Challenge Sprint 3 – 2024

Front end & Mobile development

Prof Mario Andre de Deus

Camilly Alves RM 550210

João Vitor Martins RM 98744

Murilo Krauss RM 98262

Lucas Matheus da Silva RM 550466

Introdução

Neste relatório, realizamos a modelagem para prever a rotatividade de funcionários (Attrition) usando dois modelos principais: RandomForestClassifier e LogisticRegression. A abordagem incluiu o pré-processamento dos dados, ajuste de hiperparâmetros (tuning) com GridSearchCV, validação cruzada e análise comparativa das métricas de desempenho, como Acurácia e F1-Score.

Descrição dos Dados

Os dados utilizados nesta análise consistem em um conjunto de informações de funcionários, contendo variáveis como idade, salário, departamento, e outras características, além da variável-alvo Attrition, que indica se o funcionário deixou a empresa ou não.

Feature Engineering

O processo de Feature Engineering é uma etapa fundamental para a criação de variáveis que aumentam o poder preditivo dos modelos. Para essa tarefa, foram seguidas as seguintes etapas principais:

1.1. Tratamento de Valores Ausentes

- As variáveis numéricas tiveram seus valores ausentes imputados com a mediana.
- As variáveis categóricas, por sua vez, tiveram os valores ausentes preenchidos com o valor mais frequente (moda).

1.2. Codificação de Variáveis Categóricas

- As variáveis categóricas foram codificadas usando técnicas apropriadas:
    - Label Encoding foi aplicado na variável-alvo Attrition para convertê-la em uma variável binária (0 = Não, 1 = Sim).
    - One-Hot Encoding foi utilizado para outras variáveis categóricas, permitindo que o modelo interpretasse essas categorias corretamente.

1.3. Escalonamento de Variáveis Numéricas

- Para garantir que os modelos sensíveis à escala pudessem lidar bem com os dados, as variáveis numéricas foram padronizadas utilizando o StandardScaler. Isso garante que todas as variáveis numéricas tenham média zero e desvio padrão um.

Treinamento de Modelos

Foram treinados dois modelos principais: RandomForestClassifier e LogisticRegression. Ambos os modelos foram treinados inicialmente com seus hiperparâmetros padrão para obter uma base de comparação.

2.1. RandomForestClassifier

O RandomForestClassifier é um modelo baseado em múltiplas árvores de decisão. Ele foi treinado com o objetivo de identificar padrões não lineares complexos entre as variáveis preditivas e a variável-alvo Attrition.

2.2. LogisticRegression

A Regressão Logística foi treinada para modelar a probabilidade de um funcionário sair ou não da empresa com base nas variáveis preditivas. Este é um modelo linear que se mostrou eficiente, especialmente em problemas de classificação binária como este.

Validação de Modelos

A validação foi realizada usando validação cruzada com 5 folds, permitindo uma avaliação mais robusta dos modelos. A validação cruzada garante que o desempenho dos modelos seja avaliado em diferentes subconjuntos dos dados de treino, ajudando a evitar overfitting.

3.1. Acurácia

- A acurácia média com validação cruzada foi calculada para ambos os modelos. A acurácia mede a proporção de predições corretas entre todas as predições realizadas.
- RandomForestClassifier obteve uma acurácia média de 0.84.
- LogisticRegression obteve uma acurácia média de 0.84.

3.2. F1-Score

- O F1-Score, uma métrica importante para problemas de classificação desbalanceada, também foi avaliado.
- RandomForestClassifier apresentou um F1-Score médio de 0.18.
- LogisticRegression obteve um F1-Score médio de 0.42.

3.3. Curva ROC

- RandomForestClassifier apresentou um F1-Score médio de 0.76.
- LogisticRegression obteve um F1-Score médio de 0.81.

Tuning de Hiperparâmetros

Para melhorar o desempenho dos modelos, foi realizado um ajuste de hiperparâmetros usando GridSearchCV. Esse processo permite testar várias combinações de hiperparâmetros e selecionar aquela que maximiza o desempenho do modelo.

RandomForestClassifier

Os hiperparâmetros ajustados no RandomForestClassifier foram:

- n_estimators: Número de árvores no modelo, com valores testados de [100, 200, 300].
- max_depth: Profundidade máxima das árvores, com valores testados de [10, 20, 30].
- min_samples_split: Número mínimo de amostras para realizar uma divisão em um nó, com valores testados de [2, 5, 10].

Após o ajuste, os melhores hiperparâmetros encontrados foram:

n_estimators = 200

max_depth = 20

min_samples_split = 2

O modelo com esses hiperparâmetros apresentou uma melhora em termos de acurácia e F1-Score, embora ainda tenha ficado atrás da LogisticRegression.

LogisticRegression

Os hiperparâmetros ajustados no LogisticRegression foram:

- C: Parâmetro de regularização, testado com os valores [0.01, 0.1, 1, 10, 100].
- penalty: Tipo de penalidade, testado com valores de ['l1', 'l2'].
- solver: Método de otimização, testado com os valores ['liblinear', 'saga'].

Os melhores hiperparâmetros encontrados foram:

C = 1

penalty = 'l2'

solver = 'liblinear'

Apesar do ajuste, o desempenho da LogisticRegression não teve grandes ganhos, sugerindo que o modelo já estava bem ajustado mesmo sem tuning.

Conclusão

Após o ajuste de hiperparâmetros e a validação cruzada, a LogisticRegression demonstrou melhor desempenho geral em comparação com o RandomForestClassifier, tanto em termos de acurácia quanto de F1-Score. Isso torna o modelo de Regressão Logística a escolha mais adequada para a predição de rotatividade de funcionários nesta análise.