PROJETO DE CONCLUSÃO DO MÓDULO ENGENHARIA DE MACHINE LEARNING

BRUNO ADALBERTO DOS SANTOS

1 - A solução criada nesse projeto deve ser disponibilizada em repositório git e disponibilizada em servidor de repositórios (Github (recomendado), Bitbucket ou Gitlab). O projeto deve obedecer o Framework TDSP da Microsoft (estrutura de arquivos, arquivo requirements.txt e arquivo README - com as respostas pedidas nesse projeto, além de outras informações pertinentes). Todos os artefatos produzidos deverão conter informações referentes a esse projeto (não serão aceitos documentos vazios ou fora de contexto). Escreva o link para seu repositório.

RESPOSTA: https://github.com/BrunoBersan/KobeBryant

2 - Iremos desenvolver um preditor de arremessos usando duas abordagens (regressão e classificação) para prever se o "Black Mamba" (apelido de Kobe) acertou ou errou a cesta. Baixe os dados de desenvolvimento e produção aqui (datasets: dataset_kobe_dev.parquet e dataset_kobe_prod.parquet). Salve-os numa pasta /data/raw na raiz do seu repositório.
Para começar o desenvolvimento, desenhe um diagrama que demonstra todas as etapas necessárias para esse projeto, desde a aquisição de dados, passando pela criação dos modelos, indo até a operação do modelo.

RESPOSTA: As descrições e o gráfico estão nas sessões “**Diagrama de Pipelines e Fluxos” e “Detalhes dos Pipelines”**

3 - Como as ferramentas Streamlit, MLFlow, PyCaret e Scikit-Learn auxiliam na construção dos pipelines descritos anteriormente? A resposta deve abranger os seguintes aspectos:

a. Rastreamento de experimentos;
b. Funções de treinamento;
c. Monitoramento da saúde do modelo;
d. Atualização de modelo;
e. Provisionamento (Deployment).

RESPOSTA: As descrições das ferramentas e como foram utilizadas estão na sessão “**Ferramentas”**

4 - Com base no diagrama realizado na questão 2, aponte os artefatos que serão criados ao longo de um projeto. Para cada artefato, a descrição detalhada de sua composição.

RESPOSTA: As descrições dos artefatos estão na Sessão: “**Descrição dos artefatos gerados**”

5 - Implemente o pipeline de processamento de dados com o mlflow, rodada (run) com o nome "PreparacaoDados":

a. Os dados devem estar localizados em "/data/raw/dataset_kobe_dev.parquet" e "/data/raw/dataset_kobe_prod.parquet"

b. Observe que há dados faltantes na base de dados! As linhas que possuem dados faltantes devem ser desconsideradas. Para esse exercício serão apenas consideradas as colunas:

i. lat

ii. lng

iii. minutes remaining

iv. period

v. playoffs

vi. shot_distance

A variável shot_made_flag será seu alvo, onde 0 indica que Kobe errou e 1 que a cesta foi realizada. O dataset resultante será armazenado na pasta "/data/processed/data_filtered.parquet". Ainda sobre essa seleção, qual a dimensão resultante do dataset?

vii. Separe os dados em treino (80%) e teste (20 %) usando uma escolha aleatória e estratificada. Armazene os datasets resultantes em "/Data/processed/base_{train|test}.parquet . Explique como a escolha de treino e teste afetam o resultado do modelo final. Quais estratégias ajudam a minimizar os efeitos de viés de dados.

viii. Registre os parâmetros (% teste) e métricas (tamanho de cada base) no MlFlow

RESPOSTA: Foi utilizada a estrutura do Kedro, então os nomes foram alterados. A etapa de preparação dos dados, filtro e seleção de features estão em:

