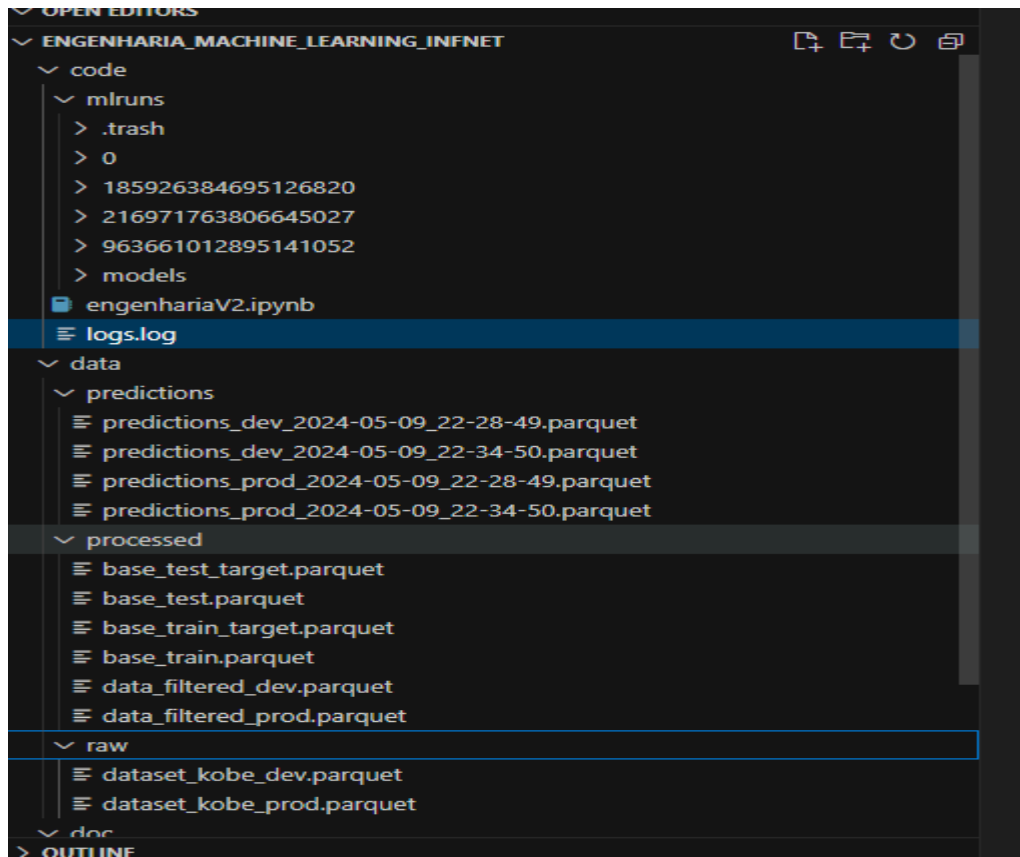


<p>1. Criar uma solução de streaming de dados usando pipelines</p> <p>O aluno criou um repositório git com a estrutura de projeto baseado no Framework TDSP da Microsoft?</p>	<p>Não demonstrou o item de rubrica</p>	<p>Demonstrou o item de rubrica</p>	<p>a estrutura nao segue o padrao de separacao de codigos, dados e documentacoes.</p>
--	---	-------------------------------------	---





ENGENHARIA_MACHINE_LEARNING_INFNET

Private

Unwatch 1

main

1 Branch 0 Tags

Go to file

Add file

<> Code



AndersonSouza22 subida

0c7b42e • 2 minutes ago

3 Commits

code	subida	2 minutes ago
data	subida	2 minutes ago
doc	subida	2 minutes ago
mlruns/0	123	1 hour ago
.gitattributes	Initial commit	1 hour ago
README.md	123	1 hour ago
requirements.txt	123	1 hour ago

máquinas?

1. Criar uma solução de streaming de dados usando pipelines

O aluno descreveu a importância de um pipeline de produção de modelos de aprendizagem de máquinas?

Não demonstrou o item de rubrica

Demonstrou o item de rubrica

Faltou mais no slide 6 a importancia de pipelines na passagem de dev para prod

Arquitetura da Solução

Importância pipelines na Passagem de DEV para PROD

Testes: Podem incluir etapas de teste e validação que verificam a qualidade do modelo antes de ser implantado em produção. Isso reduz o risco de bugs.

Controle de Versões: Podem integrar sistemas de controle de versão para rastrear todas as alterações feitas no código, dados e configurações do modelo.

Implantação Contínua: Podem facilitar a implantação contínua de modelos em produção, permitindo atualizações rápidas e frequentes com o mínimo de tempo de inatividade.

