

Project by,

RICHU RAJU

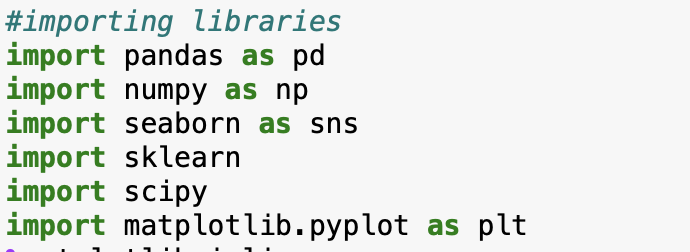
**1. Introduction**

This Project aims at creating a Credit Card Fraud detection Model using machine learning techniques. The dataset provided contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions. It contains only numerical input variables which are the result of a PCA transformation. Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependant cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise. I have used 3 techiques to train the model – Linear Regression, Decision tree classifier, Random Forest classifier. The Python codes were executed in Jupyter Notebook.

**2.Project Explanation**

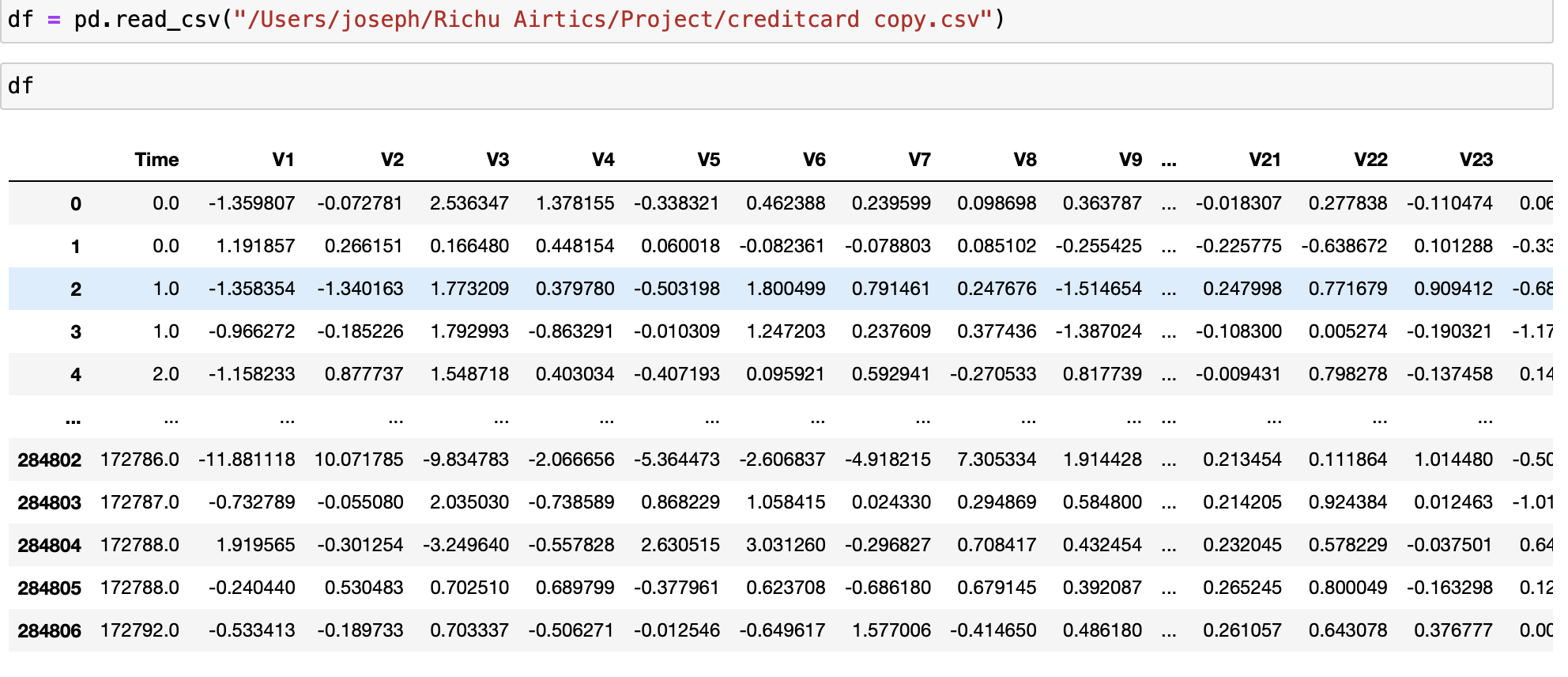
**1. Importing the libraries**

Important Python libraries like Numpy, Pandas, Matplotlib etc are imported.



**2. Loading the Dataset**

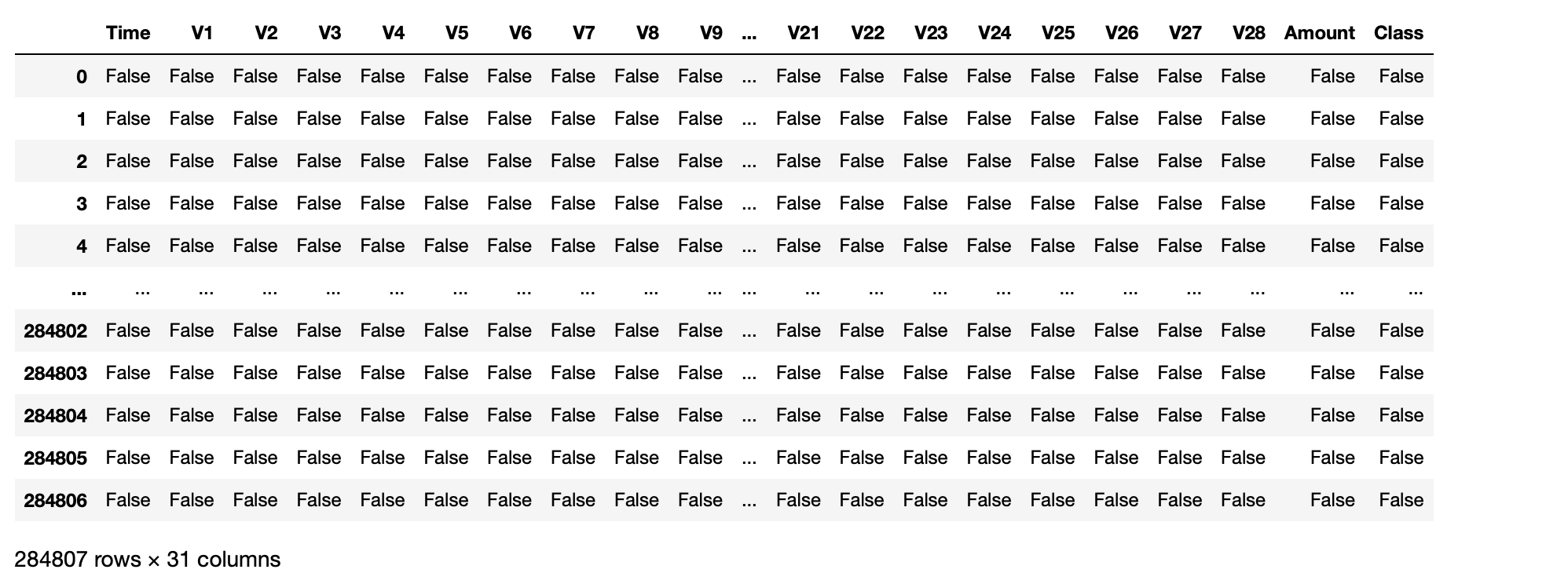
The dataset is loaded to Dataframe ‘df’ and is displayed.



The dataset contains 284,807 rows and 31 columns.

