TESTE 1

Exercício 1

Suponha que você possui uma base de dados rotulada com 10 classes não balanceadas, essa base é formada por 40 features de metadados e mais 3 de dados textuais abertos.

Para todos os itens: Informe as bibliotecas usadas, se necessário, o motivo de cada decisão, explore as possibilidades.

a. Descreva como faria a modelagem dessas classes.

A solução deste desafio será realizada com a linguagem de programação Python e suas bibliotecas. Podemos usar o Colab, VS Code, Jupyter ou outra como IDE.

**Modelagem das Classes**

Para cumprimento desta etapa, faremos a Análise Exploratória dos Dados, o Pré-processamento dos dados, a Engenharia e Seleção das features, a Criação dos Modelos, a Avaliação das Métricas e Ensemble dos modelos (se necessário).

1. **Análise Exploratória de Dados (EDA)**:

* **Bibliotecas**: pandas, numpy, matplotlib, seaborn
* **Objetivo**: Compreender a distribuição das classes, identificar a relação entre as features e as classes, verificar a necessidade de balanceamento e explorar as características dos dados textuais.
* **Ações**:
  + Verificar a distribuição das 10 classes.
  + Identificar a correlação entre as features numéricas e as classes.
  + Analisar a frequência das palavras nos dados textuais.

1. **Pré-processamento:**

* **Bibliotecas**: sklearn, nltk/spacy (para processamento de texto)
* **Objetivo**: Preparar os dados para modelagem, incluindo tratamento de desbalanceamento e pré-processamento dos textos. Cada tipo de dado deve receber um tipo de pré-processamento.
* **Ações**:
  + **Dados Numéricos e Categóricos**:
    - Normalizar ou padronizar as features numéricas (StandardScaler ou MinMaxScaler do sklearn).
    - Codificar as features categóricas, se existirem, usando OneHotEncoder ou LabelEncoder.
  + **Dados Textuais**:
    - Realizar limpeza dos textos (remoção de stopwords, pontuação, etc.).
    - Tokenizar e vetorizar os textos usando TF-IDF ou Word Embeddings (ex.: Word2Vec, GloVe).
    - OBS: Se os dados textuais abertos forem padronizados, podemos transformar em dados categóricos. Se os dados de textuais abertos não forem relevantes para a classificação, podemos desconsiderar estas features.
  + **Tratamento do Desbalanceamento**:
    - Aplicar técnicas como SMOTE (oversampling), RandomUnderSampler, ou criar classes ponderadas no modelo (class\_weight em modelos do sklearn).

1. **Engenharia e Seleção de Features:**

* **Bibliotecas**: sklearn, featuretools, statsmodels
* **Objetivo**: Criar, transformar ou remover features para melhorar o desempenho do modelo.
* **Ações**:
  + **Combinação de Features**: Criar novas features a partir da combinação das existentes.
  + **Redução de Dimensionalidade**: Usar PCA (Principal Component Analysis) ou o LDA (Linear Discriminant Analysis) para reduzir a dimensionalidade, se necessário.
  + **Seleção de Features**: Usar SelectKBest, RFE ou Feature Importance de árvores de decisão para selecionar as features mais importantes.

1. **Modelagem:**
   * **Bibliotecas**: sklearn, xgboost, catboost, lightgbm, tensorflow
   * **Objetivo:** Modelar e treinar o modelo apropriado para a tarefa de classificação.
   * **Ações**:

* Testar diferentes algoritmos de classificação:
  + - * **Modelos baseados em Árvores**: XGBoost, LightGBM, Random Forest, devido à sua robustez e interpretabilidade.
      * **Redes Neurais**: Para capturar padrões complexos, especialmente se os dados textuais forem predominantes.
      * **Modelos de Regressão Logística ou SVM**: Podem ser usados como baseline, ou se as classes estiverem linearmente separáveis.
* Avaliação dos Melhores Hiperparâmetros:
* Usar GridSearchCV ou RandomizedSearchCV para otimizar.
* Treinamento dos Modelos:
  + Realizar cross-validation para avaliar a performance do modelo e ajustar hiperparâmetros.
  + Implementar early stopping e dropout (para redes neurais) para evitar overfitting.
  + Os dados também podem ser divididos em treinamento (80%) e teste usando a função chamada train\_test\_split da biblioteca scikit-learn.

