RELATÓRIO ACCS ADML43 OFICINA DE PROJETOS EM INTELIGÊNCIA ARTIFICIAL

GOVERNO DO ESTADO
BAHIA
SECRETARIA DA SAÚDE

Projeto: Implementação de um Assistente Virtual para Atendimento ao Cliente

Equipe 7: Ana Clara Almeida Moreira, Daniel Oliveira Santiago da Silva, Emily Santos Sancho, Felipe Carvalho Goes, João Vitor Moreira de Jesus

GESTÃO DO PROJETO

- **Levantamento dos Requisitos do Projeto**

Objetivo Principal: Desenvolvimento de um protótipo de inteligência artificial para a SESAB com objetivo de predizer sobre o disgnóstico da dengue com base em dados de pacientes no Estado da Bahia.

Requisitos Funcionais:

- O protótipo de inteligência artificial deve ser capaz de predizer a ocorrência da dengue

Requisitos Não Funcionais:

- Segurança e privacidade dos dados dos usuários.
- Desempenho eficiente, com respostas rápidas às consultas dos usuários.
- Apresentar acurácia acima ou igual à mínima considerada aceitável pelo time da SESAB

DEFINIÇÃO DO ESCOPO MACRO DE TRABALHO

CAD

Coleta e Análise de Dados

Extração de dados relevantes da base SESAB e análise preliminar.

T2

Pré-processamento dos Dados

Limpeza, transformação e redução de dimensionalidade dos dados.

T3

Desenvolvimento de Algoritmos

Implementação de algoritmos de machine learning para classificação e predição.

T4

Avaliação e Validação Implementação do classificador

Produtização e Deploy

CRIAÇÃO DOS AMBIENTES DE GESTÃO

Ferramentas Utilizadas:

Trello: Para gestão de tarefas e acompanhamento do progresso do projeto.

Ambientes Criados:

- Ambiente de Desenvolvimento: Configurado com todas as ferramentas e bibliotecas necessárias para o desenvolvimento do projeto.
- Ambiente de Teste: Utilizado para testes e validação dos modelos antes do deploy.
- Ambiente de Produção: Configurado para deploy do assistente virtual, garantindo alta disponibilidade e escalabilidade.

Gestão de Projetos - ACCS

Quadro

Gestão

- Responsabilidades de gestão
- Atualizações Semanais

+ Adicionar um cartão

Pré-processamento

- Responsabilidades dev
- Atualizações Semanais

+ Adicionar um cartão

Programadores Jr e Sr

- Responsabilidades de
- Atualizações Semanais

+ Adicionar um cartão

Prioridades

- Terminar a parte da programação Machine Learning
  20 de jun.
- Fazer o Dashboard dos dados da base
  21 de jun.
- Manter o trello organizado

+ Adicionar um cartão

de Projetos - ACCS

Quadro

Entregáveis

18/06 - Estatística descritiva com distribuição dos dados, padrões e outliers para o pré-processamento definido; Divisão dos conjuntos de treino, teste e validação; Validação dos algoritmos com métricas de P, R e F1. Deploy em um container e conexão com dados;

18 de jun. 0/4

25/06 - Dashboard de visualização dos dados; Avaliação dos hiperparâmetros e proposta de novos hiperparâmetros mais adequados aos dados; Monitoramento de desempenho do modelo;

+ Adicionar um cartão

Entregáveis Atrasados

04/06 - Estatística descritiva com distribuição dos dados, padrões e outliers; Divisão dos conjuntos de treino, teste e validação; Validação dos algoritmos com métricas de P, R e F1. Deploy em um container e conexão com dados;

4 de jun. 1 0/4

11/06 - Dashboard de visualização dos dados; Avaliação dos hiperparâmetros; Monitoramento da performance do modelo;

11 de jun. 1 0/3

+ Adicionar um cartão

Entregáveis Concluídos

+ Adicionar um cartão

Etapas Concluídas

Terminar o pré-processamen
base de dados

12 de jun. 1

.

.

.

