$$\boxed{\frac{X - M}{S}}$$

Na fórmula, M seria a média dos registros dos preditores e S é o desvio padrão das amostras de treinamento.

Após esse pré-processamento, separamos os dados para o treinamento dos modelos e para o teste, isso tanto nas variáveis preditoras como na variável target. A separação foi de 70% dos dados para treino e 30% para teste. Essa técnica consiste em selecionar amostras do conjunto de preditores e target, de forma que possamos treinar o modelo com uma grande quantidade de dados (possibilidades) e, ao mesmo tempo, validá-lo com um conjunto de dados diferente do que foi usado para seu treino. Além disso, essa quebra foi feita de uma forma que, dentro do treino e teste, a quantidade de dados target da classe 0 e 1 estejam equilibrados.

V. Modelo de Classificação Linear

Para a construção da modelagem preditiva de classificação selecionamos o modelo linear LDA (Linear Discriminant Analysis) [6]. Esse algoritmo busca aumentar a separabilidade entre as classes (a diferença entre a média das classes) e calcular a distância entre a média e uma amostra para cada classe, reduzindo-a. Isso permite uma melhor separação entre as classes, facilitando o procedimento de classificação.

Após o treino e a predição, comparamos o resultado previsto com o target real. Fizemos isso via matriz de confusão. Nas figuras 6 e 7 estão presentes as matrizes de confusão, uma com valores absolutos e outra em formato de porcentagem, respectivamente.

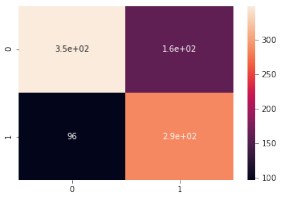

Figura 6. Matriz de confusão da predição com dados de teste da modelagem com LDA

A matriz de confusão mostra em relação à como o modelo está prevendo corretamente as classes. Na matriz as células na diagonal principal representam as predições corretas, enquanto as demais representam as predições incorretas. As linhas da matriz são as classes verdadeiras e as colunas são as predições. Pela imagem vemos que 39.21% das predições foram verdadeiros negativos (VN), ou seja, o modelo previu corretamente em relação à classe 0, e 32.13% foram verdadeiros positivos

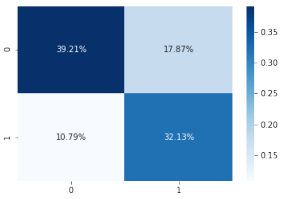

Figura 7. Matriz de confusão da predição com dados de teste da modelagem com LDA (em %)

(VP), ou seja, o modelo previu corretamente em relação à classe 1.

Para avaliarmos melhor a performance do modelo verificamos 4 métricas [5] que serão descritas a seguir: - Acurácia: essa métrica verifica em relação a quantidade de acertos em relação ao todo, sem considerar uma separação entre as classes.

$$\boxed{Acurácia = \frac{VP + VN}{VP + VN + FP + FN}}$$

- Precisão: verifica a porcentagem de acerto em relação ao verdadeiro positivo (VP), no nosso caso, a porcentagem de acerto em prever a classe 1.

$$\boxed{Precisão = \frac{VP}{VP + FP}}$$

- Recall (sensibilidade): mostra quanto um modelo consegue reconhecer de uma determinada classe.

$$\boxed{Recall = \frac{VP}{VP + FN}}$$

- F1: faz a média harmônica entre a precisão e a recall, auxiliando na performance do modelo. Essa métrica será principalmente utilizada nas comparações de modelo e no model tuning.

$$\boxed{F1 = 2 \times \frac{Precisão \times Recall}{Precisão + Recall}}$$

A tabela III mostra os resultados destas métricas.

Tabela III
MÉTRICAS DE PERFORMANCE

Métrica	Score
Acurácia	0.7135
Precisão	0.7487
Recall	0.6427
F1	0.6917

VI. MODELO DE CLASSIFICAÇÃO NÃO LINEAR

Para o modelo de classificação não-linear, utilizaremos o modelo KNN (K-Neighbors Classifier). [7] O KNN trabalha com a ideia de classificação buscando k vizinhos próximos a um determinado ponto A, verificando os k pontos com as menores distâncias euclidianas. A partir daí, é verificado a classe majoritária desses pontos vizinhos a ele e assim é dada essa classe majoritária como a resposta da classificação desse ponto A.