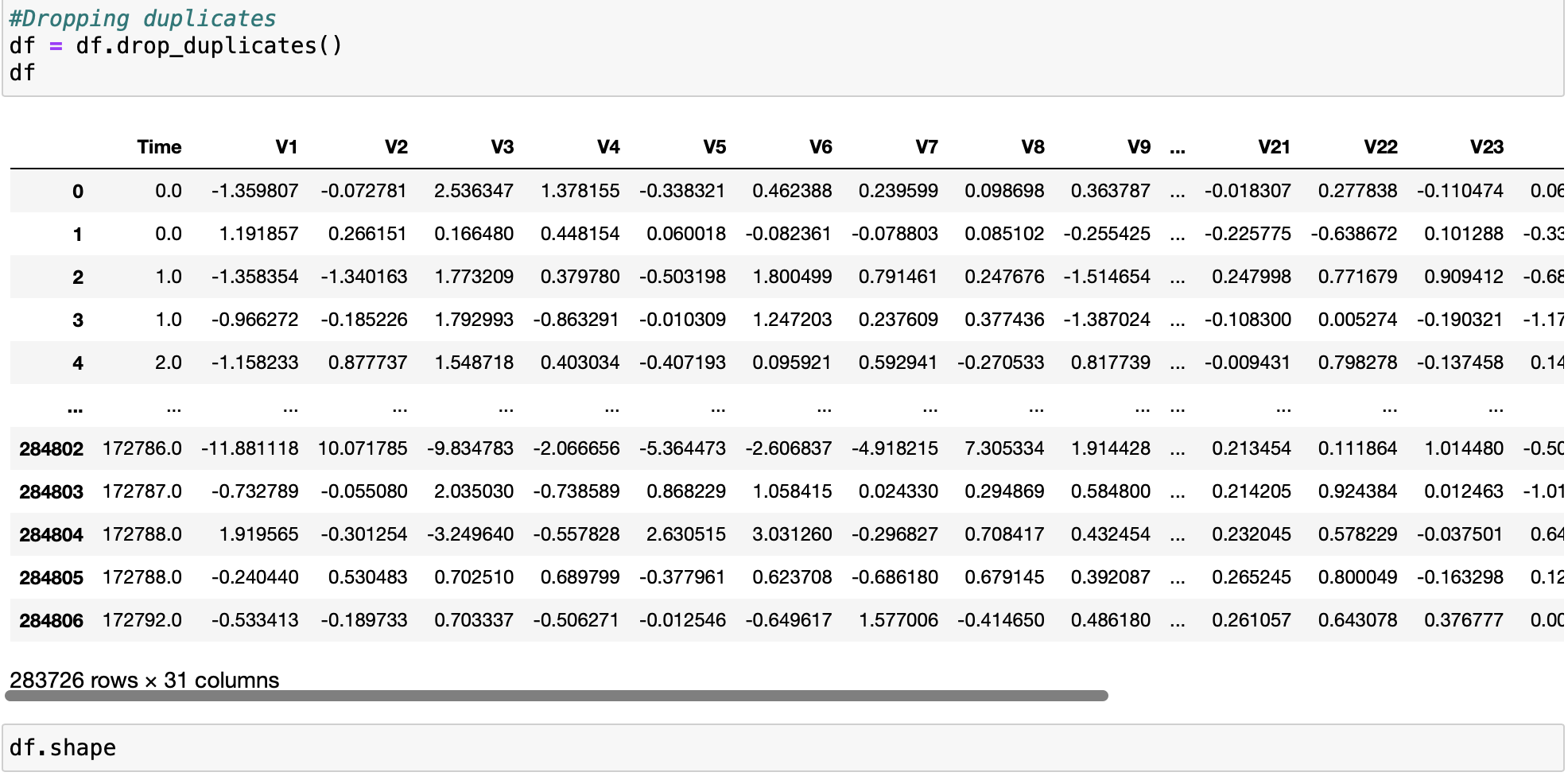
**3.Data Cleaning and Preprocessing**

In this step, we check if the Dataset contains null values, duplicates. Null values were not found. Some duplicate entries were found and are removed. Also, we drops columns which are not necessary for the training purpose.



Graphical user interface, text, application, email

Description automatically generated



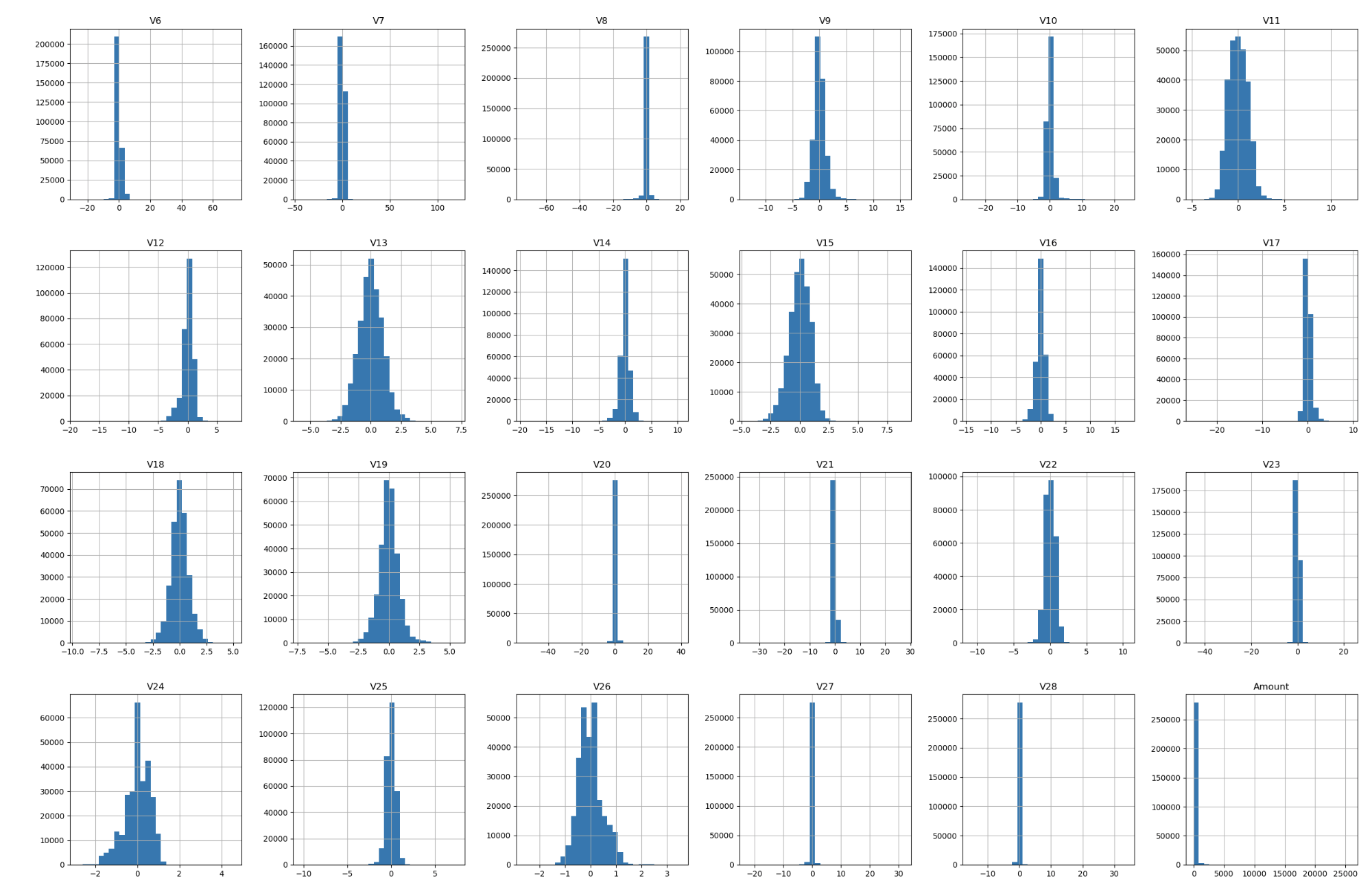
Uneven distribution is being observed for the ‘Class. 473 fraud transactions were found.

