IBM Data Science Projects

Project Instructions for Students:-

The graduation project is a key requirement for obtaining the Digital Egypt Pioneers Initiative Completion Certificate.

- Students are free to choose any of the ideas listed in the project booklet for their respective career track without any restrictions, they are able to choose other ideas not listed in the booklet but it should go in the same format of the ideas given.
- The project is a group assignment, and teams should consist of 4 to 6 students.
- Within a maximum of one week from the announcement of the project booklet, students must form their groups and inform the instructor. If they fail to do so, the instructor has the right to assign groups randomly and announce the team members.
- Students must divide the work responsibilities within the group and inform the instructor within two weeks of the project booklet announcement. During the final presentation, each group must demonstrate the work completed and each member's responsibility for their assigned tasks.
- The final evaluation will be based on the final presentation, which must include the students' adherence to the deliverables and the distribution of tasks among team members.

تعليمات المشروع للطلاب:-

مشروع التخرج هو أحد المتطلبات الأساسية للحصول على شهادة إتمام مبادرة رواد مصر الرقمية.

- يتمتع الطلاب بحرية اختيار أي من الأفكار المدرجة في كتيب المشروع لمسارهم الوظيفي دون أي قيود، أو اختيار أي فكره أخرى غير مدرجه، ولكن بنفس الطريقة المستخدمة في الأفكار المذكورة.
- المشروع عمل جماعي، ويجب أن تتكون فرق العمل من ٤ إلى ٦ طلاب.
- في غضون أسبوع كحد أقصى من إعلان كتيب المشروع، يجب على الطلاب تشكيل فرقهم وإبلاغ المدرب بذلك. في حالة عدم القيام بذلك، يحق للمدرب تقسيمهم بشكل عشوائي وإعلان أعضاء الفريق.
- يجب على الطلاب تقسيم مسؤوليات العمل داخل المجموعة وإبلاغ المدرب بها في غضون أسبوعين من إعلان كتيب المشروع. كما يجب على كل مجموعة خلال العرض النهائي توضيح الأعمال التي تم إنجازها وتحديد مسؤولية كل فرد في تنفيذها.
- سيتم التقييم النهائي بناءً على العرض النهائي، والذي يجب أن يتضمن التزام الطلاب بتسليم المخرجات وتقسيم العمل بين أعضاء الفريق.

Project Idea 1: Customer Churn Prediction and Analysis

Project Overview

Students will work on a real-world problem: predicting customer churn for a company using data science techniques. The project will include data collection, analysis, visualization, and model development, with a focus on practical applications using Python and various tools.

Week 1: Data Collection and Exploration

- **Tasks:**
  - **Data Collection:** Obtain a dataset related to customer churn. You can use publicly available datasets or synthetic data if necessary.
  - **Data Exploration:** Perform initial data exploration to understand the dataset, including data types, missing values, and basic statistics.
  - **Tools:** Python (Pandas, NumPy), SQL for data extraction if necessary.
- **Deliverables:**
  - A report summarizing the dataset and initial findings.
  - Exploratory Data Analysis (EDA) notebook with code and visualizations.

Week 2: Data Analysis and Visualization

- **Tasks:**
  - **Data Cleaning:** Handle missing values, outliers, and data transformation as needed.
  - **Data Analysis:** Perform detailed analysis, including statistical tests and feature engineering.
  - **Data Visualization:** Create visualizations to explore data patterns and relationships.
  - **Tools:** Python (Matplotlib, Seaborn, Plotly), SQL if needed.
- **Deliverables:**
  - A cleaned dataset and a comprehensive analysis report.
  - Interactive visualizations and dashboards displaying key insights.

Week 3: Machine Learning Model Development

- **Tasks:**
  - **Model Selection:** Choose appropriate machine learning models (e.g., logistic regression, decision trees, and random forests) for churn prediction.
  - **Model Training:** Train and evaluate the models using cross-validation and performance metrics.
  - **Model Optimization:** Fine-tune models and perform hyperparameter tuning.
  - **Tools:** Python (Scikit-learn, TensorFlow or PyTorch if using advanced models), MLflow for tracking experiments.
- **Deliverables:**
  - A detailed report on model performance, including metrics such as accuracy, precision, recall, and F1 score.
  - Python code for model training and evaluation.

Week 4: MLOps, Deployment, and Final Presentation

- **Tasks:**
  - **MLOps Implementation:** Use MLOps tools to manage and track model experiments. Implement model deployment using Hugging Face if applicable.
  - **Deployment:** Deploy the model to a cloud platform or create an interactive web app for predictions.
  - **Final Report and Presentation:** Prepare a final project report summarizing all work done, including data analysis, model development, and deployment. Create a presentation to display the project results.
  - **Tools:** MLflow, Hugging Face, cloud platforms (e.g., AWS, Azure) or web frameworks (e.g., Flask, Streamlit).
- **Deliverables:**
  - A deployed model or web application.
  - A final project report and presentation.

Project Idea 2: Sales Forecasting and Optimization

Week 1: Data Collection and Exploration

- **Tasks:**
  - **Data Collection:** Acquire a dataset related to sales, such as historical sales data from a retail or e-commerce store.
  - **Data Exploration:** Explore the dataset to understand trends, seasonality, and patterns.
  - **Tools:** Python (Pandas, NumPy), SQL if applicable.
- **Deliverables:**
  - Data exploration report and initial findings.
  - EDA notebook with basic statistics and visualizations.