Para usarmos o melhor parâmetro do modelo, nesse caso, o número de vizinhos, utilizaremos a técnica de model tuning com validação cruzada. Separamos a base de dados em 10 grupos para a validação cruzada, em que testaremos diferentes combinações de treino e teste. Junto a isso, variamos o parâmetro do KNN de 2 à 20 vizinhanças e coletamos o F1 score para cada parâmetro e, assim, poderemos verificar o modelo com a melhor performance.

A figura 9 exibe o gráfico dos F1 scores pelos seus números de vizinhança respectivos na modelagem.

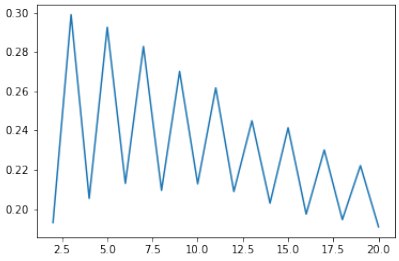

Figura 8. Gráfico do número de vizinhanças, em relação ao F1 score

Pela imagem, verificamos que o melhor valor para esse parâmetro foi o 3, que obteve a maior média de F1 Score na validação cruzada comparado aos demais valores. Logo, construímos o modelo com 3 vizinhos. Ou seja, no processo de classificar um registro, o modelo buscará os 3 dados mais próximos desse registro, via distância euclediana, para classificar qual a classe esse registro pertence.

Após o treino e a predição, comparamos o resultado previsto com o target real. Fizemos isso via matriz de confusão. Nas figuras 10 e 11 estão presentes as matrizes de confusão, uma com valores absolutos e outra em formato de porcentagem.

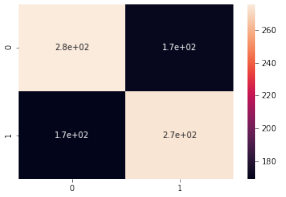

Figura 9. Matriz de confusão da predição com dados de teste da modelagem com KNN

Pela matriz de confusão vemos que 31.01% das predições foram verdadeiros negativos, ou seja, o modelo previu corretamente em relação à classe 0, e 30.79% foram verdadeiros positivos, ou seja, o modelo previu corretamente em relação à classe 1.

A tabela IV mostra os resultados das métricas de acurácia, precisão, recall e F1.

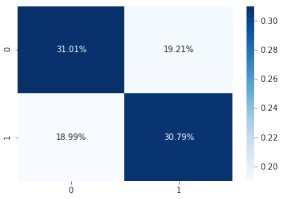

Figura 10. Matriz de confusão da predição com dados de teste da modelagem com KNN (em %)

Tabela IV
MÉTRICAS DE PERFORMANCE

Métrica	Score
Acurácia	0.618
Precisão	0.6185
Recall	0.6157
F1	0.6171

RESULTADOS

Nas seções a seguir, comentamos algumas conclusões referentes ao métodos anteriormente realizados e comentados.

VII. ANÁLISE EXPLORATÓRIA DE DADOS

Pela análise exploratória, verificamos que boa parte dos registros pertenciam a classe 0, ou seja, não estavam procurando um novo emprego. Essa análise indicou o desbalanceamento de classes na variável target. Além disso, verificamos uma grande concentração de pessoas com poucas horas de treinamento (abaixo de 100 horas). As cidades em que as pessoas trabalhavam, a grande maioria, possuía altos índices de desenvolvimento.

Pelos gráficos de barra, destacamos que boa parte das pessoas possuem graduação completa, poucos tendo mestrado ou PHD. Ressaltamos que a maioria possuíam menos de 20 anos de experiência, trabalhando em empresas da iniciativa privada com uma faixa de 50-500 funcionários.

