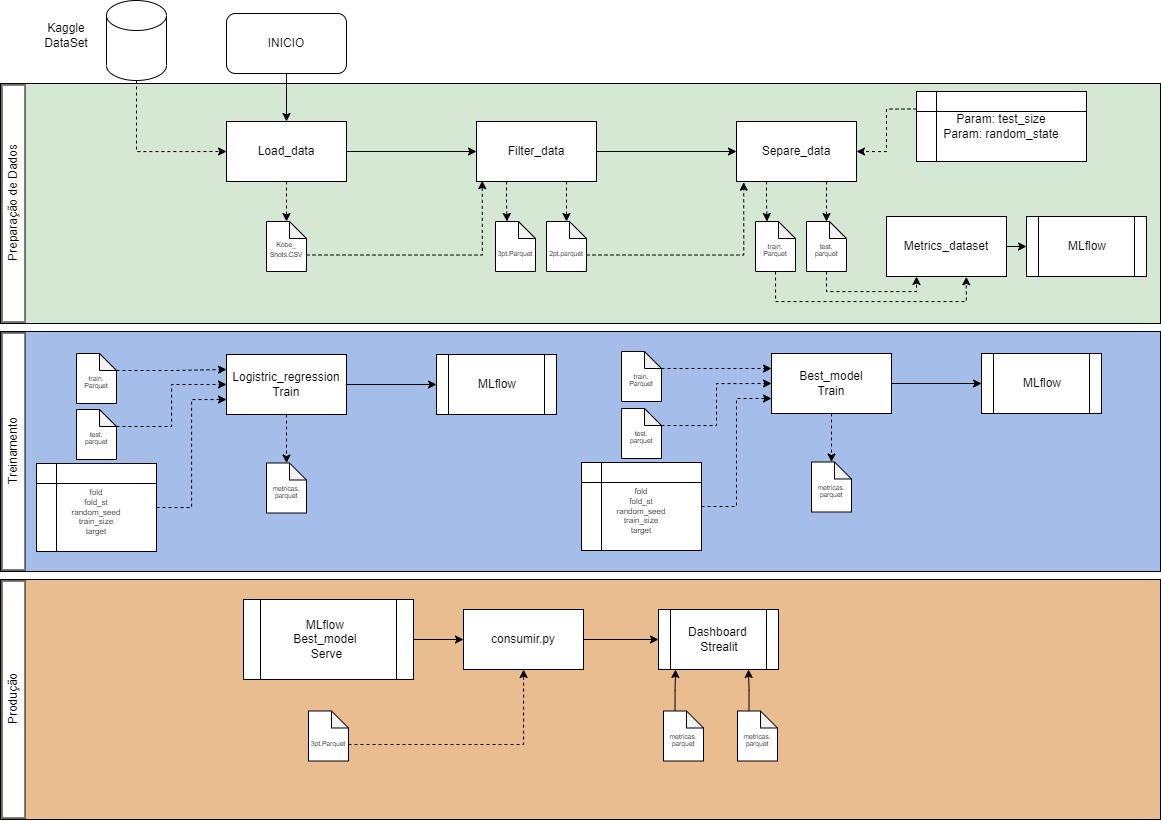
**PROJETO DA DISCIPLINA - ENGENHARIA DE MACHINE LEARNING**

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. 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.

[**https://github.com/andre-lucena12/kobe-shots**](https://github.com/andre-lucena12/kobe-shots)

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.  
Para começar o desenvolvimento, desenhe um diagrama que demonstra todas as etapas necessárias em um projeto de inteligência artificial desde a aquisição de dados, passando pela criação dos modelos, indo até a operação do modelo.'



3. Descreva a importância de implementar pipelines de desenvolvimento e produção numa solução de aprendizado de máquinas.

**A utilização de pipelines no desenvolvimento e na produção numa solução como a do projeto é importante para manter a estruturação de um projeto, sempre seguindo uma metodologia especifica de forma que uma ou mais pessoas possam trabalhar no mesmo projeto ao mesmo tempo, sem que uma interfira no trabalho da outra e que o desenvolvimento siga uma sinergia. Além disso, o fato de manter uma estruturação é importante quando a manutenção é necessária, pelo fato de que você pode mexer em funções especificas sem que tenha que verificar todo o código.**

4. 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).

