Evaluation and Deployment:

Bank Marketing

Discentes:

* Arthur Bezerra Calado
* Gabriel D'assumpção de Carvalho
* Pedro Henrique Sarmento de Paula

Data: 05/08/2024

Introdução

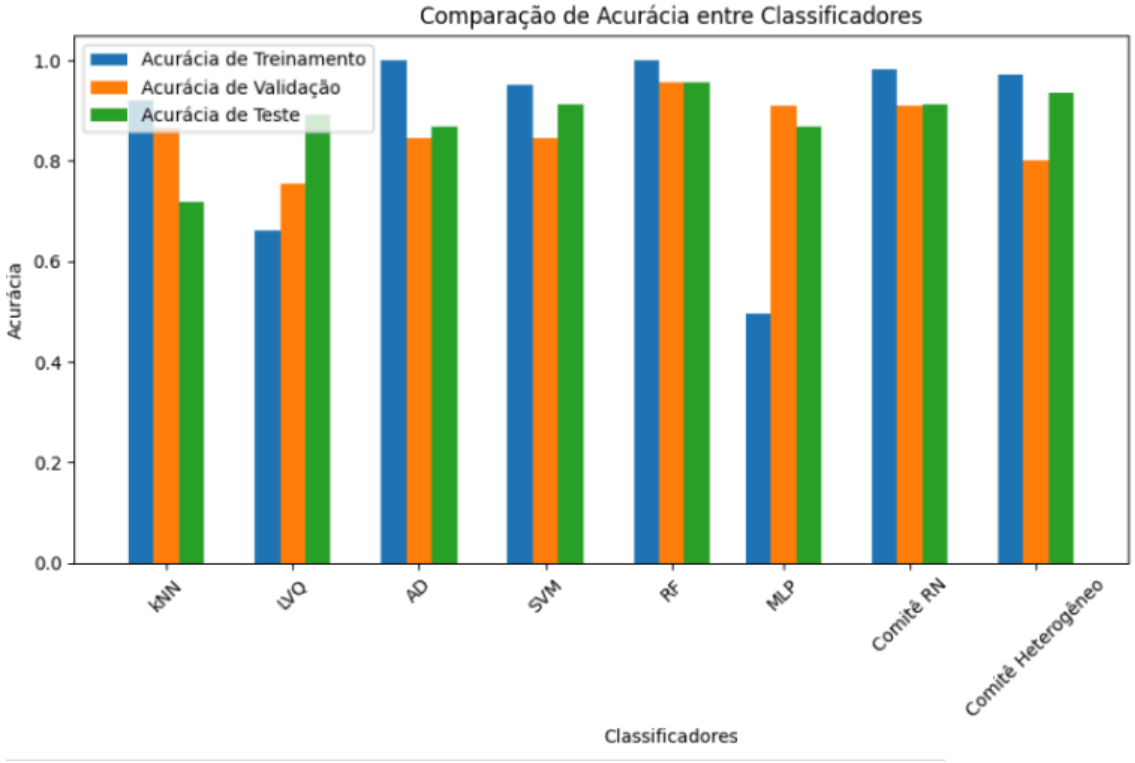
As fases de Avaliação e Implantação são essenciais para validar e aplicar os modelos desenvolvidos em um projeto de ciência de dados. Na Avaliação, os modelos são testados quanto à sua eficácia, assegurando que atendem aos objetivos definidos. Já na Implantação, os modelos são integrados ao ambiente de produção para gerar valor real ao negócio.