+ Adicionar um cartão

PRÉ-PROCESSAMENTO DOS DADOS

Desafios Iniciais

- Dificuldades com a base de dados: A base continha muitas variáveis com escalas e tipos de dados distintos, o que dificultava as análises iniciais. Dimensionalidade alta.

Análise do Dicionário de Dados

- Estudo do dicionário Compreensão de cada coluna presente na base de dados.
- Identificação das colunas.
- Determinação de colunas necessárias e desnecessárias.

FILTRAGEM DE VARIÁVEIS

Definição e colunas

- Remoção de colunas não relevantes: Exclusão das colunas que não contribuíam para os objetivos do projeto.
- Alinhamento da base de dados: Adequação da base de dados aos objetivos estabelecidos.

.

Necessário analisar o dicionário de dados

Compreensão de cada coluna presente na base de dados.

CONCLUSÕES DA ANÁLISE DO DICIONÁRIO

Variáveis descartadas

- Números de unidades hospitalares.
- Telefones.
- Datas de investigação, exames, internação e coletas devido à redundância.
- Variáveis de acompanhamento do paciente após diagnóstico.

Variáveis mantidas

- Sintomas.
- Resultados de Exames
- Hospitalização
- CLASSI_FIN (definida como alvo para os modelos de predição).

TESTES DE PRÉ-PROCESSAMENTO

Configuração do Ambiente

Primeiro, configuramos o ambiente de trabalho, carregando as bibliotecas necessárias, definindo uma função para carregar os dados e, em seguida, carregando o dataset.

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report

# Função para carregar dados
def load_data():
    # Exemplo de carga de dados (substitua pelo seu dataset)
    data = pd.read_csv('seu_dataset.csv')
    return data

# Carregar dados
data = load_data()
```

PRÉ-PROCESSAMENTO FLEXÍVEL + HIPERPARÂMETROS DEFINIDOS

Neste método, utilizamos uma abordagem flexível para o pré-processamento e utilizando os hiperparâmetros definidos pela professora para cada modelo de machine learning.

Os 5 passos

1. **Divisão dos dados:** Separamos as features (X) do target (y).
2. **Divisão treino/teste:** Dividimos os dados em conjuntos de treino e teste.
3. **Normalização**: Aplicamos a normalização usando LabelEncoder e One-hot-encoder.
4. **Treinamento dos modelos:** Modelos implementados em Python usando a API Keras sobre a biblioteca Tensorflow e biblioteca Scikit-Learn
5. **Avaliação do modelo:** Avaliamos o modelo utilizando o conjunto de teste e imprimimos o relatório de classificação.

TESTES DE PRÉ-PROCESSAMENTO COM ABORDAGENS ESPECÍFICAS

Para enriquecer o relatório de pré-processamento, incluímos novas abordagens que lidam com dados nulos e balanceamento de classes.
Aqui estão os detalhes das novas abordagens:

Abordagem 1

Remover sintomas com valores nulos e dropar todos os exames, HOSPITALIZ e SOROTIPO

1. **Remoção de sintomas com valores nulos:** Todos os registros com sintomas nulos são removidos.
2. **Dropar exames, HOSPITALIZ e SOROTIPO:** As colunas relacionadas a exames, hospitalização e sorotipo são descartadas.

```
symptoms_columns = ['FEBRE', 'MIALGIA', 'CEFALEIA', 'EXANTEMA', 'VOMITO',
'NAUSEA', 'DOR_COSTAS', 'CONJUNTVIT', 'ARTRITE', 'ARTRALGIA',
'PETÉQUIA_N', 'LEUCOPENIA', 'LACO', 'DOR_RETRO']

exam_columns =
['RES_CHIKS1','RES_CHIKS2','RESUL_PRNT','RESUL_SORO','RESUL_NS1','RES
UL_VI_N','RESUL_PCR_','HISTOPA_N','IMUNOH_N']