I have plotted the Normal and fraud transactions after checking the Class values. Value 0 represents legit transactions and 1 represents fraud transactions.

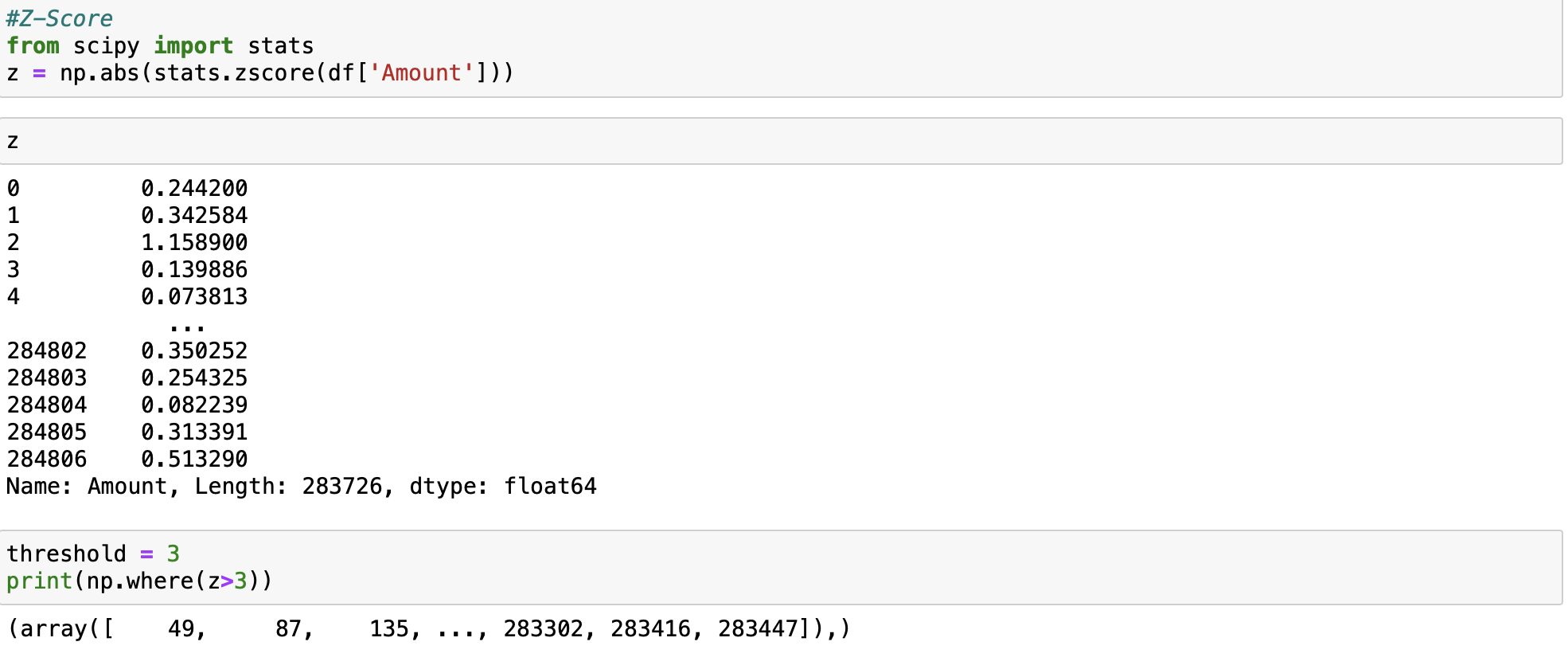
Chart

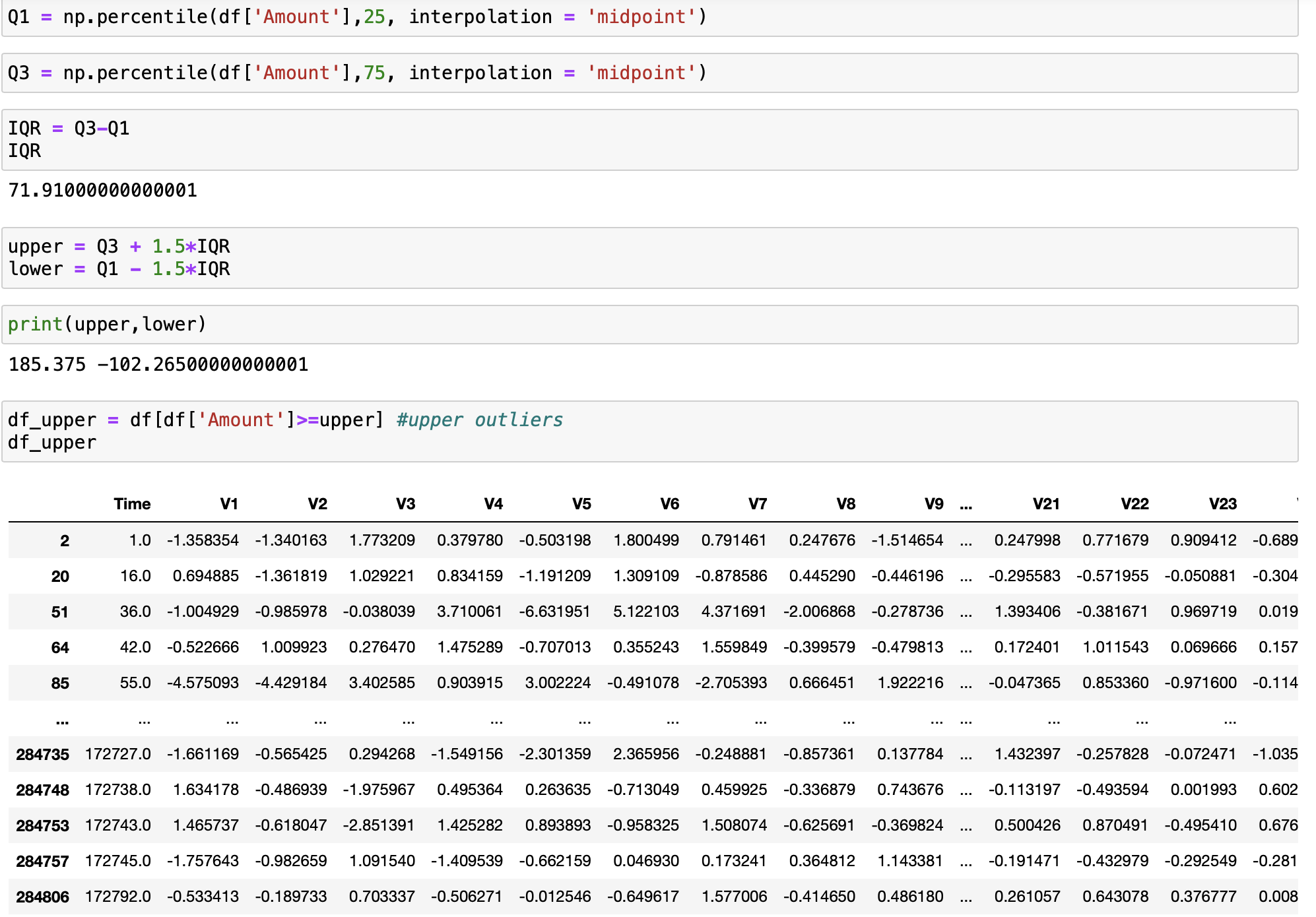
Description automatically generated

Also, I have plotted distplots and histplots for different values of the dataset like Time, Amount etc.