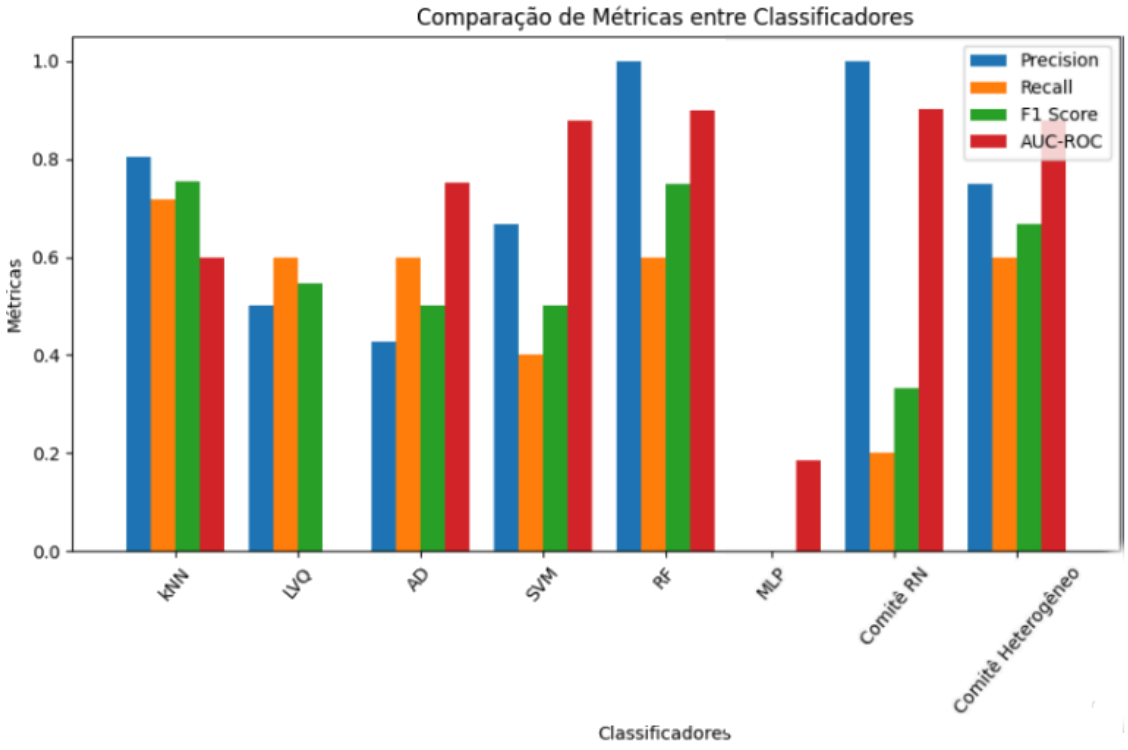
Essas fases estão diretamente conectadas às etapas anteriores do projeto: Entendimento do Negócio, Entendimento dos Dados, Preparação dos Dados e Modelagem. O sucesso da Avaliação e Implantação depende da sólida base construída nessas fases, garantindo que as soluções sejam tanto válidas quanto aplicáveis.

Avaliação dos Modelos

A avaliação dos modelos considerou a seleção dos parâmetros ótimos para cada algoritmo, com base no desempenho em termos de acurácia no conjunto de validação. Após identificar os melhores parâmetros, foram calculadas as métricas de precisão, recall, F1-score e AUC-ROC para obter uma visão detalhada da performance do modelo, especialmente em relação à sua capacidade de identificar corretamente os indivíduos que aceitariam fazer um depósito a prazo.

Os gráficos a seguir ilustram os resultados dessa análise:





Os gráficos de comparação das métricas dos modelos proporcionam uma visão abrangente do desempenho de cada um. Com essa análise, fica claro que os modelos KNN, Random Forest e Comitê Heterogêneo se destacam com as melhores métricas. Ter essa perspectiva é fundamental para identificar os modelos mais promissores, permitindo que possamos focar nos três melhores e ajustá-los continuamente ao longo do tempo, garantindo que permaneçam alinhados com os objetivos do banco.

Seleção do Modelo Final

A escolha do modelo final não será determinada apenas pela acurácia nos dados de treinamento e teste, mas também por uma análise abrangente de várias métricas de desempenho. A **precisão** é um critério essencial, pois avalia a capacidade do modelo de identificar corretamente os clientes que efetivamente realizam depósitos a prazo. Este aspecto é fundamental, dado que o objetivo principal do projeto é identificar esses clientes com alta precisão.