**As quatro ferramentas estão ligadas a todos os aspectos descritos. O MLFlow, PyCaret e Scikit-Learn estão responsáveis por rastrear, treinar, monitorar a saúde e atualizar o modelo. O PyCaret trabalha junto do scikit-learn, sendo deles a função de treinar e atualizar, se necessário, o modelo. Com essas duas ferramentas é possível realizar modelos de classificação, clusterização e regressão. O MLFlow é o responsável pelo rastreamento e monitoramento do modelo, é com ele que podemos observar todos os treinamentos realizados e suas métricas, definir o status do ciclo de vida e efetuar a comparação entre os experimentos realizados. O Streamlit, por sua vez, é uma ferramenta de visualização que nos permite realizar a conexão entre o usuário e o modelo e fornecer uma experiência de provisionamento com o modelo treinado.**

5. Com base no diagrama realizado na questão 2, aponte os artefatos que serão criados ao longo de um projeto. Para cada artefato, indique qual seu objetivo.

* **Kobe-shots.CSV – DataSet vindo do kaggle**
* **3pt.parquet – DataSet filtrado com as colunas necessárias e cortado pela quantidade de pontos. Será utilizado para testar o modelo via API e requests**
* **2pt.parquet - DataSet filtrado com as colunas necessárias e cortado pela quantidade de pontos. Será utilizado para treinar e testar o modelo.**
* **Train.parquet – DataSet de treino do modelo**
* **Test.parquet – DataSet de teste do modelo**
* **Métricas\_lr.parquet – DataSet de métricas do modelo de regressão logística**
* **Métricas\_best\_model.parquet – DataSet de métricas do melhor modelo retornado pelo PyCaret**
* **Tuned\_lr\_model.pkl – Modelo de regressão logística treinado**
* **Tuned\_best\_model.pkl – Melhor modelo retornado pelo PyCaret treinado**

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

c. Separe os dados em treino (80%) e teste (20 %) usando uma escolha aleatória e estratificada. Armazene os datasets resultantes em "/Data/operalization/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.

**A separação treino e teste impacta diretamente o desempenho final do modelo. Uma separação inadequada pode causar um overtraining ou overfitting, ainda mais quando se tem um dataset desbalanceado. É importante prestar atenção na estrutura dos seus dados e ver se existe dependência entre as linhas do dataset por exemplo, dessa forma, se as linhas forem separadas no momento da divisão de treino e teste, isso vai impactar o desempenho do modelo. As estratégias que auxiliam na minimização dos efeitos danosos na construção do modelo podem ser validação cruzada, estratificação na separação treino/teste e um pré-processamento de dados, como normalização, visualização de correlação, feature engeeniering, etc.**

7. Implementar o pipeline de treinamento do modelo com o Mlflow usando o nome "Treinamento"

c. Com os dados separados para treinamento, treine um modelo de classificação do sklearn usando a biblioteca pyCaret. A escolha do algoritmo de classificação é livre. Justifique sua escolha.

**Para realização dessa etapa, eu usei o método compare\_models() do PyCaret. Esse método compara todos os modelos de classificação que a biblioteca possui e retorna o que tem a melhor performance para o dataset inferido.**

8. Registre o modelo de classificação e o disponibilize através do MLFlow através de API. Selecione agora os dados da base de dados original onde shot\_type for igual à 3PT Field Goal (será uma nova base de dados) e através da biblioteca requests, aplique o modelo treinado. Publique uma tabela com os resultados obtidos e indique o novo log loss e f1\_score.

a. O modelo é aderente a essa nova base? Justifique.

**Não. O DataSet inferido no modelo é de dados para cestas de 3 pontos. As features como longitude e latitude mudam bastante, uma vez que, pelas regras do esporte, as cestas de 3 pontos são computadas quando o jogador arremessa em uma área longe da cesta. Como o modelo foi treinado para cesta de 2 pontos, que são cestas que são feitas mais de perto, ele pensa que essas cestas não foram feitas e vai classificar como 0 a maioria dos dados. Isso fica evidente quando as métricas de**

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

**Para monitorar a saúde do modelo devemos sempre estar atentos as predições erradas, a distribuição dos dados, verificando se as novas coletas estão dentro da distribuição normal dos dados quando o modelo foi treinado e, sempre que necessário, atualizar o modelo.**

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

**A estratégia reativa de treinamento envolve o retreinamento do modelo somente após detectar uma queda significativa das métricas do modelo e definindo um limite aceitável em que elas podem operar, essa abordagem é útil quando o modelo está em um ambiente de produção controlado e previsível, onde a faixa dinâmica dos dados não muda muito.**

**Por outro lado, a estratégia preditiva de retreinamento envolve retreinar o modelo de tempos em tempos, quando uma nova coleta de dados é feita ou toda vez que um novo dado é inferido. Essa abordagem é mais útil em ambientes onde a produção não é previsível e a faixa dinâmica dos dados está sempre em alteração.**

9. Implemente um dashboard de monitoramento da operação usando Streamlit.

****

****

****

****

****