Microsoft Machine Learning 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.

<u>Project Idea 1: Customer Sentiment and Trend Analysis</u>

Week 1: Data Collection and Preprocessing

- **Tasks:**
  - **Data Collection:** Collect customer feedback data from social media or surveys.
  - **Data Preprocessing:** Clean and preprocess the data using Python (e.g., text normalization, removing stop words).
  - **Tools:** Python (Pandas, NLTK, SpaCy).
- **Deliverables:**
  - Cleaned dataset ready for analysis.
  - Data preprocessing notebook.

Week 2: Data Science and Machine Learning

- **Tasks:**
  - **Data Science Analysis:** Perform exploratory data analysis (EDA) to identify trends and patterns in the data.
  - **Machine Learning:** Build and evaluate sentiment analysis models using Python (e.g., Logistic Regression, Random Forest).
  - **Tools:** Python (Scikit-learn, Matplotlib).
- **Deliverables:**
  - EDA report with visualizations.
  - Sentiment analysis model and performance metrics.

Week 3: Advanced Techniques and Azure Integration

- **Tasks:**
  - **Natural Language Processing (NLP):** Apply attention-based NLP models to improve sentiment analysis (e.g., BERT).
  - **Azure AI Fundamentals:** Utilize Azure services to deploy and scale the sentiment analysis model.
  - **Tools:** Azure Machine Learning, Python (Transformers library).
- **Deliverables:**
  - Enhanced sentiment analysis model using attention mechanisms.
  - Azure deployment setup and integration.

Week 4: MLOps, GANs, and Final Presentation

- **Tasks:**
  - **MLOps:** Use MLflow to track and manage machine-learning models.
  - **Generative Models:** Implement a basic Generative Adversarial Network (GAN) to generate synthetic customer feedback data for analysis.
  - **Final Presentation:** Prepare a report and presentation summarizing the project work, including data analysis, model development, and deployment.
  - **Tools:** MLflow, Python (TensorFlow/PyTorch for GANs), Azure services.
- **Deliverables:**
  - Deployed sentiment analysis model and GAN implementation.
  - Final report and presentation.

<u>Project Idea 2: Predictive Maintenance for Industrial Equipment</u>

Week 1: Data Collection and Analysis

- **Tasks:**
    - **Data Collection:** Gather historical maintenance data from industrial equipment.
    - **Data Preprocessing:** Clean and preprocess the data for analysis.
    - **Tools:** Python (Pandas, NumPy).
- **Deliverables:**
    - Cleaned dataset with preprocessing steps documented.

Week 2: Machine Learning Model Development

- **Tasks:**
    - **Exploratory Data Analysis (EDA):** Analyze data to identify patterns related to equipment failures.
    - **Predictive Modeling:** Develop and evaluate machine-learning models to predict equipment failures (e.g., Decision Trees, SVM).
    - **Tools:** Python (Scikit-learn, Matplotlib).
- **Deliverables:**
    - EDA report and predictive maintenance models with performance metrics.

Week 3: Advanced Modeling and Azure Integration

- **Tasks:**
    - **NLP for Log Data:** Apply NLP techniques to analyze equipment logs and extract actionable insights.
    - **Azure AI Fundamentals:** Use Azure services to deploy the predictive model and integrate it with real-time monitoring systems.
    - **Tools:** Azure Machine Learning, Python (NLTK, Transformers).
- **Deliverables:**
    - Predictive model integrated with Azure services.
    - Real-time monitoring setup and documentation.

Week 4: MLOps, GANs, and Final Presentation

- **Tasks:**
    - **MLOps Implementation:** Track and manage predictive models using MLflow.
    - **GANs for Simulation:** Create a GAN to simulate various maintenance scenarios for further analysis.
    - **Final Report and Presentation:** Summarize the project, including data preprocessing, model development, and deployment.
    - **Tools:** MLflow, Python (TensorFlow/PyTorch for GANs), Azure services.
- **Deliverables:**
    - Deployed predictive maintenance model and GAN simulation.
    - Final report and presentation.

Project Idea 3: Personalized Recommendation System

Week 1: Data Collection and Preprocessing

- **Tasks:**
  - **Data Collection:** Gather user interaction data (e.g., product reviews, clickstream data).
  - **Data Preprocessing:** Clean and preprocess the data for recommendation modeling.
  - **Tools:** Python (Pandas, NumPy).
- **Deliverables:**
  - Cleaned dataset and preprocessing documentation.

Week 2: Machine Learning and Recommendation Modeling

- **Tasks:**
  - **Recommendation Algorithms:** Implement collaborative filtering and content-based recommendation models.
  - **Model Evaluation:** Evaluate the performance of recommendation models.
  - **Tools:** Python (Scikit-learn, Surprise library).
- **Deliverables:**
  - Recommendation models and performance evaluation.

Week 3: Advanced Techniques and Azure Integration

- **Tasks:**
  - **Generative Models:** Use GANs to generate synthetic user profiles for improving recommendations.
  - **Azure AI Fundamentals:** Deploy the recommendation system using Azure services for scalability.
  - **Tools:** Azure Machine Learning, Python (TensorFlow/PyTorch for GANs).
- **Deliverables:**
  - Enhanced recommendation system using GAN-generated profiles.
  - Azure deployment setup.

Week 4: MLOps and Final Presentation

- **Tasks:**
  - **MLOps:** Manage and track recommendation models with MLflow.
  - **Prompt Engineering:** Develop prompts for personalized user interactions (e.g., Chatbot).
  - **Final Report and Presentation:** Document the project, including recommendation models, deployment, and prompt engineering.
  - **Tools:** MLflow, Azure services, Python (for prompt engineering).
- **Deliverables:**
  - Deployed recommendation system with GANs and prompt engineering.
  - Final report and presentation.