(a) - data/01_raw – Arquivos baixados para o projeto
(b) – Questão b
    a. data/02_intermediate – Remoção de nulos
    b. data/03_primary – Remoção de duplicatas e normalização dos dados
    c. data/04_feature – Separação das features do projeto conforme o anuncio
    d. data/05_model_input – Separação de treino e teste estratificado 80% / 20%
    e. data/06_models – Modelos criados e treinados
    f. data/07_model_output – Análises e métricas dos modelos
    g. Os registros no MLFLow podem ser encontrados no pipeline "model_training" - os detalhes e explicações podem ser encontrados no Read na sessão "**Detalhes dos Pipelines**"

6 - Implementar o pipeline de treinamento do modelo com o MlFlow usando o nome "Treinamento"

a. Com os dados separados para treinamento, treine um modelo com regressão logística do sklearn usando a biblioteca pyCaret.

b. Registre a função custo "log loss" usando a base de teste

c. Com os dados separados para treinamento, treine um modelo de árvore de decisão do sklearn usando a biblioteca pyCaret.

d. Registre a função custo "log loss" e F1_score para o modelo de árvore.

e. Selecione um dos dois modelos para finalização e justifique sua escolha.

RESPOSTA:

(a) – O resultado pode ser visualizado no pipeline "model_training"
(b) – O registro do logloss e outras métricas podem ser observados em 07_model_output/metrics_* para visualizar em formato .csv e também em 08_reporting/metrics_* para visualizar em formato de tabela e plot
(c) – Pode ser observado no pipeline "model_training" no nó: "logistic_regression_model"
(d) – pode ser observado em metrics_report_table_DT_test, metrics_report_table_DT_train e metrics_report_table_decision_tree_model_prod
(e) – Conclusão sobre os modelos estão nas sessões "Resultados" e "Possíveis Melhorias" no Readme

7 - Registre o modelo de classificação e o sirva através do MLFlow (ou como uma API local, ou embarcando o modelo na aplicação). Desenvolva um pipeline de aplicação (aplicacao.py) para carregar a base de produção (/data/raw/dataset_kobe_prod.parquet) e aplicar o modelo. Nomeie a rodada (run) do mlflow como "PipelineAplicacao" e publique, tanto uma tabela com os resultados obtidos (artefato como .parquet), quanto log as métricas do novo log loss e f1_score do modelo.

a. O modelo é aderente a essa nova base? O que mudou entre uma base e outra? Justifique.

b. Descreva como podemos monitorar a saúde do modelo no cenário com e sem a disponibilidade da variável resposta para o modelo em operação.

c. Descreva as estratégias reativa e preditiva de retreinamento para o modelo em operação.

RESPOSTA:

(a) – A conclusão pode ser encontrada na sessão "**Problemas no Projeto**" do readme
(b) – A conclusão pode ser encontrada na sessão "**Monitoramento e Saúde do Modelo**"
(c) – A conclusão pode ser encontrada na sessão "**Monitoramento e Saúde do Modelo**"

8 - Implemente um dashboard de monitoramento da operação usando Streamlit.

RESPOSTA:

A implementação pode ser visualizada seguindo a estrutura da sessão "**Como executar o projeto" etapa "5. Para executar a Dashboard do stremlit".**

RUBRICAS

1 - **O aluno categorizou corretamente os dados?**

RESPOSTA: Pode ser avaliado em pipelines/data_preparation

2 - O aluno integrou a leitura dos dados corretamente à sua solução?

RESPOSTA: Pode ser avaliado em pipelines/data_preparation e data_processing

3 - **O aluno aplicou o modelo em produção (servindo como API ou como solução embarcada)?**

RESPOSTA: Pode ser avaliado em pipelines/predict_api_decision_tree e também em pipelines/predict_api_logistic_regression e através do passo a passo 4. Da sessão "**Como executar o projeto**"

4 - **O aluno indicou se o modelo é aderente a nova base de dados?**

RESPOSTA: Pode ser avaliado na sessão "**Problemas no Projeto**" e "**Resultados**"

5 - O aluno criou um repositório git com a estrutura de projeto baseado no Framework TDSP da Microsoft?

