Mini Project 1 – Productionising Telco Churn Prediction

Transform your telco churn prediction project into a production-ready machine learning system with complete model and inference pipelines, tracking tools, and orchestrated workflows.

Dataset URL: https://www.kaggle.com/datasets/blastchar/telco-customer-churn

🎯 Project Objective

Advance your Telco Churn Prediction project by building complete **Model & Inference Pipelines**, integrating tracking tools, and orchestrating workflows. This project continues from the **data pipelines** you previously built in Mini Project 0.

Model Training

Build robust training pipelines with proper validation and testing

Inference Deployment

Create scalable prediction systems for real-time and batch processing

Experiment Tracking

Monitor and compare model performance across different iterations

📌 Project Assumptions

We assume that **data pipelines for cleaning, preprocessing, and feature engineering** have already been implemented. This project focuses on the advanced components of the ML lifecycle:

Model Training

Implementing robust training workflows with proper validation

Inference Deployment

Creating scalable prediction systems for production use

Experiment Tracking

Monitoring model performance and comparing iterations

Distributed Computing

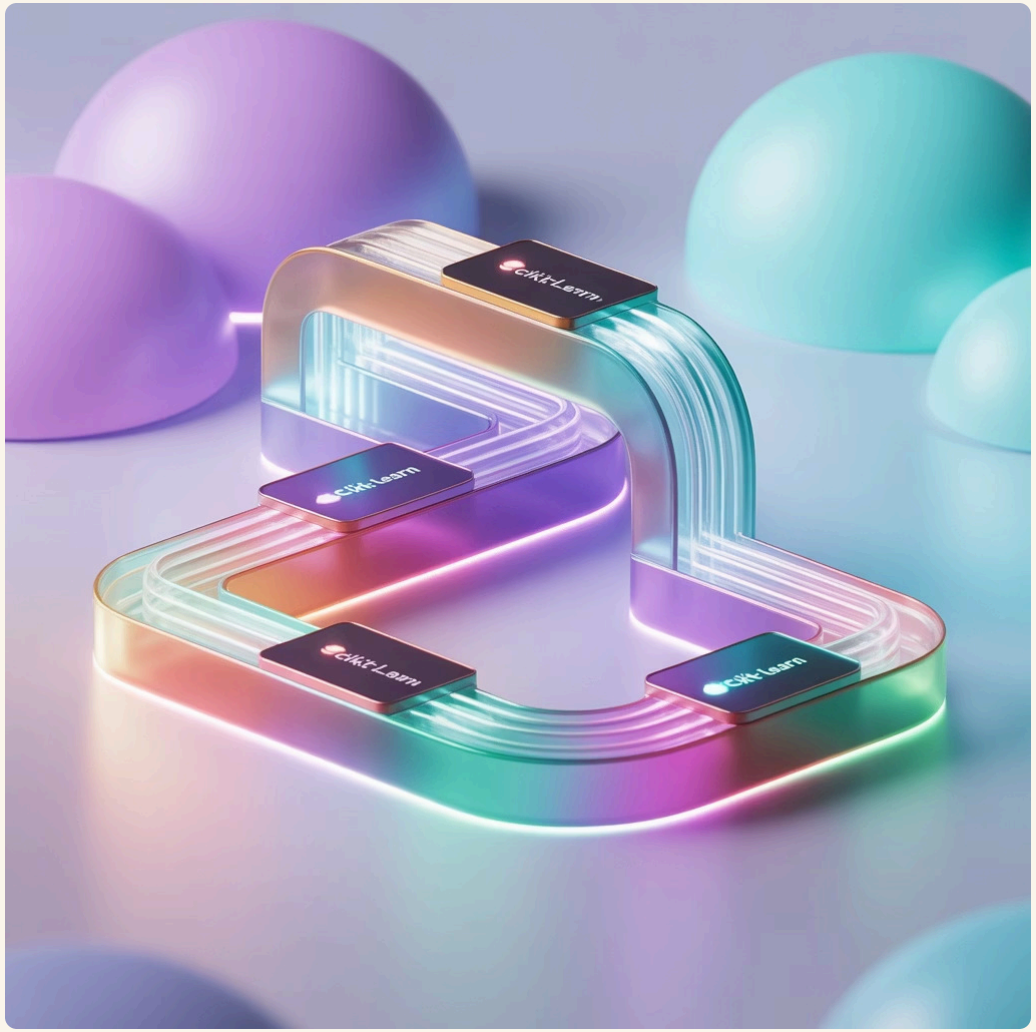
Leveraging Spark for large-scale data processing

Workflow Orchestration

Automating the entire ML pipeline with Airflow

⚙️ Part 1 – Build Model & Inference Pipelines

Scikit-Learn Implementation – 25 Marks



1 Reproducible Pipeline

Create a Scikit-learn pipeline that includes preprocessing, model training, and inference components