Monitoramento e Manutenção: Podem incluir ferramentas de monitoramento que acompanham o desempenho do modelo em produção e alertam sobre quaisquer problemas ou mudanças no comportamento do modelo ao longo do tempo.

<p>3. Preparar um modelo previamente treinado para uma solução de streaming de dados</p> <p>O aluno usou o MLFlow para registrar a rodada "Preparação de Dados" com as métricas e argumentos</p>	Não demonstrou o item de rubrica	Demonstrou o item de rubrica	Nao foi observado no codigo do git um run criado com esse nome
---	----------------------------------	------------------------------	--

```

import os
import pandas as pd
from sklearn.model_selection import train_test_split
from pycaret.classification import setup
import mlflow

if mlflow.active_run():
    mlflow.end_run()
# MLflow
mlflow.set_experiment("PreparacaoDados")

# Iniciar novo MLflow
with mlflow.start_run(run_name="PreparacaoDados"):

    # C treinamento e teste
    dev_data_path = "C:/Users/Anderson/Desktop/REDES_NEURAI/ENGENHARIA_MACHINE_LEARNING/REDES_NEURAI/dev_data.parquet"
    prod_data_path = "C:/Users/Anderson/Desktop/REDES_NEURAI/ENGENHARIA_MACHINE_LEARNING/REDES_NEURAI/prod_data.parquet"
    dev_df = pd.read_parquet(dev_data_path)
    prod_df = pd.read_parquet(prod_data_path)

    # colunas necessárias
    selected_columns = ['lat', 'lon', 'minutes_remaining', 'period', 'playoffs', 'shot_attempts']
    dev_df = dev_df[selected_columns]
    prod_df = prod_df[selected_columns]

    # Remover linhas com dados faltantes
    dev_df.dropna(inplace=True)
    prod_df.dropna(inplace=True)

```

codigo está no arquivo

C:\Users\Anderson\Desktop\REDES_NEURAI\ENGENHARIA_V2\ENGENHARIA_MACHI

NE_LEARNING_INFNET\code\engenhariaV2.ipynb

3. Preparar um modelo previamente treinado para uma solução de streaming de dados O aluno criou um pipeline chamado "Treinamento" no MlFlow?	Não demonstrou o item de rubrica	Demonstrou o item de rubrica	Nao foi observada a criacao de um run no mlflow com o nome Treinamento
--	----------------------------------	------------------------------	--

```
import mlflow
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import log_loss, f1_score
from pycaret.classification import setup, create_model

# MLflow
mlflow.set_experiment("Treinamento")

# treinamento e teste filtrados
dev_processed_path = "C:/Users/Anderson/Desktop/REDES_NEURAI/ENGENHARIA_MACHINE_LEARNING/da
prod_processed_path = "C:/Users/Anderson/Desktop/REDES_NEURAI/ENGENHARIA_MACHINE_LEARNING/d
dev_df = pd.read_parquet(dev_processed_path)
prod_df = pd.read_parquet(prod_processed_path)

# Separar os dados filtrados para treinamento e teste
X_dev = dev_df.drop(columns=['shot_made_flag'])
y_dev = dev_df['shot_made_flag']
X_train, X_test, y_train, y_test = train_test_split(X_dev, y_dev, test_size=0.2, stratify=y_

# Iniciar do MLflow
with mlflow.start_run(run_name="Treinamento"):

    # Treinar o modelo de regressão logística
    with mlflow.start_run(run_name="Logistic Regression", nested=True):
        lr_model = create_model('lr', verbose=False)
        lr_pred_test = lr_model.predict_proba(X_test)
        lr_logloss = log_loss(y_test, lr_pred_test)
        mlflow.log_param("model", "Logistic Regression")
        mlflow.log_metric("log_loss", lr_logloss)
        print(f"Métricas para Regressão Logística:\nLog Loss: {lr_logloss}")
```

código está no arquivo

C:\Users\Anderson\Desktop\REDES_NEURAI\ENGENHARIA_V2\ENGENHARIA_MACHI
NE_LEARNING_INFNET\code\engenhariaV2.ipynb

3. Preparar um modelo previamente treinado para uma solução de streaming de dados O aluno treinou um modelo de regressão usando PyCaret e MLflow?	Não demonstrou o item de rubrica	Demonstrou o item de rubrica	O aluno usou um automl e nao explicitou os resultados da regressao logistica
---	----------------------------------	------------------------------	--

