AI/ML Task

Objective

Develop a machine learning model to detect fraudulent transactions using a Kaggle dataset, with a focus on data handling, model training, evaluation, and explainability. **Implementing an unsupervised model will be given higher preference** to showcase skills in handling unlabelled data and anomaly detection.

Dataset

Use the **Kaggle Credit Card Fraud Detection dataset** (or similar) to ensure consistency and familiarity. This dataset is commonly used for fraud detection challenges, making it accessible and comparable across candidates.

Task Breakdown

1. Data Exploration and Preprocessing

- **Goal**: Analyze and prepare the data for model training.
- **Steps**:
    - Summarize key statistics of the dataset, including class distribution, feature distribution, and any missing values.
    - Handle any missing values and perform required transformations (e.g., scaling or normalizing).
    - **Imbalance Handling**: Use an imbalance handling technique, such as **undersampling**, **SMOTE**, or **class weights** during training.
- **Deliverable**: Documented analysis and preprocessing steps, ideally in a Jupyter notebook.

2. Model Development (Supervised)

- **Goal**: Train a supervised model to classify fraudulent vs. non-fraudulent transactions.
- **Steps**:
    - Start with a baseline model (e.g., **Logistic Regression**) to establish a starting point.
    - Train an **XGBoost model** (or similar) as the primary supervised model.

    - Perform basic hyperparameter tuning to optimize for accuracy and recall.
- **Deliverable**: Code to train both models and a summary of performance metrics (accuracy, recall, precision, F1-score).

3. Model Development (Unsupervised)

- **Goal**: Implement an unsupervised model to identify potential fraudulent activities.
- **Steps**:
    - Use an **anomaly detection algorithm** (e.g., **Isolation Forest** or **Autoencoder**) to detect outliers.
    - Configure the model to detect rare events, capturing transactions that differ significantly from typical patterns.
- **Deliverable**: Code and explanation of the unsupervised approach, including sample results or anomalies detected.
- **Note**: This unsupervised model is optional but will be given **higher preference** in evaluations.

4. Model Evaluation

- **Goal**: Measure and compare model performance for both supervised and unsupervised approaches.
- **Steps**:
    - For supervised models, use **Recall**, **Precision**, and **F1-score** to understand model performance, especially for the fraud class.
    - For unsupervised models, evaluate the effectiveness of anomaly detection by checking identified anomalies against known fraudulent transactions.
    - Generate and analyze **Confusion Matrices** for supervised models and visualize ROC-AUC/PR-AUC curves where possible.
- **Deliverable**: Model evaluation section with metrics, confusion matrix, and performance plots.

5. Basic Explainability

- **Goal**: Provide insights into model behavior and feature importance.
- **Steps**:
    - For the XGBoost model, extract **feature importance** to identify the top features impacting fraud classification.
    - Use **SHAP** or another interpretability tool to explain sample predictions.
- **Deliverable**: Summary of top features and visual explanations for individual predictions.

Deliverables Summary

Submit the following as a GitHub repository or similar:

1. **Jupyter Notebook or Python Script**: Contains all steps from data exploration to model evaluation.
2. **README.md**: Briefly outlines the setup, key choices made, and instructions for reproducing the results.
3. **Documentation**:
    - Model performance summary with key metrics and explanations.
    - Visualizations and plots for evaluation (confusion matrix, ROC/PR curves).
    - Overview of the unsupervised approach if implemented, including examples of detected anomalies.

Evaluation Criteria

- **Data Preprocessing**: Thorough handling of missing values, outliers, and class imbalance.
- **Modeling and Metrics**: Effective use of both supervised and unsupervised models (with preference for unsupervised).
- **Explainability and Insights**: Ability to extract feature importance and interpret model decisions.
- **Documentation**: Clear code comments, a well-organized notebook/script, and a concise README.