Distinction Task 10.1D: Model evaluation metrics

- **An analysis of a fraud detection model utilizing decision trees and K-Nearest Neighbors (KNN)**
  Detecting fraudulent credit card transactions from a data set is the goal of this Python-based machine learning project, which makes use of the supervised learning models, KNN and Decision Trees. Class labels that aid in determining whether a transaction is fraudulent and normalized transacted value are among the variables in the dataset, along with raw values for dependent and independent variables. This study does a good job of mitigating data distortion, especially when it comes to a low percentage of fraudulent transactions.

- **Data Preprocessing**
  The dataset's Amount feature set is chosen from the previously downloaded CSV file. The StandardScaler from sklearn is then used to normalize this feature set. Finally, the features, Time and Amount, are removed. The data has a significant class imbalance because the percentage of fraudulent transactions makes up a very small component of the data. Under sampling, which lowers the percentage of honest transactions to match the proportion of fraudulent ones, is used to solve the issue. Consequently, a balanced under sampled dataset is produced, and it is further split into training and testing data sets, which have a 7:3 ratio, respectively.

- **K-Nearest neighbors (KNN)**
  Grid Search CV is used to train the KNN model on an undersampled dataset in order to improve the hyper parameter n_neighbors. The value of n_neighbors is tested between 1 and 5. The goal of cross-validation is to estimate the optimal value of "n_neighbors," and the recall is the next parameter used to select this value. The recall measure is highlighted because false negatives, or missed fraudulent cases, are among the most important components in fraud analysis.

  The test set's confusion matrix is displayed, and the recall value is computed, to assess the model's performance. The recall score for the program's capacity to spot fraudulent transactions serves as evidence of this. Ultimately, the complete dataset is run through the KNN model to equitably evaluate the algorithm's performance on the undersampled data.

- **Threshold Tuning**
  Better model outcomes are achieved by fine-tuning the category determination thresholds. The probabilities calculated by the KNN classifier for different thresholds (between 0. 1 and 0. 9) are computed at each stage of the procedure, along with confusion matrices and recall metrics. The goal of this stage is to enable finding a good balance by determining the optimal recall level and precision level.

- **Decision Tree Classifier**
  In addition to KNN classifiers, Decision Tree classifiers are also implemented. Tuning the Decision Tree's characteristic max_depth, whose values vary from 3 to 7, is what cross-validation entails. Once more, the model is tested using cross-validation, with recall being the primary focus this time. KNN is the format utilized, and accuracy and recall are given for both the full and under sample data sets. One can determine whether model is more effective at identifying fraudulent transactions by comparing the Decision Tree and KNN models in the following comparison.

- **Model Assessment**
  Therefore, while evaluating both models, the recall and confusion matrix outcomes are used, with a focus on the quantity of false negatives. The ability to see how the models' accuracy and recall are swapped at various thresholds is one of the benefits of the charting method, which aids in the models' further improvement.

- **Conclusion**
  This paper describes how to detect fraud transactions in unbalanced datasets using KNN and Decision Tree classifiers, based on the research. One of the key evaluation metrics is the confusion matrix, and recall is taken into account while adjusting the models. To enhance fraud detection, the models can be further refined by employing under sampling to reduce data imbalances and adjusting the bounds of classification.