# Implementing Machine Learning Models using Python

# Introduction

This presented the **bill\_authentication.csv** dataset on Python implementation of machine learning models inclusive of Data Preprocessing, Classification Models, and Clustering Models. It is consistent with learning outcomes addressing data types, elements of predictive analytics and project life-cycle when solving analytical problems.

## Learning Outcomes

* LO1: A background check on data types and the usual preprocessing procedures that data scientists employ on data.
* LO2: Understanding of what predictive analytical elements are such as regression, classification and clustering.
* LO3: Prediction in business and analysis: application of a machine learning approach.

# Task 1: Load and Clean the Dataset

The bills authentication dataset that was used in this analysis is very small and was downloaded from Drive. Subsequently, it is cleaned in order to condition the data for analysis.

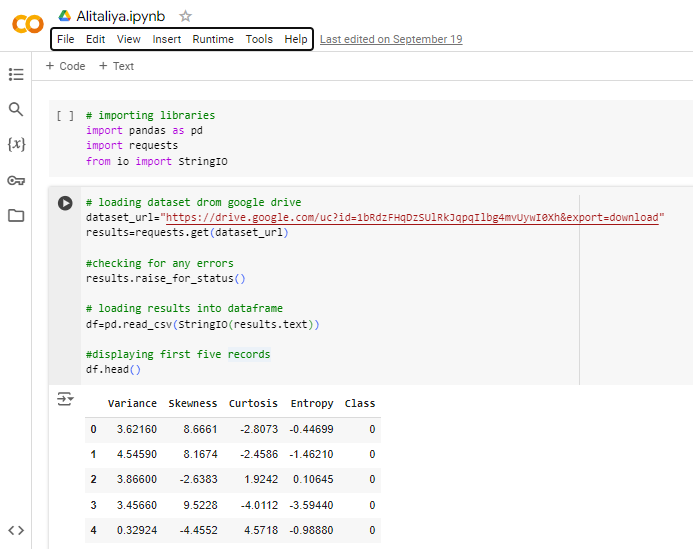


Figure : Jupyter Notebook Code: Loading and Inspecting Dataset

## Data Cleaning Steps

* Missing Values: Alternatively, in case of any missing values in the dataset, they are deleted.
* Outlier Detection: In this task, Outliers are identified based on IQR method.
* Outlier Treatment: Outliers are treated whereby any number beyond the IQR method and the Z-score method is either rejected or converted.
* Encoding Categorical Variables: While Nominal variables are encoded with One-Hot Encoding and Label Encoding.
* Scaling Numerical Data: For the scaling of numerical features use StandardScaler and MinMaxScaler is used.

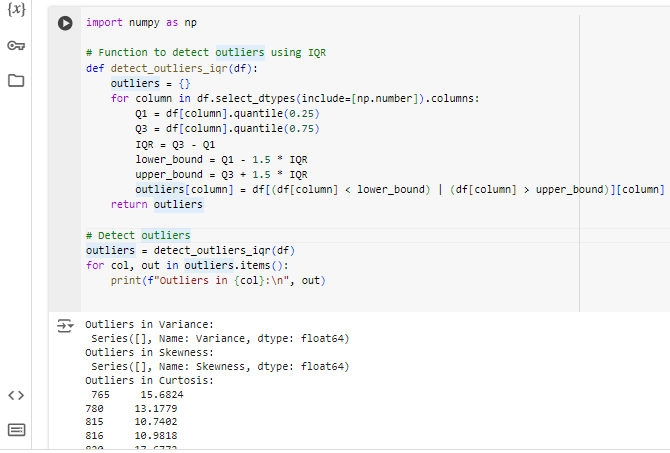


Figure : Data Preprocessing: Outlier Detection

# Task 2: Classification Problem using Decision Trees

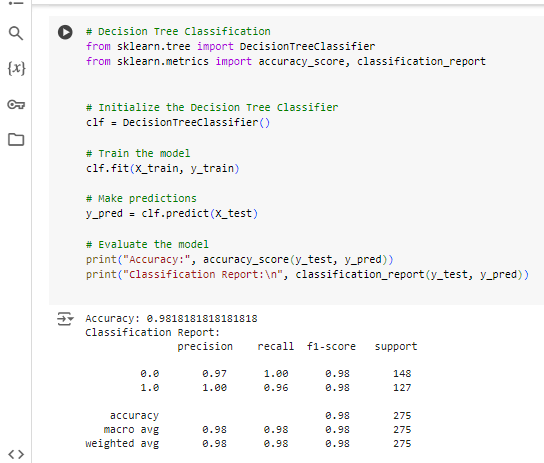
In this research, Decision Tree Classifier is applied to the dataset to create classification models. Decision trees are easily understandable models, which make predictions based on certain features values, and which, as well, can capture a non-linearity between features.

