### **Practical Assessment 3:**

## **Machine Learning Foundations and Model Building**

This assessment will evaluate your understanding of machine learning concepts, data preprocessing, exploratory data analysis, model training, and evaluation using Python.

### **Task 1: Data Preprocessing and Exploratory Data Analysis (EDA)**

Download a dataset from Kaggle (e.g., Titanic, Iris, or any dataset with categorical and numerical features).

* Perform the following preprocessing steps:
  + Handle missing values.
  + Encode categorical variables (Label Encoding / One Hot Encoding).
  + Feature scaling using StandardScaler or MinMaxScaler.
* Conduct EDA:
  + Summary statistics.
  + Correlation matrix.
  + Visualizations (distributions, pair plots, box plots, etc.).

**Deliverables:**

* Python Jupyter Notebook (.ipynb)
* Visualizations (.png or embedded in notebook)

### **Task 2: Supervised Learning Model**

Build a classification model using **Logistic Regression** or **Decision Tree**:

* Split the dataset into training and testing sets (use train\_test\_split).
* Train the model and evaluate using:
  + Accuracy, Precision, Recall, F1-Score
  + Confusion Matrix
  + ROC-AUC (if applicable)

**Deliverables:**

* Python script or Notebook (.ipynb)
* Model evaluation metrics

### **Task 3: Unsupervised Learning**

Use a clustering algorithm (e.g., KMeans):

* Choose a suitable dataset.
* Normalize the data.
* Apply KMeans and determine the optimal number of clusters using Elbow Method or Silhouette Score.
* Visualize the clusters using PCA or TSNE.

**Deliverables:**

* Python script or Notebook
* Clustering visualization and explanation of cluster characteristics

### **Task 4: Feature Engineering and Selection**

Using any dataset:

* Create new features based on domain knowledge or interactions.
* Apply feature selection techniques:
  + Correlation analysis
  + Recursive Feature Elimination (RFE)
  + Tree-based feature importance

**Deliverables:**

* Python script or Notebook
* Explanation of selected features

### **Task 5: Model Deployment Basics**

* Save the trained model using joblib or pickle.
* Create a simple Flask or FastAPI web server that accepts user input and returns predictions.
* Test the API locally with sample inputs.

**Deliverables:**

* .py file with model API code
* Screenshot or screencast of running the API

A close up of a number

AI-generated content may be incorrect.

A black screen with blue text

AI-generated content may be incorrect.

### **Task 6: GitHub and Documentation**

* Upload all your scripts, notebooks, and assets to GitHub.
* Include a README.md file with:
  + Project overview
  + Steps followed
  + Instructions to run the code

**Deliverables:**

* GitHub repo link