MLOps Hands-on Assignment Topics

Course: MLOps
Team Size: 5 Members per Group

General Instructions for All Assignments:

1. **Data Collection:** Students must crawl/scrape, collect, or use an API to gather relevant data.
2. **Data Processing:** Perform data cleaning, transformation, and feature engineering.
3. **Model Training & Evaluation:** Train models and evaluate their performance using appropriate metrics.
4. **MLOps Implementation:** Use tools like **MLflow, Airflow, DVC, Kubeflow, Docker, Kubernetes, and CI/CD pipelines** to automate the ML lifecycle.
5. **Deployment:** Deploy the trained model using **Flask, FastAPI, Kubernetes, or Serverless Architectures**.
6. **Monitoring & Maintenance:** Implement logging and monitoring to track model performance and drift.
7. **Final Deliverables:**
    - **GitHub Repository** with code, dataset, and documentation.
    - **Presentation** covering project implementation.
    - **Final Report** summarizing findings, challenges, and improvements.

---

1. Predict Top 10 Most Watched Netflix Movies

Description:
Scrape Netflix's trending movies or use **Kaggle datasets** to analyze patterns in popular movies. Build a model to predict which movies are likely to become top trending based on metadata (genre, actors, director, release date, etc.).

Expected Outcome:

- Web scraping pipeline using **BeautifulSoup/Scrapy** or API-based data collection.
- Machine learning model to predict trending movies.
- Model deployed as a REST API for recommendations.
- CI/CD pipeline for automated retraining when new data arrives.

---

2. Twitter Sentiment Analysis for Trending Topics

Description:
Use the **Twitter API (Tweepy) or scrape tweets** to analyze sentiment on trending topics. Build a **sentiment classification model** to determine if tweets are positive, neutral, or negative.

Expected Outcome:

- Data ingestion pipeline for collecting tweets.
- NLP-based model (LSTM, BERT, or traditional ML) for sentiment classification.
- Deployment as a sentiment analysis API.
- Model monitoring dashboard using **Grafana/Prometheus**.

---

3. Real Estate Price Prediction

Description:
Scrape real estate websites (Zillow, Realtor) or use **public datasets** to build a **house price prediction model** based on location, size, number of rooms, etc.

Expected Outcome:

- Data scraping pipeline for real estate listings.

- Regression model predicting house prices.
- CI/CD pipeline for model retraining.
- Deployment as a REST API for property price estimation.

---

4. Fake News Detection

Description:
Scrape news articles from various sources and build a **classification model** to detect fake news using NLP techniques.

Expected Outcome:

- Data pipeline collecting news articles from multiple sources.
- NLP-based model for fake news classification.
- Model deployed as a **browser extension or API**.
- Experiment tracking using **MLflow**.

---

5. Automated Stock Market Price Prediction

Description:
Collect **real-time stock market data** (Yahoo Finance API, Alpha Vantage) and build a **time-series forecasting model** to predict stock prices.

Expected Outcome:

- Data pipeline fetching stock market data every day.
- LSTM/ARIMA-based model for time-series forecasting.
- Deployment as a **dashboard with price predictions**.
- Model drift detection and retraining automation.

---

6. E-commerce Product Recommendation System

Description:
Scrape product reviews from e-commerce websites (Amazon, Flipkart) and build a **recommendation system** based on user preferences and product ratings.

Expected Outcome:

- Data pipeline collecting product reviews and ratings.
- Collaborative filtering or deep learning-based recommendation model.
- Model deployed as an API for personalized recommendations.
- A/B testing for model performance comparison.

---

7. Predict Customer Churn for a Subscription Service

Description:
Use **customer behavior data** (Netflix, Spotify, SaaS platforms) to predict whether a customer is likely to cancel their subscription.

Expected Outcome:

- Data collection pipeline for customer activity logs.
- Classification model predicting churn probability.
- Deployment as a **dashboard for company executives**.
- Model monitoring for real-time churn alerts.

---

8. Automated Resume Screening using NLP

Description:
Scrape job descriptions from LinkedIn/Indeed, collect resumes, and build a **resume screening model** that ranks candidates based on job fit.

Expected Outcome:

- Data pipeline collecting job descriptions and resumes.
- NLP-based model ranking resumes.
- Deployment as a **web app for recruiters**.
- CI/CD pipeline automating model improvements.

---

9. AI-Powered Chatbot for Customer Support

Description:
Scrape FAQs from company websites and build an **AI-powered chatbot** that answers customer queries automatically.

Expected Outcome:

- Data collection pipeline for FAQs and support tickets.
- NLP-based chatbot (Rasa, Dialogflow, GPT-3/4).
- Deployment as a **Telegram/WhatsApp bot**.
- Model monitoring with real-time feedback analysis.

Final Deliverables for Each Group:

1. **GitHub Repository** with source code, dataset, and deployment scripts.
2. **Final Report** documenting project implementation.
3. **Presentation** demonstrating the working solution.