Project Idea 4: Fraud Detection in Financial Transactions

Week 1: Data Collection and Preprocessing

- **Tasks:**
    - **Data Collection:** Obtain financial transaction data, including labeled fraudulent and non-fraudulent transactions.
    - **Data Preprocessing:** Clean and preprocess the data, addressing missing values and normalizing features.
    - **Tools:** Python (Pandas, NumPy).
- **Deliverables:**
    - Cleaned and preprocessed dataset.
    - Data preprocessing notebook.

Week 2: Statistical Analysis and Machine Learning

- **Tasks:**
    - **Statistical Analysis:** Perform statistical analysis to understand the distribution of fraud-related features.
    - **Machine Learning:** Develop and evaluate classification models for fraud detection (e.g., Logistic Regression, Random Forest).
    - **Tools:** Python (Scikit-learn, Statsmodels).
- **Deliverables:**
    - Statistical analysis report.
    - Fraud detection models and performance metrics.

Week 3: Advanced Techniques and Azure Integration

- **Tasks:**
    - **NLP for Transaction Notes:** Apply NLP techniques to analyze transaction descriptions or notes.
    - **Azure AI Fundamentals:** Deploy the fraud detection model using Azure Machine Learning or Azure Synapse.
    - **Tools:** Azure Machine Learning, Python (NLTK, SpaCy).
- **Deliverables:**
    - Enhanced fraud detection model with NLP integration.
    - Deployment setup on Azure.

Week 4: MLOps, GANs, and Final Presentation

- **Tasks:**
    - **MLOps:** Use MLflow to manage and track fraud detection models.
    - **GANs for Synthetic Data:** Implement a GAN to generate synthetic fraud transaction data for training and validation.
    - **Final Report and Presentation:** Document the project including data analysis, model development, and deployment.
    - **Tools:** MLflow, Python (TensorFlow/PyTorch for GANs), Azure services.
- **Deliverables:**
    - Deployed fraud detection model with synthetic data.
    - Final report and presentation.

AI & DATA SCIENCE TRACK

<u>Project Idea 5: Real-Time Speech Recognition and Translation</u>

Week 1: Data Collection and Preprocessing

- **Tasks:**
  - **Data Collection:** Collect speech data in different languages (e.g., public datasets or recorded samples).
  - **Data Preprocessing:** Clean and preprocess the audio data, including noise reduction and feature extraction.
  - **Tools:** Python (Librosa, PyDub).
- **Deliverables:**
  - Preprocessed audio dataset.
  - Data preprocessing report.

Week 2: Machine Learning and Speech Recognition

- **Tasks:**
  - **Speech Recognition:** Build and train a speech recognition model to transcribe audio to text (e.g., using Deep Learning models).
  - **Machine Translation:** Develop a translation model to translate transcribed text into different languages.
  - **Tools:** Python (TensorFlow/Keras, Hugging Face Transformers).
- **Deliverables:**
  - Speech recognition model and translation model with performance metrics.

Week 3: Advanced Techniques and Azure Integration

- **Tasks:**
  - **Attention Models for Translation:** Apply attention mechanisms to improve translation accuracy.
  - **Azure AI Fundamentals:** Deploy the speech recognition and translation models on Azure Cognitive Services.
  - **Tools:** Azure Cognitive Services, Python (Transformers library).
- **Deliverables:**
  - Enhanced translation model with attention mechanisms.
  - Deployment setup on Azure Cognitive Services.

Week 4: MLOps, GANs, and Final Presentation

- **Tasks:**
  - **MLOps:** Manage and track models using MLflow.
  - **GANs for Audio Data Augmentation:** Use GANs to augment audio data for improving model robustness.
  - **Final Report and Presentation:** Document the project including speech recognition, translation, and deployment.
  - **Tools:** MLflow, Python (TensorFlow/PyTorch for GANs), Azure services.
- **Deliverables:**
  - Deployed speech recognition and translation system with augmented data.
  - Final report and presentation.

Project Idea 6: Intelligent Chatbot with Custom Prompt Engineering

Week 1: Data Collection and Preprocessing

- **Tasks:**
  - **Data Collection:** Gather conversational data or use existing datasets for Chatbot training.
  - **Data Preprocessing:** Clean and preprocess the data, including tokenization and text normalization.
  - **Tools:** Python (NLTK, SpaCy).
- **Deliverables:**
  - Cleaned conversational dataset.
  - Data preprocessing documentation.

Week 2: Machine Learning and NLP

- **Tasks:**
  - **Chatbot Development:** Build a Chatbot using NLP techniques (e.g., Seq2Seq models).
  - **Prompt Engineering:** Develop custom prompts to improve Chatbot responses and user engagement.
  - **Tools:** Python (Transformers library, Hugging Face).
- **Deliverables:**
  - Functional Chatbot with custom prompts.
  - Evaluation of Chatbot performance.

Week 3: Azure Integration and Advanced Features

- **Tasks:**
  - **Azure AI Fundamentals:** Deploy the Chatbot using Azure Bot Services.
  - **Advanced Features:** Add advanced features like sentiment analysis and context-aware responses.
  - **Tools:** Azure Bot Services, Python (Azure SDK).
- **Deliverables:**
  - Deployed Chatbot with advanced features on Azure.
  - Integration setup and documentation.

Week 4: MLOps and Final Presentation

- **Tasks:**
  - **MLOps:** Track and manage Chatbot models using MLflow.
  - **Final Report and Presentation:** Summarize the project, including Chatbot development, prompt engineering, and deployment.
  - **Tools:** MLflow, Azure services.
- **Deliverables:**
  - Deployed Chatbot with MLOps tracking.
  - Final report and presentation.