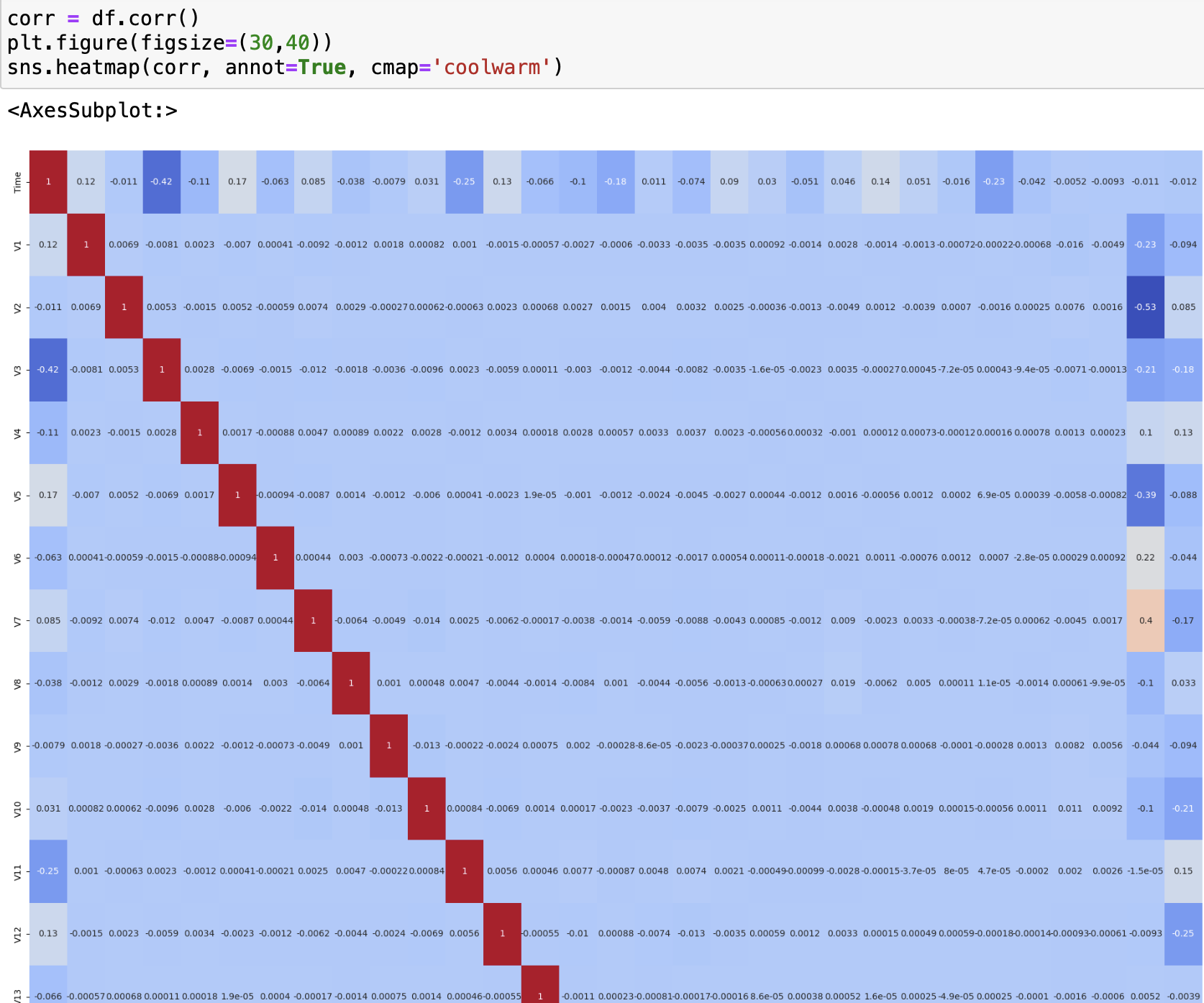


Z-scores and Interquartile range is being calculated.