usado o PyCaret e MLflow

```
import mlflow
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import log_loss, f1_score
from pycaret.classification import setup, create_model

# MLflow
mlflow.set_experiment("Treinamento")

# treinamento e teste filtrados
dev_processed_path = "C:/Users/Anderson/Desktop/REDES_NEURAIS/ENGENHARIA_MACHINE_LEA
prod_processed_path = "C:/Users/Anderson/Desktop/REDES_NEURAIS/ENGENHARIA_MACHINE_LEA
dev_df = pd.read_parquet(dev_processed_path)
prod_df = pd.read_parquet(prod_processed_path)

# Separar os dados filtrados para treinamento e teste
X_dev = dev_df.drop(columns=['shot_made_flag'])
y_dev = dev_df['shot_made_flag']
X_train, X_test, y_train, y_test = train_test_split(X_dev, y_dev, test_size=0.2, str

# Iniciar do MLflow
with mlflow.start_run(run_name="Treinamento"):

    # Treinar o modelo de regressão logística
    with mlflow.start_run(run_name="Logistic Regression", nested=True):
        lr_model = create_model('lr', verbose=False)
        lr_pred_test = lr_model.predict_proba(X_test)
        lr_logloss = log_loss(y_test, lr_pred_test)
        mlflow.log_param("model", "Logistic Regression")
        mlflow.log_metric("log_loss", lr_logloss)
        print(f"Métricas para Regressão Logística:\nLog Loss: {lr_logloss}")
```

resultados abaixo

```
... Métricas para Regressão Logística:
    Log Loss: 0.6777885179806518
```

<p>3. Preparar um modelo previamente treinado para uma solução de streaming de dados</p> <p>O aluno calculou o Log Loss para o modelo de regressão e registrou no mlflow?</p>	<p>Não demonstrou o item de rubrica</p>	<p>Demonstrou o item de rubrica</p>	<p>O aluno usou um auttml e nao explicitou os resultados da regressao logistica</p>
--	---	-------------------------------------	---

log loss registrado e calculado

```
import mlflow
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import log_loss, f1_score
from pycaret.classification import setup, create_model

# MLflow
mlflow.set_experiment("Treinamento")

# treinamento e teste filtrados
dev_processed_path = "C:/Users/Anderson/Desktop/REDES_NEURAIIS/ENGENHARIA_MACHINE
prod_processed_path = "C:/Users/Anderson/Desktop/REDES_NEURAIIS/ENGENHARIA_MACHINE
dev_df = pd.read_parquet(dev_processed_path)
prod_df = pd.read_parquet(prod_processed_path)

# Separar os dados filtrados para treinamento e teste
X_dev = dev_df.drop(columns=['shot_made_flag'])
y_dev = dev_df['shot_made_flag']
X_train, X_test, y_train, y_test = train_test_split(X_dev, y_dev, test_size=0.2,

# Iniciar do MLflow
with mlflow.start_run(run_name="Treinamento"):

    # Treinar o modelo de regressão logística
    with mlflow.start_run(run_name="Logistic Regression", nested=True):
        lr_model = create_model('lr', verbose=False)
        lr_pred_test = lr_model.predict_proba(X_test)
        lr_logloss = log_loss(y_test, lr_pred_test)
        mlflow.log_param("model", "Logistic Regression")
        mlflow.log_metric("log_loss", lr_logloss)
        print(f"Modelo de Regressão Logística treinado com log_loss={lr_logloss}")
```

... Métricas para Regressão Logística:
Log Loss: 0.6777885179806518

3. Preparar um modelo previamente treinado para uma solução de streaming de dados O aluno treinou um modelo de árvore de decisão usando PyCaret e MLflow?	Não demonstrou o item de rubrica	Demonstrou o item de rubrica	O aluno usou um automl e nao explicitou os resultados da arvore de decisao
---	----------------------------------	------------------------------	--

```
# árvore de decisão
with mlflow.start_run(run_name="Decision Tree", nested=True):
    tree_model = create_model('dt', verbose=False)
    tree_pred_test = tree_model.predict_proba(X_test)
    tree_logloss = log_loss(y_test, tree_pred_test)
    tree_f1 = f1_score(y_test, tree_model.predict(X_test))
    mlflow.log_param("model", "Decision Tree")
    mlflow.log_metric("log_loss", tree_logloss)
    mlflow.log_metric("f1_score", tree_f1)
    print(f"Métricas para Árvore de Decisão:\nLog Loss: {tree_logloss}\nF1 Score: {tree_f1}")
```

Métricas para Árvore de Decisão:
Log Loss: 4.378011482875853
F1 Score: 0.8137661703685624

