**Documentação do Projecto de Aprendizado de Máquina**

**Implantação**

**1. Visão geral**

A fase de implantação do projeto Análise de Fluxo Turístico Sustentável em Angola teve como objectivo disponibilizar o modelo de previsão de fluxo turístico em um ambiente acessível e interativo.  
O processo consistiu em:

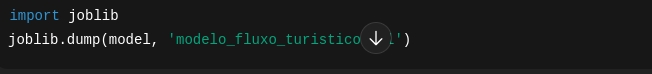
* Treinamento de modelos de Machine Learning (Regressão Linear e Random Forest) em Python usando dados históricos de chegadas e saídas de turistas;
* Serialização do modelo final para uso posterior;
* Criação de uma interface visual com Streamlit, permitindo interação com os resultados e geração de previsões futuras;
* Implementação de um script auxiliar para geração automática de relatórios (.docx) com indicadores e previsões.

Essa estrutura torna o sistema facilmente utilizável em computadores locais ou em servidores de nuvem (Google Colab, Streamlit Cloud, etc.).

**2. Serialização de modelos**

Após o treino e validação, os modelos de Machine Learning foram serializados usando a biblioteca joblib, que é otimizada para armazenar objetos Python grandes, como árvores de decisão ou modelos de regressão.

Formato: .pkl (pickle joblib)  
Exemplo:



Vantagens:

* Permite reusar o modelo sem precisar reentreinar;
* Ocupa pouco espaço de armazenamento;
* Pode ser carregado rapidamente para previsão:

Considerações:  
Os ficheiros são armazenados localmente na pasta do projeto, podendo futuramente ser migrados para um bucket seguro (como Google Cloud Storage ou AWS S3) para uso em produção.

**3. Modelo de Serviço**

O modelo serializado é utilizado para gerar previsões diretamente no dashboard Streamlit**.**O usuário pode carregar novos ficheiros CSV ou utilizar os dados padrões do INE (2010–2023).

Fluxo de utilização:

1. O usuário abre o painel interativo via streamlit run app.py;
2. Os dados são carregados e pré-processados;
3. O modelo lê a variável “Ano” e prevê o número de chegadas de turistas até 2025;
4. Os resultados são exibidos em forma de gráficos e tabelas.

Plataforma de Implantação:

* Ambiente Local (Linux – Ubuntu) durante o desenvolvimento;
* Pode ser facilmente migrado para Streamlit Cloud**,** RenderouHeroku, para acesso via navegador.

**4. Integração de API**

Embora o projeto atual utilize uma interface Streamlit, ele foi estruturado de modo a permitir integração futura com uma API REST usando o Flask ou FastAPI.

Exemplo de endpoint planeado:

* POST /predict
* Entrada (JSON):

Saída (JSON):



Essa integração permitirá conectar o modelo a sistemas externos, como plataformas de estatísticas do Ministério do Turismo ou dashboards online.

**5. Considerações de segurança**

Durante a implantação, foram consideradas boas práticas de segurança:

* Uso apenas de dados públicos e agregados, sem informações pessoais;
* Execução local em ambiente controlado (sem exposição direta à internet);
* Em caso de implantação em nuvem:
  + HTTPS para criptografia de dados em trânsito;
  + Tokens de autenticação para acesso à API;
  + Controle de permissões para uploads de arquivos.

Essas medidas garantem integridade dos dados e segurança no uso do modelo.

**6. Monitoramento e registro**

O desempenho do modelo pode ser monitorado através de:

* Métricas de erro registradas no treino: R², RMSE, MAE;
* Logs automáticos no console Streamlit e no arquivo de saída do Python;

Alertas visuais no dashboard, exibindo:

* Crescimento percentual anual;
* Top 5 províncias mais visitadas;
* Diferenças entre previsões da Regressão Linear e Random Forest.

Próximos passos:  
Para uso contínuo, pode ser implementado um sistema de monitoramento automatizado**,** com registro das previsões reais versus previstas, a fim de recalibrar o modelo periodicamente.