Além disso, o **recall** é uma métrica importante, pois mede a proporção de clientes que realmente fazem depósitos a prazo e que o modelo consegue identificar corretamente. O recall reflete a habilidade do modelo em captar todos os verdadeiros positivos, ou seja, os clientes que estão dispostos a realizar um depósito a prazo.

O **F1-score** será utilizado para equilibrar a precisão e o recall, oferecendo uma visão mais completa do desempenho do modelo. Esta métrica é particularmente útil quando se busca um equilíbrio entre identificar corretamente os clientes que fazem depósitos a prazo e minimizar a quantidade de falsos positivos e falsos negativos.

Finalmente, a métrica **AUC-ROC** será empregada para avaliar a capacidade do modelo de distinguir entre as classes positiva e negativa, refletindo a probabilidade de o modelo identificar corretamente a classe positiva ao longo de diferentes limiares de decisão.

Com base na análise dessas métricas, o modelo **Random Forest (RF)** com profundidade máxima de 124 e 206 estimadores se destacou como o melhor para o nosso objetivo. Este modelo apresentou uma precisão de 100%, indicando que não classificou nenhum cliente que não faria um depósito a prazo como propenso a fazê-lo. No entanto, devido ao desbalanceamento entre as classes, o recall foi de 60%. Apesar disso, o F1-score de 75% e o AUC-ROC de 90% evidenciam que o Random Forest oferece o melhor equilíbrio entre identificação precisa e abrangente dos clientes interessados em depósitos a prazo, alinhando-se mais efetivamente com os objetivos do projeto.

Implantação do Modelo

Bibliotecas

In [ ]:
```
!pip3 install scikit-learn==1.5.1
!pip3 install ucimlrepo
```

In [ ]:
```
import pandas as pd

# Configurando o modo de exibição do pandas
```

```
pd.options.display.float_format = "{:.4f}".format

import numpy as np

# Desativa todos os avisos
import warnings
warnings.filterwarnings("ignore")

from ucimlrepo import fetch_ucirepo

import joblib

import requests
```

O modelo de Random Forest com melhor desempenho foi salvo em um arquivo `.joblib`. Este arquivo permite carregar o modelo treinado em qualquer ambiente que tenha Python instalado, desde que as bibliotecas `scikit-learn` e `joblib` estejam nas versões 1.5.1 e 1.2, respectivamente.

In [ ]:
```
# URL do arquivo do modelo
url = 'https://github.com/gabrieldadcarvalho/machine_learning/raw/6c07cb6f9c

# Caminho local onde o arquivo será salvo
local_filename = 'rf_model.joblib'

# Baixar o arquivo
response = requests.get(url)
with open(local_filename, 'wb') as file:
    file.write(response.content)

# Carregar o modelo
best_rf = joblib.load(local_filename)

best_rf
```

Out[ ]:
```
RandomForestClassifier
RandomForestClassifier(max_depth=124, n_estimators=206, random_stat
e=10)
```

Após o carregamento do modelo, para utilizá-lo em futuras previsões, basta chamar `best_rf.predict(X)`, onde `X` é o vetor ou matriz contendo as características transformadas dos clientes que o banco está considerando para oferecer o depósito a prazo via telemarketing. As características dos clientes devem ser fornecidas no script abaixo devido às transformações propostas na etapa de preparação dos dados.

Coleta dos dados

A seguir, vamos coletar os dados necessários para demonstrar a utilização do modelo. Para isso, precisaremos apenas das características dos clientes.

In [ ]:
```
# Baixando os dados
bank_marketing = fetch_ucirepo(id=222)

# data (as pandas dataframes)
X = bank_marketing.data.features
```

Transformação dos Dados

Esta etapa é fundamental na implementação dos dados. O modelo final foi treinado e avaliado utilizando dados transformados, e, para implementações futuras, é essencial usar o script que receberá as características originais e aplicará as mesmas transformações propostas na terceira etapa do projeto.