'HOSPITALIZ', 'SOROTIPO’
```

Abordagem 1.1 - Remover sintomas com valores nulos e preencher todos os exames, HOSPITALIZ e SOROTIPO nulos com a moda

Abordagem 2 - Preencher valores de sintomas, exames, HOSPITALIZ e SOROTIPO nulos com a moda

Abordagem 2.1 - Preencher valores de sintomas nulos com a moda e dropar todos os exames, HOSPITALIZ e SOROTIPO

Abordagens 1, 1.1, 2, 2.1
Balanceadas - Abordagem 1, 1.1, 2, 2.1, respectivamente, com classificador Binário

RESULTADOS

Abordagem	MLP	KNN RS	KNN Cross	Regreg RS	Regreg Cross	Arvore RS	Arvore Cross	Random F RS	Random F Cross	Media RS	Media Cross	Desvio padrão
Base1	0.47080	0.38077	0.36228	0.44203	0.44006	0.43817	0.43615	0.44310	0.43929	0.44203	0.43772	0.03814750538
Base1_1	0.49800	0.43877	0.43668	0.47547	0.47315	0.46577	0.46871	0.47130	0.47430	0.47130	0.47093	0.01785014099
Base2	0.54690	0.49217	0.39420	0.48797	0.48459	0.51480	0.51236	0.52147	0.51595	0.51480	0.49848	0.05680574413
Base2_1	0.50560	0.43527	0.41312	0.45823	0.45617	0.47687	0.47944	0.48127	0.48150	0.47687	0.46781	0.03177369811
Base1 Balanceada	0.58790	0.52823	0.52894	0.56647	0.56287	0.56427	0.56210	0.56407	0.56147	0.56427	0.56179	0.01661320057
Base1_1 Balanceada	0.66660	0.61687	0.61728	0.66413	0.66103	0.64927	0.65362	0.65273	0.65695	0.65273	0.65529	0.02018871467
Base2 Balanceada	0.71060	0.66943	0.67294	0.69443	0.69456	0.68620	0.68967	0.69007	0.69341	0.69007	0.69154	0.01002314156
Base2_1 Balanceada	0.63680	0.58303	0.55742	0.60433	0.59985	0.61177	0.60711	0.61003	0.60727	0.61003	0.60348	0.0239134082

CONCLUSÃO

Os testes de pré-processamento ajudam a determinar qual método prepara melhor os dados para o modelo de machine learning, resultando em melhor desempenho e previsões mais precisas. Cada abordagem tem suas vantagens e pode ser escolhida com base nas necessidades específicas do projeto e nos dados disponíveis. Os diferentes métodos de pré-processamento permitem abordar as questões de dados nulos e balanceamento de classes de maneiras variadas. Os métodos balanceados ajudam a garantir que o modelo não seja tendencioso em relação a uma classe específica, enquanto as diferentes formas de tratar dados nulos podem impactar a performance do modelo.

FINE-TUNING DE HIPER PARÂMETROS

MELHORES MODELOS

MLP: [optimizer=Adam; layers=[100, relu; 1, sigmoid]; loss=binary_crossentropy; loss_weights=0.01; epochs=20] - Accuracy: 0.7012

KNN: [n_neighbors=50; weights=uniform; metric=euclidean] - Accuracy: 0.69502*

Regressão Logística: [penalty=l2; solver=lbfgs; max_iter=100; C=1.0] - Accuracy: 0.69510

Árvore de Decisão: [criterion=log_loss; max_depth=100; min_samples_split=2; min_samples_leaf=50] - Accuracy: 0.70064

Random Forest: [criterion=log_loss n_estimators=2000 max_features=sqrt] - Accuracy: 0.69515*

*Reduced set 100.000 instances

CONCLUSÃO DOS TESTES

Pelos resultados pode-se observar que apesar dos extensivos testes, pouca acurácia foi ganha com a variação dos hiper parâmetros, levando a conclusão de que o maior potencial para ganho de performance está na melhora do pré-processamento. Isso, contudo, só pode ser feito em trabalho conjunto com a SESAB, pois há dados faltantes e incoerentes na base original que tiveram que ser supostos ou removidos, prejudicando a acurácia dos dados na representação de cenários positivos e negativos.