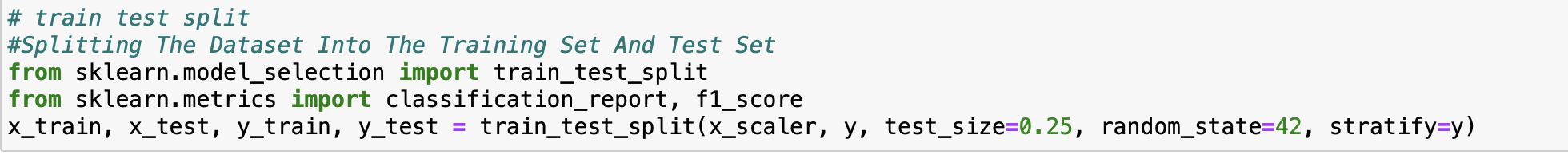


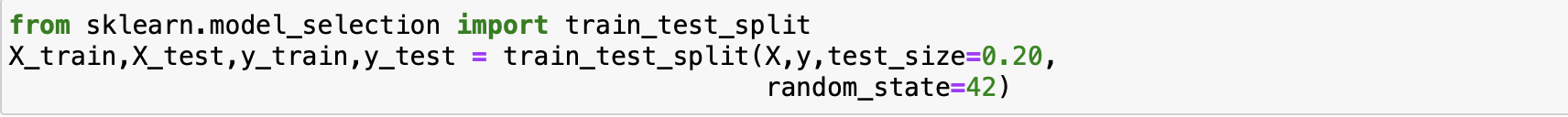
Plotting Correlation matrix



**4.Training the model**

**Splitting the Test and Train data.**





**5.Handling the Imbalanced Data**

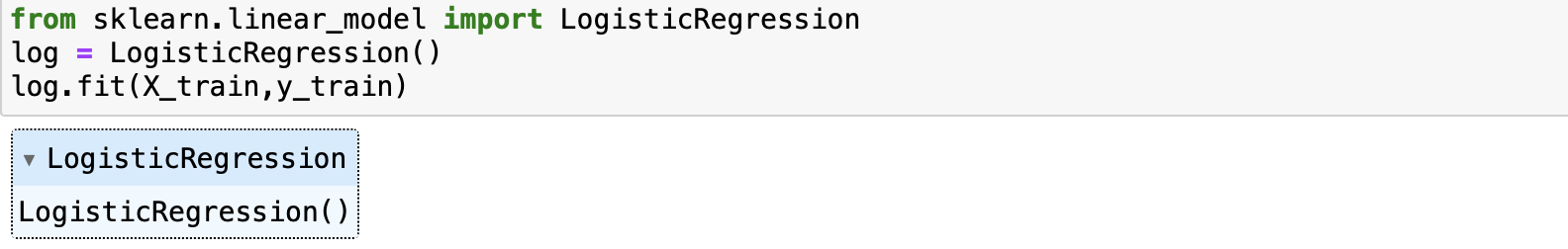
**Undersampling**

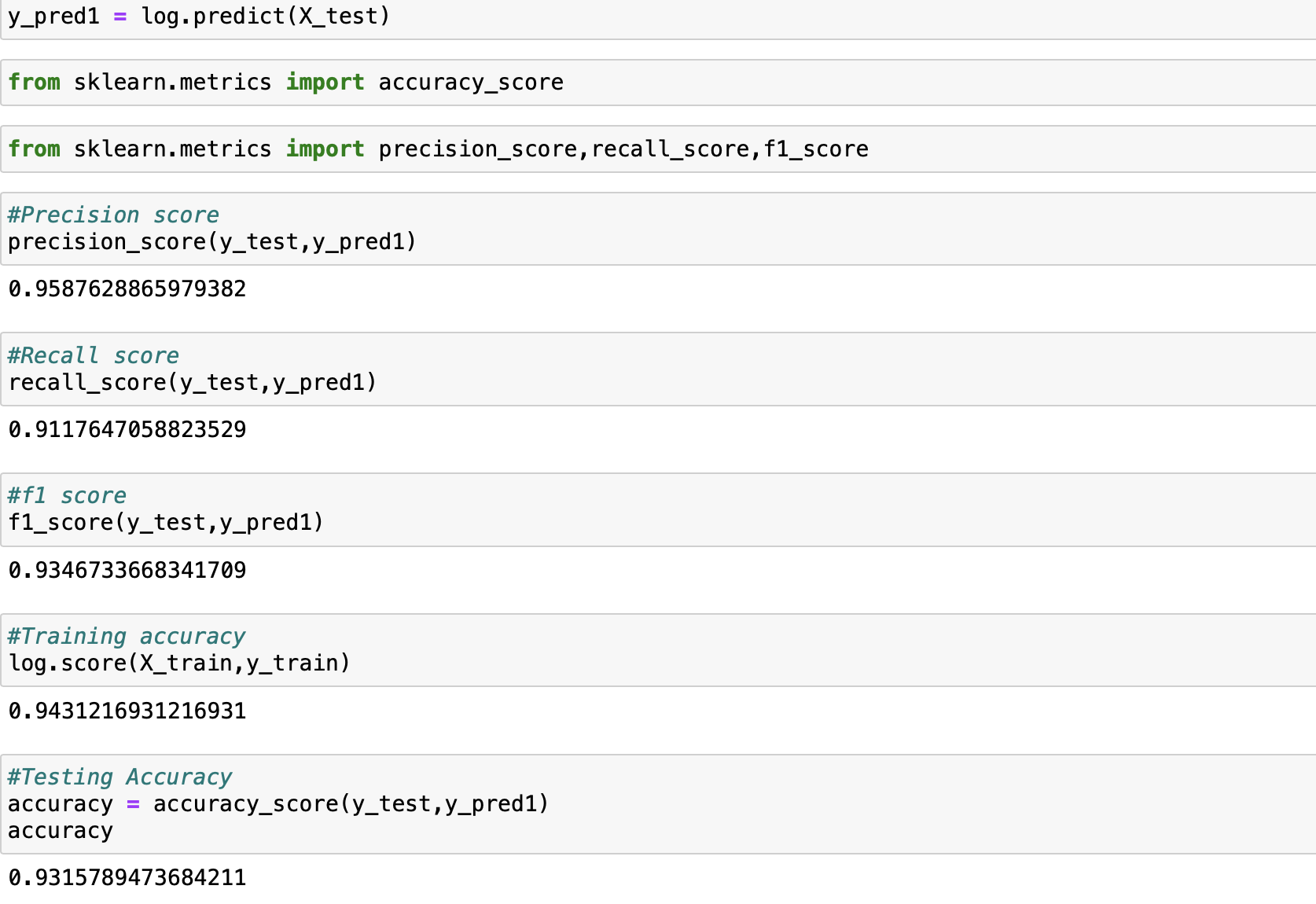
In undersampling, we randomly delete rows form majority class to match them with minority class.



**Logical Regression**

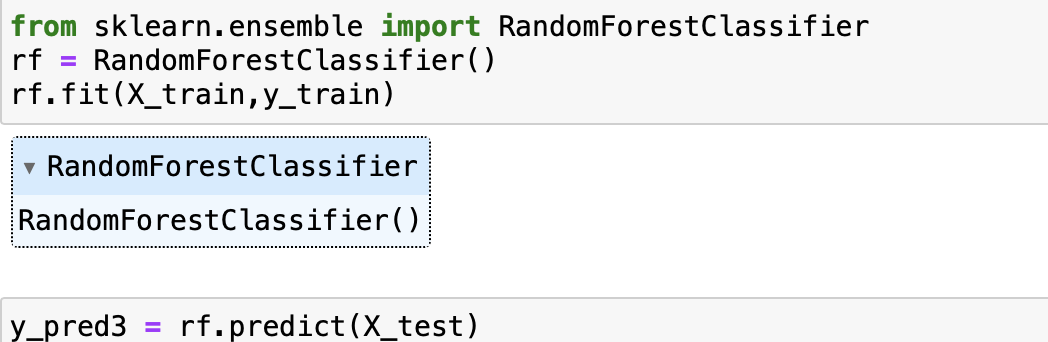
The Testing Accuracy, Training Accuracy, Precision Accuracy, f1 score and recall score has been calculated.

****

****

**Random Forest Classifier**

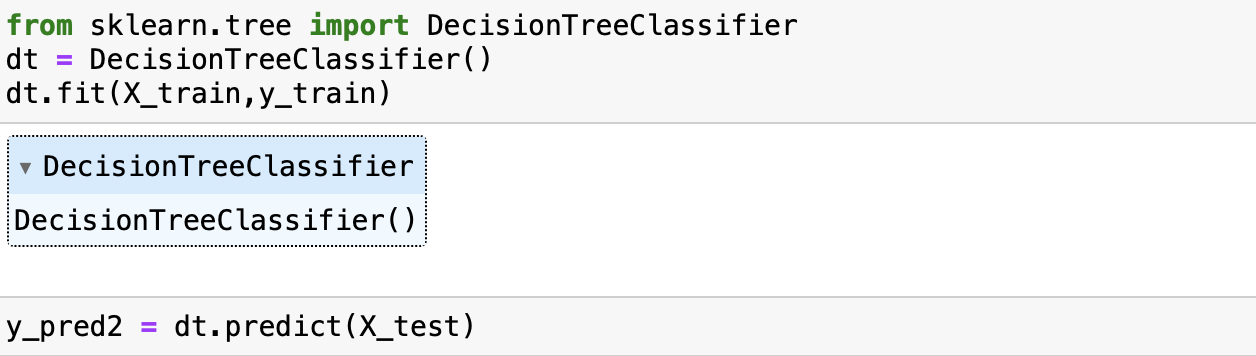
The Testing Accuracy, Training Accuracy, Precision Accuracy, f1 score and recall score has been calculated.

****

****

**Decision tree Classifier**

The Testing Accuracy, Training Accuracy, Precision Accuracy, f1 score and recall score has been calculated.