In [ ]:
```
# Preenchendo valores nulos com valores padrão
X['job'] = X['job'].fillna('management')
X['education'] = X['education'].fillna('secondary')
X['contact'] = X['contact'].fillna('cellular')

# Removendo a coluna 'poutcome'
X.drop(columns=['poutcome'], inplace=True)

# Aplicando transformação e winsorizing na coluna 'age'
X['age'] = np.log(X['age'])**(1/2.15)
X['age'] = np.clip(X['age'], 1.6783890653308224, 1.9959212972209788)
X['age'] = (X['age'] - X['age'].min()) / (X['age'].max() - X['age'].min())

# Aplicando transformação e winsorizing na coluna 'balance'
X['balance'] = np.sqrt(X['balance'] * 2) * (1/7)
X['balance'] = np.clip(X['balance'], 1.3204692477561237, 3.596352079685467)
X['balance'] = (X['balance'] - X['balance'].min()) / (X['balance'].max() - X

# Aplicando transformação e winsorizing na coluna 'duration'
X['duration'] = X['duration']**(1/5)
X['duration'] = np.clip(X['duration'], 1.615394266202178, 4.02087473276376)
X['duration'] = (X['duration'] - X['duration'].min()) / (X['duration'].max()

# Convertendo variáveis categóricas em variáveis dummy
X = pd.get_dummies(X, columns=['job', 'marital', 'education', 'default', 'ho

for column in X.columns:
    if X[column].dtype == 'bool':
        X[column] = X[column].astype(int)
```

Classificação

Vamos selecionar 2000 observação aleatória do banco de dados para demonstrar o funcionamento do algoritmo de predição. O objetivo é avaliar se o cliente possui

características semelhantes às dos clientes que adquiriram um depósito a prazo como resultado da campanha de telemarketing.

In [ ]:
```python
# Inicializa um contador para clientes propensos
propensos_count = 0

print("Analisando 100 clientes aleatórios...")

for i in range(100):
    # Seleciona um índice aleatório
    random_index = np.random.randint(0, len(X))

    # Seleciona a linha correspondente do DataFrame
    single_row_df = X.iloc[[random_index]]

    # Faz a previsão
    prediction = best_rf.predict(single_row_df)

    # Verifica se a previsão é 1
    if prediction[0] == 1:
        print(f'O cliente com índice {random_index} é propenso a fazer um de
        propensos_count += 1

# Imprime o número total de clientes propensos encontrados
print(f'\nTotal de clientes propensos encontrados: {propensos_count} de 100
```

```
Analisando 100 clientes aleatórios...
O cliente com índice 10403 é propenso a fazer um depósito a prazo.
O cliente com índice 33312 é propenso a fazer um depósito a prazo.
O cliente com índice 29020 é propenso a fazer um depósito a prazo.
O cliente com índice 40618 é propenso a fazer um depósito a prazo.
O cliente com índice 40853 é propenso a fazer um depósito a prazo.
O cliente com índice 43102 é propenso a fazer um depósito a prazo.

Total de clientes propensos encontrados: 6 de 100 analisados.
```

Propostas de Decisões e Ações

Com base nos resultados obtidos ao longo do projeto, diversas decisões estratégicas e ações podem ser implementadas para otimizar o uso do modelo desenvolvido e maximizar a eficácia das campanhas de marketing. A primeira ação recomendada é a adoção do modelo como o principal mecanismo de previsão para identificar clientes com alta propensão a realizar depósitos a prazo. Este modelo foi escolhido por sua combinação de alta acurácia, robustez e capacidade de generalização, alcançando 95,65% de acurácia no conjunto de teste e exibindo excelente desempenho nas métricas de recall, F1-score e AUC-ROC, fundamentais para garantir previsões precisas e confiáveis.

A partir da integração deste modelo ao sistema do banco, será possível segmentar as campanhas de marketing de forma mais precisa, focando especificamente nos clientes classificados como "propensos" pelo modelo. Essa segmentação não apenas deve aumentar

a taxa de conversão das campanhas, mas também otimizará o uso dos recursos da equipe de marketing, permitindo uma abordagem mais eficiente e personalizada. Além disso, a automatização da aplicação das previsões do modelo nas listas de campanhas permitirá que as ações de marketing sejam rapidamente ajustadas com base nas mudanças dos dados e nos resultados obtidos.

