**FRAUD\_DETECTION\_AUTOENCODER**

**Author: Swarna Anjani Devershetty**

**Description: Fraud detection using AutoEncoder (Unsupervised Learning) with PyOD**

The given Python script is designed to perform credit card fraud detection using an unsupervised deep learning technique—AutoEncoder—implemented via the PyOD library. The goal is to detect anomalous patterns in transaction data that may indicate fraudulent behavior. The script begins by importing necessary libraries. pandas and numpy are used for data manipulation and numerical operations, while matplotlib.pyplot is used for plotting visualizations. sklearn.preprocessing.StandardScaler is imported for feature scaling, and sklearn.metrics provides various tools for evaluating the model's performance. The AutoEncoder class is imported from the pyod.models.auto\_encoder module, which is a part of the PyOD library specifically designed for anomaly detection.

The second section focuses on loading the dataset. The dataset is a CSV file named creditcard.csv, which contains anonymized credit card transaction records. The dataset includes various features such as transaction time, amount, and engineered principal components (V1 to V28), along with a binary Class label indicating whether a transaction is fraudulent (1) or legitimate (0). The script loads this file into a Pandas DataFrame using pd.read\_csv(). In the preprocessing step, the Class column is separated as the target variable y, while the remaining columns form the feature matrix X. Because the features are on different scales, standardization is performed using StandardScaler, which transforms the data to have a mean of zero and unit variance. This ensures that all features contribute equally during model training.

Next, the AutoEncoder model is initialized and trained. AutoEncoders are neural networks that attempt to reconstruct their input; higher reconstruction errors typically indicate anomalies. In this assignment, the AutoEncoder is instantiated with default parameters using model = AutoEncoder(verbose=1), which enables logging during training. The model is then trained on the standardized dataset using the .fit() method. Since this is an unsupervised learning task, the AutoEncoder learns to reconstruct typical (non-fraudulent) transactions. Fraudulent transactions, which differ significantly in structure, will have higher reconstruction errors and thus be flagged as outliers.

After the model is trained, predictions are generated. The predicted labels are obtained using model.labels\_, where a value of 0 denotes an inlier (normal transaction), and 1 denotes an outlier (potential fraud). Additionally, model.decision\_scores\_ provides the raw outlier scores, which are useful for visualization. The script evaluates the model's performance using a confusion matrix and a classification report generated by confusion\_matrix and classification\_report, respectively. These metrics help measure how well the model distinguishes between normal and fraudulent transactions. The ROC-AUC score, calculated via roc\_auc\_score, further assesses the model’s ability to rank observations correctly.

The final section involves visualizing the reconstruction error distribution. A histogram of the y\_scores is plotted using Matplotlib, providing a visual insight into how well the model separates normal and abnormal transactions. The histogram helps in identifying thresholds for outlier detection. The resulting figure is saved as reconstruction\_error.png for documentation or report purposes. Finally, the script prints a confirmation message indicating that the experiment has been successfully completed. This modular, well-structured approach enables efficient experimentation with fraud detection using deep learning and can be easily extended or fine-tuned for other anomaly detection tasks.