Figure : Machine Learning Model: Decision Tree

# Task 3: Clustering Problem using K-Means

The K-Means clustering algorithm is used on the dataset as a method of clustering in its own right without referencing any prior sorting by the analyst or the computer. Here, the data is divided into various clusters depending on similarities in features. This can be particularly beneficial in identifying certain trends that may not easily be identifiable by applying some of the supervised learning approaches.

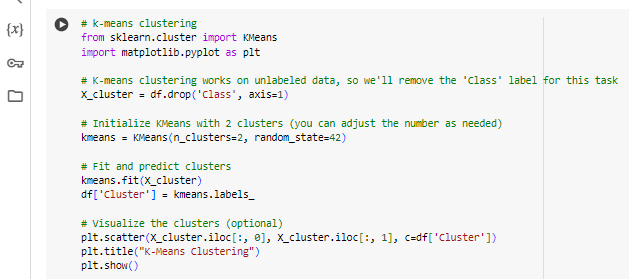


Figure : Jupyter Notebook Code: K-Means Clustering

# Task 4: Evaluating the Classification Algorithm

Evaluation of the classification model is based on the confusion matrix and ROC AUC score is used. It is crucial to assess one’s models to know their valuable input and output since without such findings it is impossible to fix the model’s mistakes, improving the rate of accuracy.

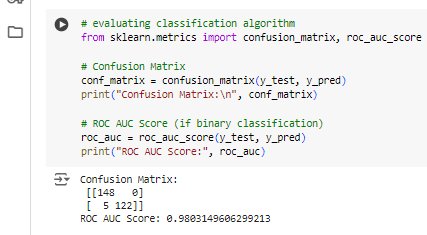


Figure : Jupyter Notebook Code: Evaluating Classification Algorithm

# Task 5: Evaluating a Linear Regression Algorithm

A Linear Regression model is used for evaluating the regression performance. This model is used for continuously dependent variable predictions with respect to the features and is used for business forecasting and risks analysis.

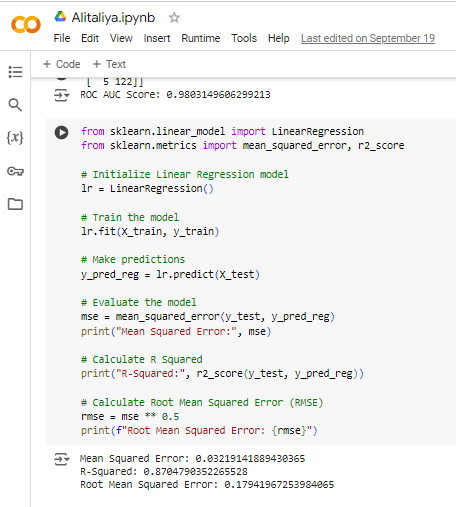


Figure : Jupyter Notebook Code: Linear Regression Model Training and Evaluation

# Conclusion

Here, the report outlines the use of different machine learning models in a python coded environment. The data loading, data cleaning as well as model evaluation were in the context of understanding the learning outcomes with specific tasks. The report demonstrates knowledge of data management and forecasting to a great extent.

# Appendix

## Python Libraries Used

* Pandas
* Requests
* Numpy
* Scikit-learn
* Matplotlib

## Key Functions and Methods

* **train\_test\_split():** Divides the database into the training data and the test data since the main scope of this phase is to evaluate the model.
* **DecisionTreeClassifier():** Used to initialize a decision tree classifier for classification based on problems.
* **KMeans():** Uses a clustering algorithm, namely K-means for data clustering with a view of categorizing data into separate clusters.
* **Mean\_squared\_error():** Estimates the variance of the prediction residuals, which in return measures the goodness of fit of the regression models.

## Colab Code Implementation

You can get the full code showing how this project was completed including data loading/visualization, data cleaning/preparation and both classification and regression models from the Google Colab notebook provided.

[View the Colab Notebook](https://colab.research.google.com/drive/1NVvy4ughSnWD5HRe_FXBQR-GItwmvcup?usp=sharing#scrollTo=h0CyvE3UxTa1)