2 Data Handling

Ensure the pipeline handles unseen data properly and returns accurate predictions

3 Model Persistence

Use Pickle/Joblib to save and reload model and pipeline artifacts efficiently

4 Testing Framework

Include comprehensive test scripts that simulate inference on new input data

📋 Part 2 – Integrate MLflow Tracking

Experiment Management – 25 Marks

Comprehensive Tracking

Track experiments with MLflow including parameters, metrics, artifacts, and models for complete visibility

Model Comparison

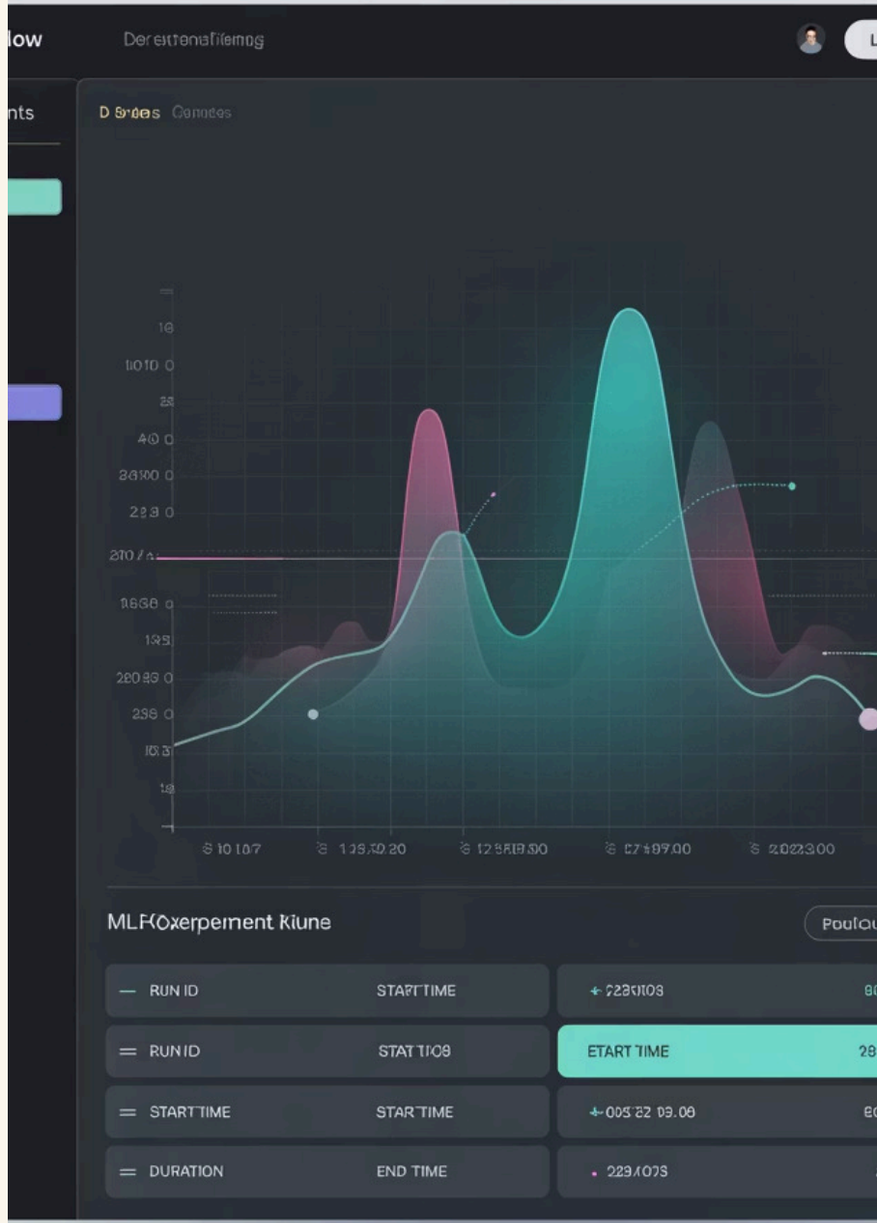
Use MLflow UI to compare multiple model versions and training runs for optimal performance selection

Detailed Logging

Log preprocessing steps, model hyperparameters, evaluation metrics, and visualisation plots

Documentation

Include screenshots of MLflow UI showing experiments, models, and runs for verification



🔥 Part 3 – Integrate Spark (PySpark MLlib)

Distributed Computing – 30 Marks

Leverage the power of distributed computing to handle large-scale telco datasets efficiently.