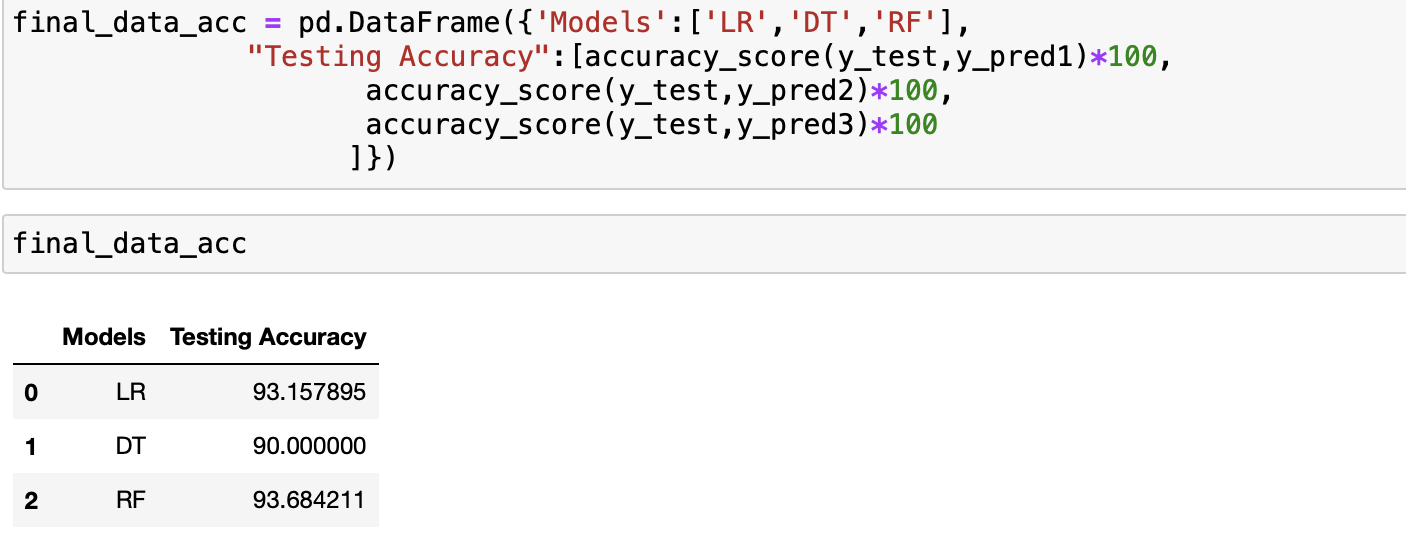


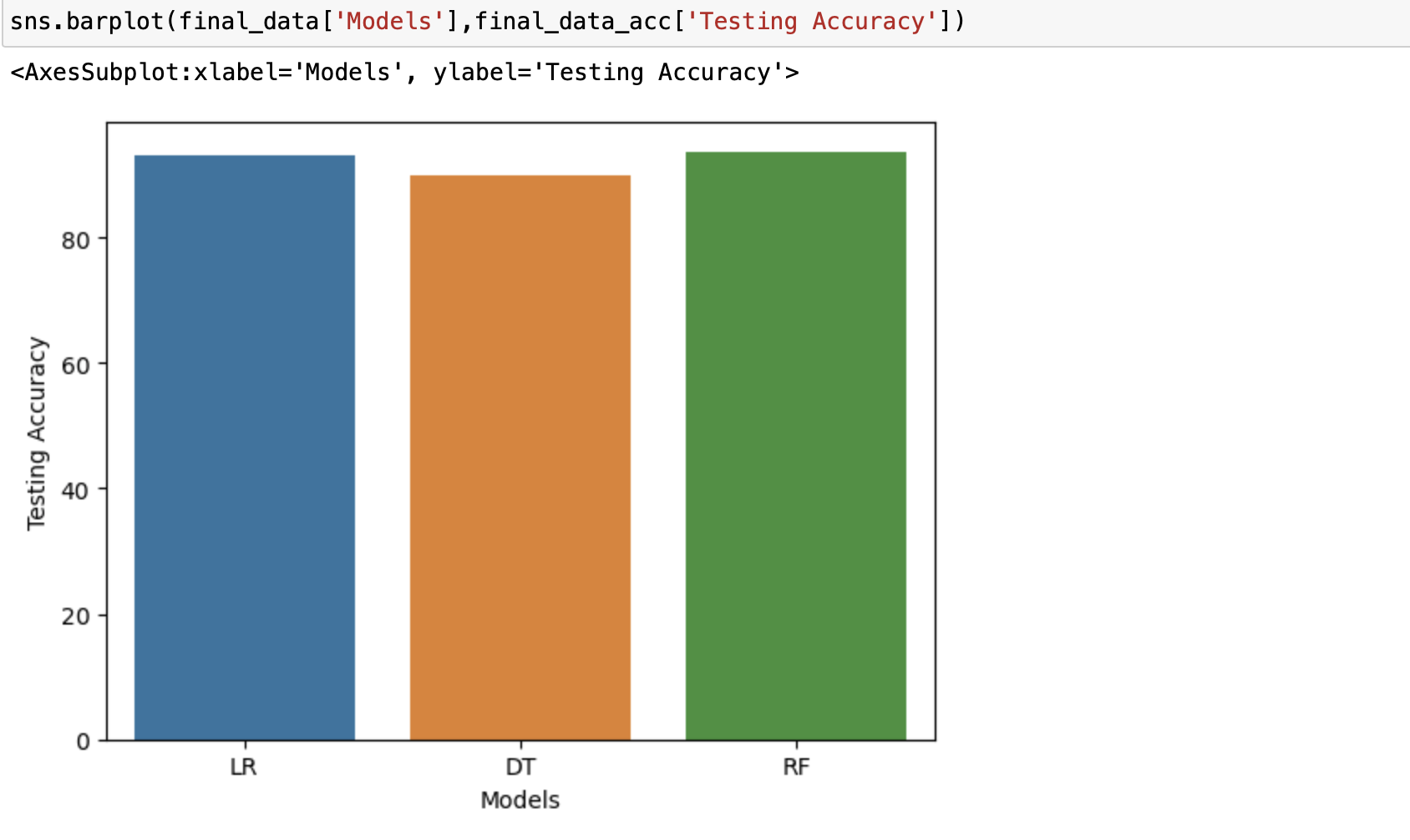
**Graphical user interface, text, application, email