VIII. PRÉ-PROCESSAMENTO

A aplicação do one hot enconding permitiu a utilização das features categóricas dentro da modelagem, auxiliando o modelo no processo de treinamento e predição, já que não existiam features com uma alta correlação com a variável target.

O procedimento de balanceamento das classes (undersampling) foi importante, pois conseguimos manter um melhor equilíbrio entre a assertividade dentro das classes, ou seja, a quantidade que foi acertado da classe 0 é proporcionalmente similar a quantidade que foi acertado da classe 1. Podemos afirmar isso observamos as matrizes de confusão e também pela métrica recall. Para o modelo de LDA, cerca de 72% das predições foram assertivas (em que 39,21% foi em relação a classe 0 e 32,13% foi em relação a classe 1), com um recall de 64,3%. Resumindo, o modelo se mostrou eficaz em diferenciar as classes e classificar exclusivamente o registro dentro de cada classe.

IX. COMPARATIVO ENTRE A MODELAGEM LINEAR E NÃO LINEAR

A partir dos métodos descritos, observamos que a modelagem utilizando LDA se mostrou mais eficiente do que a modelagem utilizando KNN, mesmo escolhendo o melhor parâmetro (número de vizinhos) via validação cruzada. Podemos afirmar isso comparando a métrica F1, que é muito eficiente na avaliação da performance do modelo, em que no caso do LDA foi cerca de 0,69 (aproximadamente), enquanto no KNN foi em torno de 0,61 (aproximadamente). Além disso, nas outras métricas, no caso, acurácia, precisão e recall o modelo do LDA se mostrou mais assertivo na classificação dos registros. A tabela 4 mostra um comparativo das métricas entre as duas modelagens.

Tabela V
MÉTRICAS DE PERFORMANCE

Métrica	Score LDA	Score KNN
Acurácia	0.7135	0.618
Precisão	0.7487	0.6185
Recall	0.6427	0.6157
F1	0.6917	0.6171

O fato do modelo LDA ter se saído melhor que o KNN pode ter sido pela situação se enquadrar, já de uma maneira interessante, em um contexto linear, já que o KNN é um modelo não-linear. Podemos citar também o LDA como um modelo um pouco mais complexo que o KNN, o que pode ter, também, colaborado para uma melhor flexibilidade, comparando os dois modelos.

REFERÊNCIAS

[1] Kaggle. Disponível em: https://www.kaggle.com/. Acesso em: 20 mar. 2021.
[2] HR Analytics: Job Change of Data Scientists. Disponível em: https://www.kaggle.com/arashnic/hr-analytics-job-change-of-data-scientists. Acesso em: 19 mar. 2021.
[3] Engenharia de Features: Transformando dados categóricos em dados numéricos. Disponível em: https://bit.ly/3cUtrPs. Acesso em: 20 mar. 2021.
[4] Como lidar com dados desbalanceados em problemas de classificação. Disponível em: https://medium.com/data-hackers/como-lidar-com-dados-desbalanceados-em-problemas-de-classifição-17c4d4357ef9. Acesso em: 19 mar. 2021.
[5] Indo Além da Acurácia: Entendo a Acurácia Balanceada, Precisão, Recall e F1 score. Disponível em: https://bit.ly/315ClnC. Acesso em: 19 mar. 2021.
[6] LINEAR Discriminant Analysis. In: JAMES, Gareth; WITTEN, Daniela; HASTIE, Trevor; TIBSHIRANI, Robert. An Introduction to Statistical Learning: with Applications in R. [S. l.: s. n.], 2013. cap. 4.4.
[7] Assessing Model Accuracy - K-Nearest Neighbors. In: JAMES, Gareth; WITTEN, Daniela; HASTIE, Trevor; TIBSHIRANI, Robert. An Introduction to Statistical Learning: with Applications in R. [S. l.: s. n.], 2013. cap. 2.2.
[8] Sobre Correlações e visualizações de matrizes de correlação no R. Disponível em: https://bit.ly/3vKoQIn. Acesso em: 21 mar. 2021.