1. **Avaliação das métricas:**

**Bibliotecas**: sklearn, seaborn, matplotlib, scikit\_posthocs, sciPy, numpy

**Objetivo**: Avaliar a performance dos modelos.

**Ações**:

* Utilizamos algumas métricas para avaliação dos modelos, tais como:
* Acurácia, Precision, Recall, F1-score, Matriz de Confusão e AUC-ROC.
* Podemos fazer também testes de hipóteses: Teste de Friedman, Teste de Nemenyi e Teste T de Student para sabermos se há diferença estatisticamente significativa entre os modelos e qual se comporta da melhor forma.

1. **Ensemble de Modelos (se necessário):**

**Objetivo**: Combinar modelos complementares para melhorar a performance final.

**Ações**:

* A depender dos resultados encontrados, pode ser interessante fazer um ensemble que é a combinação de modelos diferentes com o objetivo de melhor os resultados individuais.

b. Ao finalizar essa modelagem, como iria apresentar essa modelagem para a área contratante?

Para a apresentação da modelagem, seria feito um Relatório Detalhado das técnicas utilizadas, Visualizações dos Resultados e Discussões dos Impactos, da seguinte forma:

* **Relatório Detalhado**: Usar Jupyter Notebooks, Colab ou ferramentas de visualização como Power BI e Looker Studio para criar um documento interativo que inclua:
  + Explicação sobre o dataset e a EDA.
  + Descrição do pré-processamento realizado.
  + Resultados comparativos dos modelos testados (métricas de performance como AUC, Acurácia, F1-score, Recall, Precision).
  + Interpretação dos resultados e decisões tomadas durante a modelagem.
* **Visualizações**: Gráficos comparativos de performance, matrizes de confusão, e gráficos de importância das features.
* **Discussão de Impacto**: Explicação sobre como o modelo pode ser usado pela área contratante, impactos esperados, e considerações sobre possíveis limitações.

c. Como faria a validação desse modelo?

Na etapa de validação do modelo, podemos aplicar a técnica de dividir a base de dados em treinamento (80%) e teste (20%) ou optamos por dividir em treinamento (60%), validação (20%) e teste (20%).

Outra técnica que pode ser usada é a de **Cross-Validation, usando a biblioteca** sklearn. Esta técnica utiliza validação cruzada (K-Fold) para garantir a generalização do modelo.

Para mensurar os resultados, podemos usar as métricas AUC-ROC, Acurácia, Precision, Recall, F1-score, e Matriz de Confusão. Para isso, usamos a biblioteca sklearn.

Por fim, podemos usar em outros dados reais e nunca apresentados ao modelo, preferencialmente de períodos temporais diferentes, para simular o comportamento em produção.

d. Supondo que esses dados são recebidos diariamente, como iria trabalhar com esse desafio?

Para trabalhar com dados recebidos diariamente, temos que usar:

* **Pipeline de dados automatizados:**
  + **Bibliotecas**: Airflow
  + **Objetivo**: Automação do processo de ingestão de dados, pré-processamento e scoring diário.
* **Treinamento Contínuo:**
  + **Bibliotecas**: sklearn, TensorFlow
  + **Objetivo**: Implementar retraining periódico ou conforme necessário (baseado em monitoramento de drift de dados).
* **Monitoramento e Alerta:**
  + **Bibliotecas**: Prometheus, Grafana
  + **Objetivo**: Monitorar a performance do modelo em produção e alertar em caso de queda de performance.

e. Como levaria esse projeto para um ambiente produtivo?

Pensando em subir este projeto para um ambiente de prod, temos que fazer o Deploy no Ambiente de Produção, atendendo aos requisitos de:

* **Model Deployment:**
  + **Bibliotecas/Plataformas**: Docker, Kubernetes, FastAPI, Flask, MLflow
  + **Objetivo**: Containerizar o modelo e disponibilizá-lo via API para que outras aplicações possam consumir.
* **Documentação:**
  + **Objetivo**: Criar documentação detalhada para manutenção, escalabilidade, e troubleshooting do modelo em produção.

f. EXTRA - Existe mais algo que gostaria de relatar sobre esse caso?

**Algumas considerações adicionais podem ser levadas em consideração:**

* **Explicabilidade:**
  + **Bibliotecas**: SHAP, LIME
  + **Objetivo**: Garantir que o modelo seja explicável, o que é crucial para a aceitação por parte da área contratante.
* **Escalabilidade:**
  + **Motivo**: Considerar a escalabilidade do modelo e do pipeline, utilizando cloud computing se necessário (por exemplo, AWS, Azure, GCP).
* **Segurança e Compliance:**
  + **Objetivo**: Verificar conformidade com a legislação de proteção dos dados ou outras regulações, especialmente em relação aos dados textuais e sensíveis.