Week 2: Data Analysis and Visualization

- **Tasks:**
  - **Data Cleaning:** Clean and preprocess the sales data, including handling missing values and outliers.
  - **Data Analysis:** Analyze sales patterns, seasonal trends, and correlations.
  - **Data Visualization:** Create visualizations to display sales trends and forecasts.
  - **Tools:** Python (Matplotlib, Seaborn, Plotly).
- **Deliverables:**
  - Cleaned dataset and analysis report.
  - Interactive visualizations and dashboards.

Week 3: Forecasting Model Development

- **Tasks:**
  - **Model Selection:** Choose time series forecasting models (e.g., ARIMA, Prophet).
  - **Model Training:** Train and evaluate forecasting models.
  - **Model Optimization:** Fine-tune models for better accuracy.
  - **Tools:** Python (Statsmodels, Prophet).
- **Deliverables:**
  - Forecasting model performance report.
  - Python code for model training and evaluation.

Week 4: MLOps, Deployment, and Final Presentation

- **Tasks:**
  - **MLOps Implementation:** Use tools like MLflow to track and manage forecasting models.
  - **Deployment:** Deploy the forecasting model via a web app or dashboard for users to input new data and see predictions.
  - **Final Report and Presentation:** Summarize the project work, including data analysis, forecasting models, and deployment.
  - **Tools:** MLflow, web frameworks (e.g., Flask, Streamlit).
- **Deliverables:**

- Deployed forecasting model or web app.
- Final report and presentation.

Project Idea 3: Sentiment Analysis of Social Media Posts

Week 1: Data Collection and Exploration

- **Tasks:**
  - **Data Collection:** Collect social media posts using APIs (e.g., Twitter API) or public datasets.
  - **Data Exploration:** Explore the dataset to understand text data characteristics.
  - **Tools:** Python (Tweepy for Twitter API, Pandas).
- **Deliverables:**
  - Data exploration report.
  - EDA notebook with basic text statistics and sample posts.

Week 2: Data Analysis and Visualization

- **Tasks:**
  - **Data Cleaning:** Preprocess text data (e.g., removing stop words, tokenization).
  - **Data Analysis:** Perform text analysis, including word frequency and sentiment scoring.
  - **Data Visualization:** Visualize sentiment trends and keyword frequencies.
  - **Tools:** Python (NLTK, SpaCy, Matplotlib, Seaborn).
- **Deliverables:**
  - Preprocessed text data and analysis report.
  - Sentiment trends visualizations and dashboards.

Week 3: Sentiment Analysis Model Development

- **Tasks:**
  - **Model Selection:** Choose sentiment analysis models (e.g., logistic regression, BERT).
  - **Model Training:** Train and evaluate sentiment analysis models.
  - **Model Optimization:** Fine-tune models for improved sentiment classification.
  - **Tools:** Python (Scikit-learn, Hugging Face Transformers).
- **Deliverables:**
  - Sentiment analysis model performance report.
  - Python code for model training and evaluation.

Week 4: MLOps, Deployment, and Final Presentation

- **Tasks:**
  - **MLOps Implementation:** Track and manage models using MLflow.
  - **Deployment:** Deploy the sentiment analysis model via a web app or API.
  - **Final Report and Presentation:** Summarize the project, including data analysis, sentiment analysis models, and deployment.
  - **Tools:** MLflow, web frameworks (e.g., Flask, Streamlit).
- **Deliverables:**
  - Deployed sentiment analysis model or web app.
  - Final report and presentation.

Project Idea 4: Healthcare Predictive Analytics

Week 1: Data Collection and Exploration

- **Tasks:**
  - **Data Collection:** Obtain healthcare-related datasets, such as patient records or health metrics.
  - **Data Exploration:** Explore the dataset to understand key features and data distribution.
  - **Tools:** Python (Pandas, NumPy), SQL if needed.
- **Deliverables:**
  - Dataset exploration report.
  - EDA notebook with basic statistics and visualizations.

Week 2: Data Analysis and Visualization

- **Tasks:**
  - **Data Cleaning:** Clean and preprocess healthcare data (e.g., handling missing values, normalization).
  - **Data Analysis:** Analyze health metrics and patient outcomes.
  - **Data Visualization:** Create visualizations to highlight health trends and anomalies.
  - **Tools:** Python (Matplotlib, Seaborn, Plotly).
- **Deliverables:**
  - Cleaned dataset and analysis report.
  - Visualizations of health trends and anomalies.

Week 3: Predictive Model Development

- **Tasks:**
  - **Model Selection:** Develop predictive models for healthcare outcomes (e.g., patient risk prediction).
  - **Model Training:** Train and evaluate models using appropriate metrics.
  - **Model Optimization:** Optimize models for better prediction accuracy.
  - **Tools:** Python (Scikit-learn, TensorFlow, or PyTorch).
- **Deliverables:**
  - Predictive model performance report.
  - Python code for model development and evaluation.

Week 4: MLOps, Deployment, and Final Presentation

- **Tasks:**
  - **MLOps Implementation:** Use MLflow for managing and tracking models.
  - **Deployment:** Deploy the predictive model via a web app or API for healthcare professionals.
  - **Final Report and Presentation:** Document the project, including data analysis, predictive models, and deployment.
  - **Tools:** MLflow, web frameworks (e.g., Flask, Streamlit).
- **Deliverables:**
  - Deployed predictive model or web app.

- Final report and presentation.