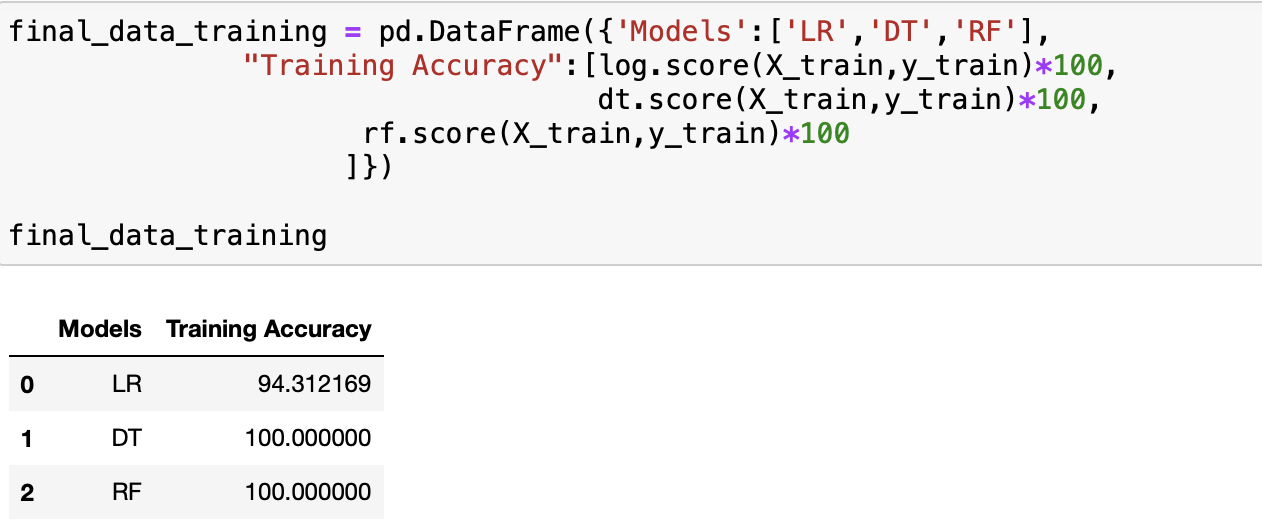
Description automatically generated**

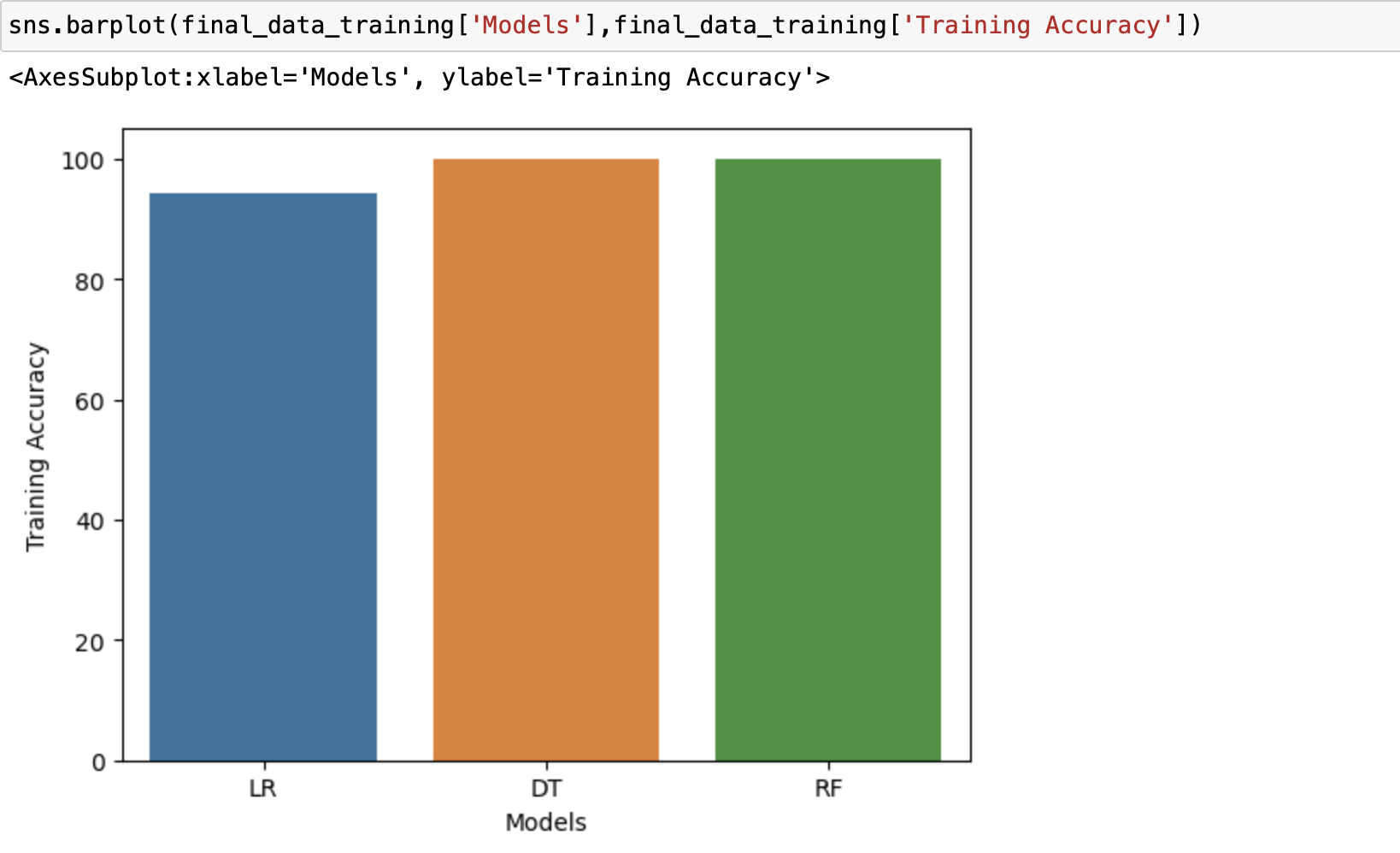
**Comparing and evaluating different models**

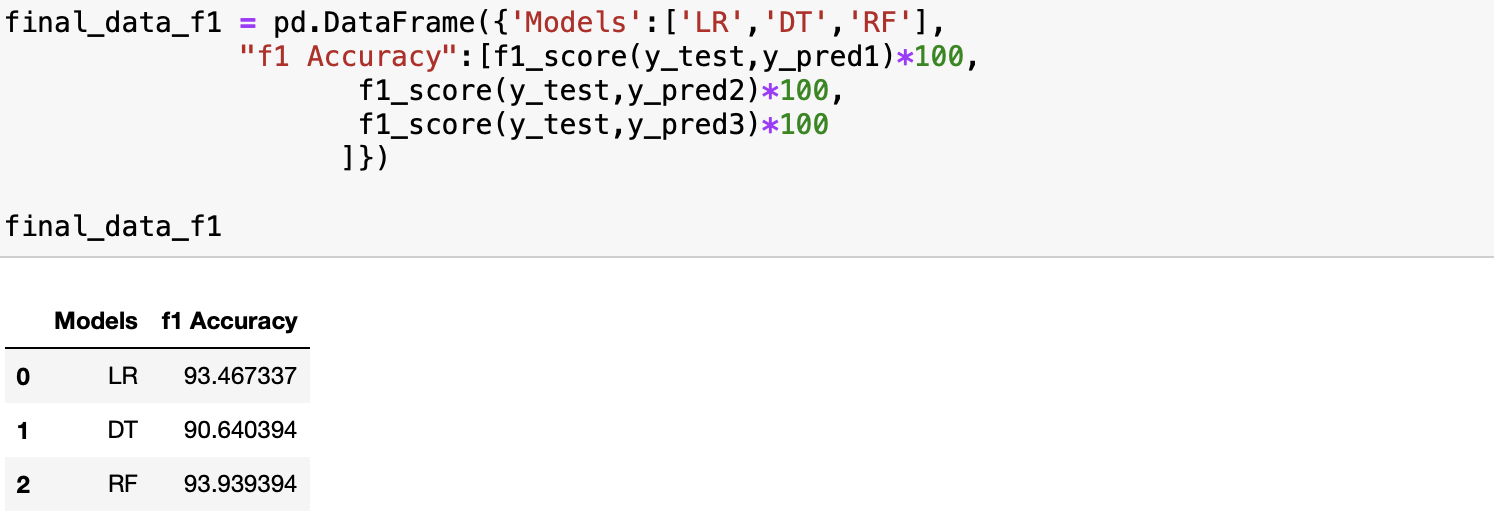
The accuracy scores calculated using the above 3 training models are being plotted.