4. Utilizar algoritmo de AutoML O aluno aplicou o modelo em produção (servindo como API ou como solução embarcada)?	Não demonstrou o item de rubrica	Demonstrou o item de rubrica	O aluno nao usou o mlflow para gerenciar o modelo usado em prod, mas somente o pycaret.
--	----------------------------------	------------------------------	---

```

# Salvar o melhor modelo como um arquivo .pkl
save_model(automl_model, "best_model_pycaret")

# Fazer previsões nos dados de teste com o melhor modelo
predictions = predict_model(automl_model, data=dev_test)

# Log das métricas
mlflow.log_metric("AUC", roc_auc_score(predictions['shot_made_flag'], predictions['prediction_label']))
mlflow.log_metric("Log Loss", log_loss(predictions['shot_made_flag'], predictions['prediction_label']))
mlflow.log_metric("F1 Score", f1_score(predictions['shot_made_flag'], predictions['prediction_label']))

# Salvar o modelo no MLflow
mlflow.sklearn.log_model(automl_model, "best_model")

# Registrar o modelo para produção
model_uri = f"file:///C:/Users/Anderson/Desktop/REDES_NEURAI/ENGENHARIA_MACHINE_LEARNING/models/best_model"
mlflow.register_model(model_uri, "best_model_production")

```


4. Utilizar algoritmo de AutoML O aluno indicou se o modelo é aderente a nova base de dados?	Não demonstrou o item de rubrica	Demonstrou o item de rubrica	O aluno nao comparou os resultados do modelo para a base de dev e prod
---	----------------------------------	------------------------------	--

```
# Imprimir as métricas
print("Métricas para os dados de produção:")
print(f"Acurácia: {accuracy_prod}")
print(f"Precisão: {precision_prod}")
print(f"Recall: {recall_prod}")

print(f"F1-Score: {f1_prod}")
print(f"Log Loss: {logloss_prod}")
print("\nMétricas para os dados de desenvolvimento:")
print(f"Acurácia: {accuracy_dev}")
print(f"Precisão: {precision_dev}")
print(f"Recall: {recall_dev}")
print(f"F1-Score: {f1_dev}")
print(f"Log Loss: {logloss_dev}")
```

Métricas para os dados de produção:

Acurácia: 0.6

Precisão: 0.6

Recall: 1.0

F1-Score: 0.7499999999999999

Log Loss: 0.6931471805599453

Métricas para os dados de desenvolvimento:

Acurácia: 0.6

Precisão: 0.6

Recall: 1.0

F1-Score: 0.7499999999999999

Log Loss: 0.8171703953063254

Ambos modelos têm desempenho semelhante em termos de acurácia, precisão e recall para ambos os conjuntos de dados de produção e desenvolvimento. No entanto, o log loss é ligeiramente maior nos dados de desenvolvimento em comparação com os dados de produção

Com base nessa análise, pode-se argumentar que o modelo parece ser mais aderente aos dados de produção, pois possui um log loss mais baixo nesse conjunto de dados. Isso sugere que as previsões do modelo estão mais calibradas em

relação às probabilidades verdadeiras no conjunto de dados de produção

4. Utilizar algoritmo de AutoML O aluno identificou a diferença entre a base de desenvolvimento e produção?	Não demonstrou o item de rubrica	Demonstrou o item de rubrica	O aluno nao comparou os resultados do modelo para a base de dev e prod
---	----------------------------------	------------------------------	--

```
• Dados de produção:  
  probability
```

```
0      0.5  
1      0.5  
2      0.5  
3      0.5  
4      0.5
```

```
Dados de desenvolvimento:  
  probability
```

```
0      0.500000  
1      0.500000  
2      0.500000  
3      0.731059  
4      0.500000
```

```
• Métricas para os dados de produção:
```

```
Acurácia: 0.6  
Precisão: 0.6  
Recall: 1.0  
F1-Score: 0.7499999999999999  
Log Loss: 0.6931471805599453
```

```
Métricas para os dados de desenvolvimento:
```

```
Acurácia: 0.6  
Precisão: 0.6  
Recall: 1.0  
F1-Score: 0.7499999999999999  
Log Loss: 0.8171703953063254
```

4. Utilizar algoritmo de AutoML O aluno descreveu como monitorar a saúde do modelo no cenário com e sem a disponibilidade da variável alvo?	Não demonstrou o item de rubrica	Demonstrou o item de rubrica	O aluno nao descreveu estrategias de monitoramento nos dois cenarios.
---	----------------------------------	------------------------------	---

Monitoramento Contínuo de Métricas de Desempenho

Acompanhar métricas de desempenho do modelo, como acurácia, precisão, recall, F1-score e log loss, em intervalos regulares

Validação Cruzada Temporal

Técnicas de validação cruzada temporal para avaliar o desempenho do modelo ao longo do tempo. Isso ajuda a identificar se o modelo está mantendo seu desempenho ao longo do tempo ou se está sofrendo de degradação do desempenho

Implementação de Sistemas de Alerta

sistemas de alerta para notificar quando o desempenho do modelo cair abaixo de um limiar aceitável. Isso pode ajudar a detectar rapidamente problemas no modelo e tomar medidas para solução do problema.