É essencial que o modelo seja monitorado continuamente para assegurar que ele permaneça eficaz ao longo do tempo. Dado que os dados dos clientes e as condições de mercado são dinâmicos, ajustes periódicos no modelo podem ser necessários. A reavaliação dos hiperparâmetros, a incorporação de novas variáveis e a aplicação de técnicas de reamostragem devem ser consideradas como parte de uma estratégia de aprendizado contínuo, garantindo que o modelo se adapte às mudanças e continue a fornecer previsões precisas.

Reflexões Críticas

Durante o desenvolvimento deste projeto, alguns desafios importantes surgiram, oferecendo lições valiosas para futuros projetos. O principal desafio foi o desbalanceamento das classes, onde a maioria dos clientes não realizaram depósitos a prazo, enquanto apenas uma pequena fração o fez. Este desequilíbrio afetou significativamente as métricas de recall e F1-score, exigindo a aplicação de técnicas como o SMOTE para balancear as classes. Embora essa abordagem tenha ajudado a melhorar a capacidade do modelo de identificar clientes potenciais, ela também introduziu o risco de sobreajuste. Futuramente, pode ser benéfico explorar técnicas alternativas de balanceamento ou até mesmo algoritmos projetados especificamente para lidar com classes desbalanceadas.

Outro desafio foi a limitação de recursos computacionais, que restringiu a capacidade de trabalhar com todo o conjunto de dados disponível. A necessidade de usar uma amostra reduzida para o treinamento e validação dos modelos pode ter impactado a generalização dos resultados. Este desafio ressalta a importância de garantir que os recursos computacionais sejam adequados para a tarefa, especialmente em projetos que envolvem grandes volumes de dados. A utilização de soluções de computação em nuvem ou clusters de processamento poderia ter permitido uma análise mais ampla e um treinamento mais eficiente dos modelos.

Finalmente, a complexidade dos modelos foi outro fator crítico. Embora modelos mais complexos, como o Comitê Heterogêneo, tenham apresentado bom desempenho, sua implementação prática e interpretabilidade foram mais desafiadoras. Optar pelo Random Forest foi uma decisão baseada em um equilíbrio entre alta performance e facilidade de implementação, o que é crucial em cenários onde a transparência e a justificabilidade das decisões são importantes. Essas reflexões críticas devem ser levadas em conta em projetos

futuros, para que as escolhas de modelagem não comprometam a aplicabilidade prática das soluções.

Conclusão

O projeto desenvolvido foi bem-sucedido em alcançar seus objetivos principais, que incluíam a criação de um modelo capaz de otimizar campanhas de marketing direto para um banco. O modelo Random Forest emergiu como a melhor solução, apresentando um desempenho excelente em termos de acurácia, precisão e capacidade de discriminação, o que é essencial para identificar clientes com alta propensão a realizar depósitos a prazo. A integração deste modelo ao sistema de marketing do banco promete melhorar significativamente a eficiência das campanhas, resultando em maiores taxas de conversão e, consequentemente, em uma maior rentabilidade para a instituição.

As reflexões críticas sobre os desafios enfrentados durante o projeto, como o desbalanceamento das classes e as limitações computacionais, destacam a importância de uma abordagem cuidadosa na preparação dos dados e na escolha dos modelos. Essas lições não apenas ajudam a melhorar a eficácia do modelo atual, mas também fornecem diretrizes valiosas para futuros projetos que envolvam grandes volumes de dados e requisitos de alta precisão.

Em resumo, este projeto não apenas demonstrou o potencial do machine learning para transformar campanhas de marketing, mas também estabeleceu uma base sólida para o desenvolvimento de futuras iniciativas de inteligência artificial no banco. O sucesso do modelo Random Forest e as estratégias de implementação sugeridas oferecem um caminho claro para melhorar a eficácia operacional e a satisfação do cliente. Continuar a monitorar e ajustar o modelo conforme novos dados se tornam disponíveis será crucial para garantir que ele permaneça eficaz e relevante, adaptando-se às necessidades em evolução do banco e de seus clientes.