RESPOSTA: Estrutura utilizada foi o Kedro, conforme discutido em aulas.

6 - O aluno criou um diagrama que mostra todas as etapas necessárias para a criação de modelos?

RESPOSTA: Pode ser observado em /docs no projeto ou no Readme

7 - O aluno treinou um modelo de regressão usando PyCaret e MLflow?

RESPOSTA: Pode ser avaliado no pipeline "model_training" no nó "logistic_regression_model"

8 - O aluno calculou o Log Loss para o modelo de regressão e registrou no mlflow?

RESPOSTA: Pode ser avaliado em /07_model_output em metrics.* como .csv e também em 08_reporting em Metrics.* no formato .png

9 - O aluno treinou um modelo de árvore de decisao usando PyCaret e MLflow?

RESPOSTA: Pode ser avaliado no pipeline "model_training" no nó "decision_tree_model"

10 - O aluno calculou o Log Loss e F1 Score para o modelo de árvore de decisão e registrou no mlflow?

RESPOSTA: Pode ser avaliado em /07_model_output em metrics.* como .csv e também em 08_reporting em Metrics.* no formato .png e no pipeline "model_training" no nó "decision_tree_model"

11 - O aluno indicou o objetivo e descreveu detalhadamente cada artefato criado no projeto?

RESPOSTA: Pode ser avaliado na sessão “**Descrição dos artefatos gerados**” no readme

12 - O aluno cobriu todos os artefatos do diagrama proposto?

RESPOSTA: Pode ser avaliado na sessão “**Descrição dos artefatos gerados**” no readme

13 - O aluno usou o MLFlow para registrar a rodada "Preparação de Dados" com as métricas e argumentos relevantes?

RESPOSTA: Pode ser avaliado no pipeline “model_training”

14 - O aluno removeu os dados faltantes da base?

RESPOSTA: Pode ser avaliado no pipeline “data_preparation”

15 - O aluno selecionou as colunas indicadas para criar o modelo?

RESPOSTA: Pode ser avaliado no pipeline “data_processing”

16 - O aluno indicou quais as dimensões para a base preprocessada?

RESPOSTA: Pode ser avaliado no pipeline “data_processing”

17 - O aluno criou arquivos para cada fase do processamento e os armazenou nas pastas indicadas?

RESPOSTA: Pode ser avaliado na estrutura do projeto em /data e /src

18 - O aluno separou em duas bases, uma para treino e outra para teste?

RESPOSTA: Pode ser avaliado na estrutura do projeto 05_model_input e no pipeline “data_processing”

19 - O aluno criou um pipeline chamado "Treinamento" no MlFlow?

RESPOSTA: Foram criados 2 pipelines 1 para Logistic_regression e outro para decision_tree. Podem ser avaliados nos nós decision_tree_model e logistic_regression_model no pipeline “model_training”

20 - O aluno identificou a diferença entre a base de desenvolvimento e produção?

RESPOSTA: Pode ser avaliado no Readme na sessão "**Problemas no Projeto**" e "**Resultados**"

21 - O aluno descreveu como monitorar a saúde do modelo no cenário com e sem a disponibilidade da variável alvo?

RESPOSTA: Pode ser avaliado no Readme na sessão "**Monitoramento e Saúde do Modelo**"

22 - O aluno implementou um dashboard de monitoramento da operação usando Streamlit?

RESPOSTA: A implementação pode ser visualizada seguindo a estrutura da sessão "**Como executar o projeto" et**
Dashboard do stremlit".

23 - O aluno descreveu as estratégias reativa e preditiva de retreinamento para o modelo em operação?

RESPOSTA: Pode ser avaliado no Readme na sessão "**Monitoramento e Saúde do Modelo**" e "**Estratégias de Retreinamento de Modelos em Produção**"