Pipeline Reconstruction

Rebuild preprocessing and model pipelines using PySpark DataFrame and MLlib APIs

Distributed Compatibility

Ensure pipeline compatibility with distributed processing using Spark's architecture

MLlib Models

Train equivalent models using MLlib (LogisticRegression, RandomForest, XGBoost)

Performance Analysis

Compare performance and execution time of Spark vs. Scikit-learn pipelines

🚀 Part 4 – Integrate Airflow for Orchestration

Workflow Automation – 20 Marks

1. **DAG Creation**
   Create an Airflow DAG defining the complete ML pipeline workflow
2. **Preprocessing**
   Data cleaning and feature engineering tasks
3. **Model Training**
   Automated model training with hyperparameter tuning
4. **Evaluation**
   Model performance assessment and validation
5. **Inference**
   Prediction generation and result storage

📦 Final Deliverables Checklist

Bonus 5 Marks Available

Project Structure

Well-organised codebase with

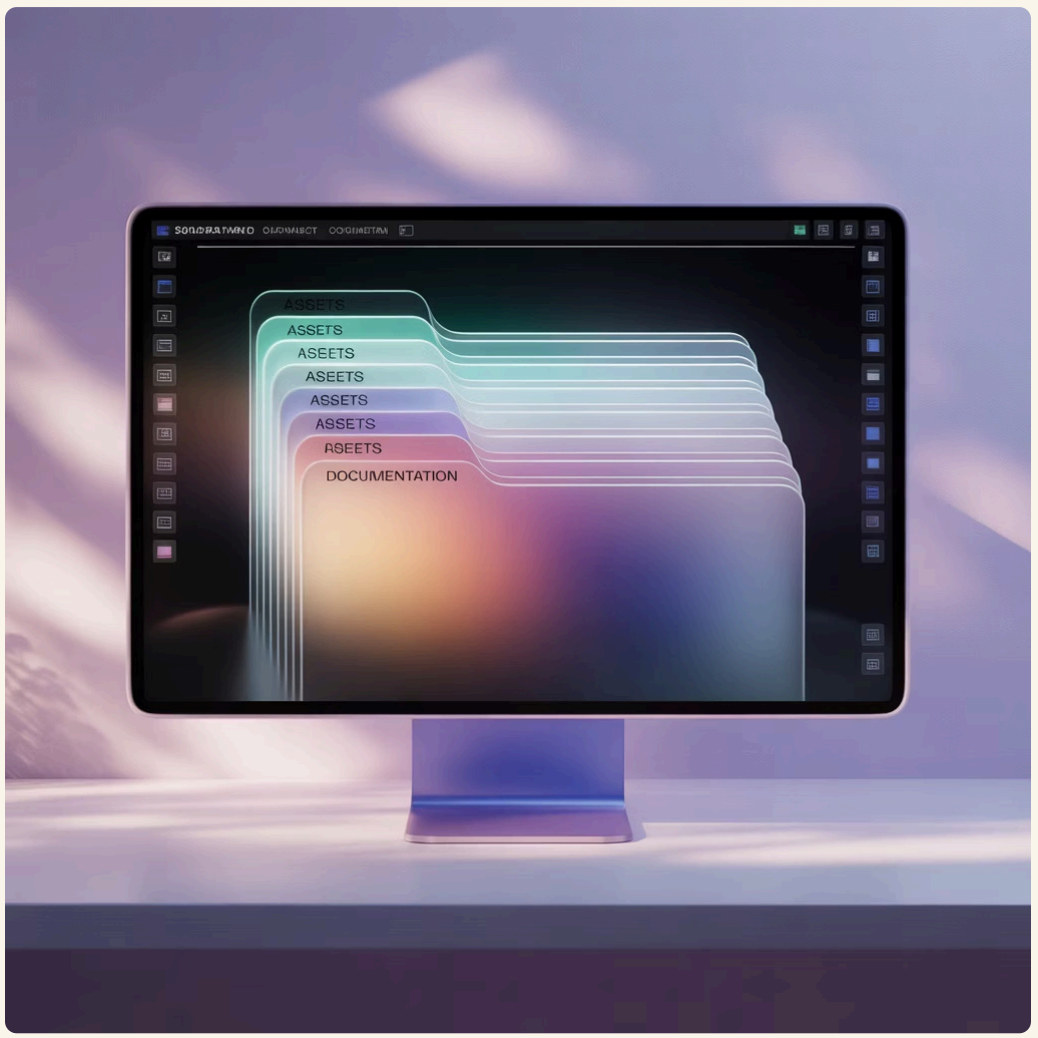
src/, pipelines/, notebooks/, and dags/ folders

Implementation Scripts

Model & inference scripts in both Scikit-learn and PySpark

MLflow Artifacts

Complete tracking artifacts and UI screenshots



Airflow Components

DAG file and screenshot of successful DAG run

Documentation

Comprehensive README file with instructions to run each component locally

Score Breakdown

Total: 105 Marks

This comprehensive project will demonstrate your ability to build production-ready machine learning systems with proper tracking, orchestration, and scalability considerations.