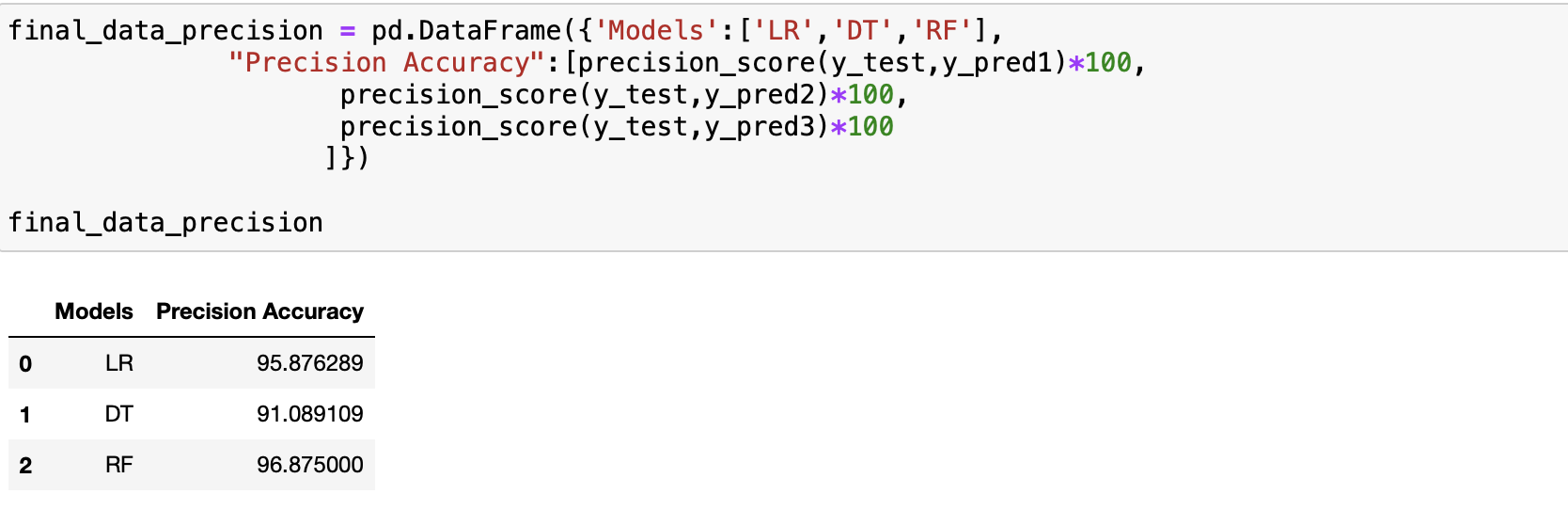






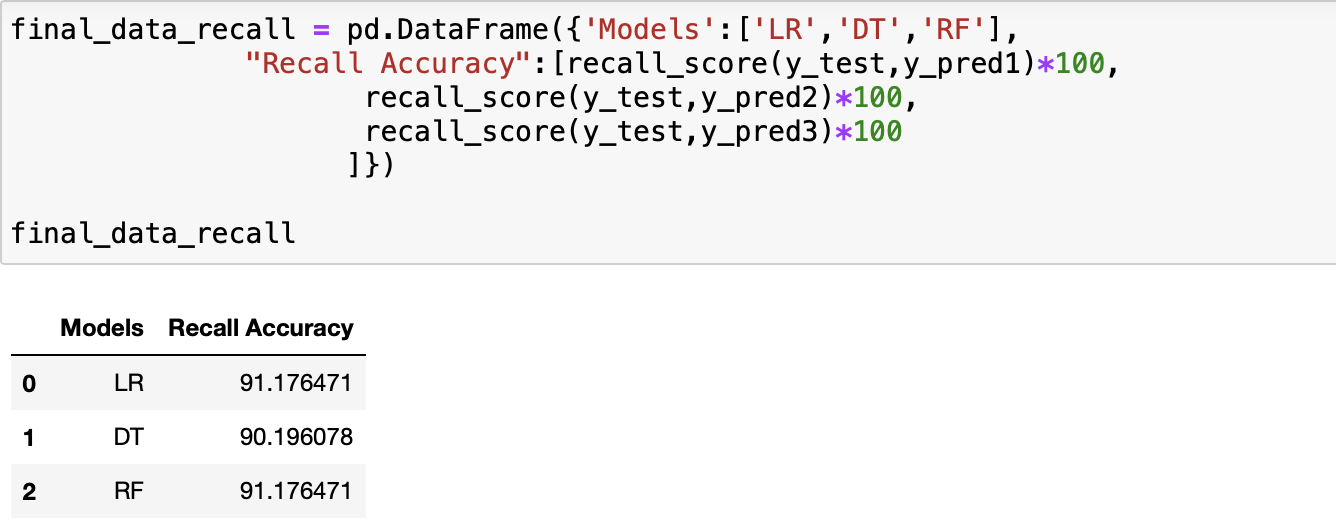
****

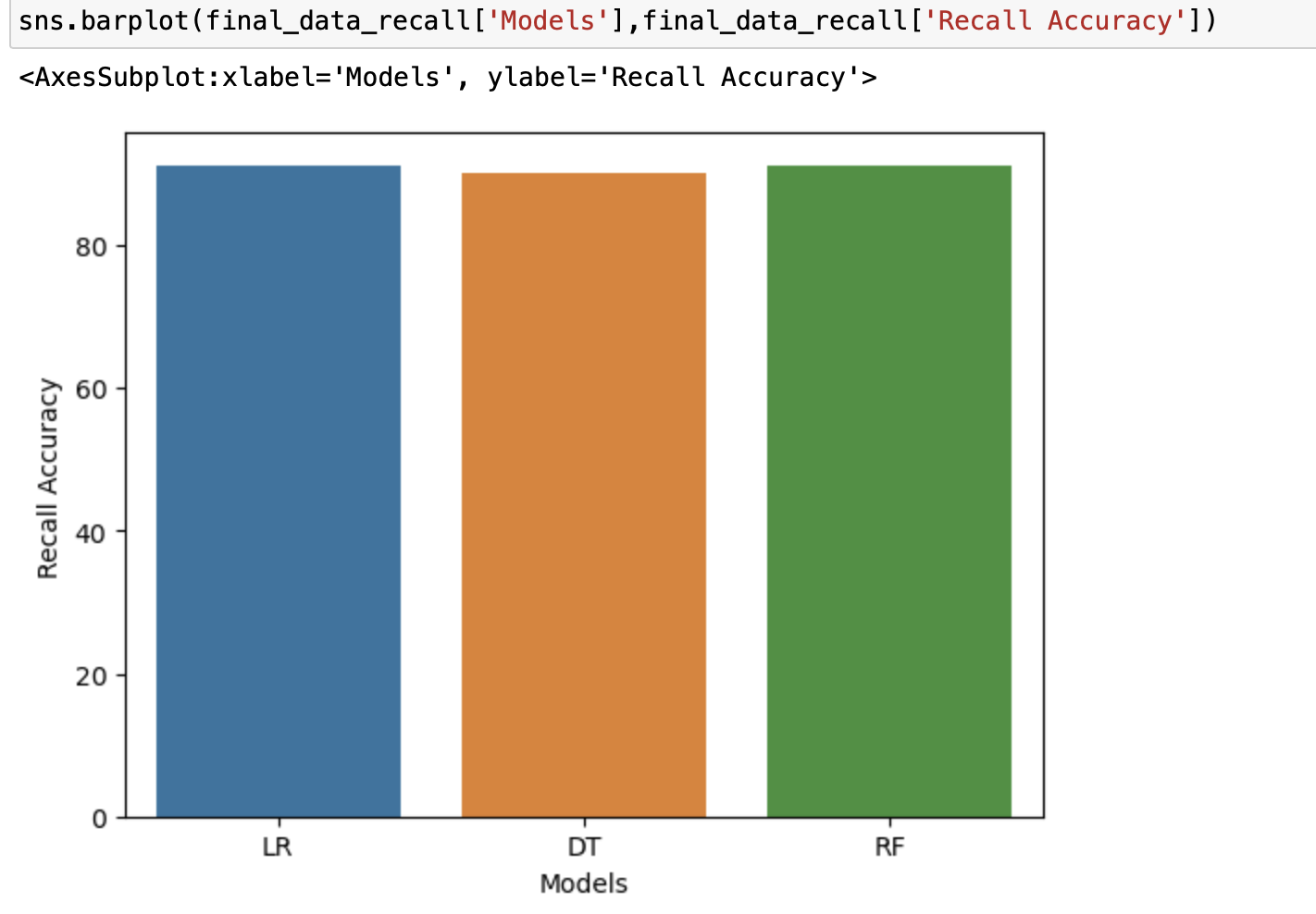
****

****

**Chart, bar chart

Description automatically generated**

****

****

**Oversampling**

We do the Oversampling using SMOTE- Synthetic Minority Oversampling Technique. We use minority data examples by replicating them. Rather creating duplicates, in SMOTE it creates synthetic data points from original data.



**Logical Regression**

**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface, text, application, email

Description automatically generated**

**Decision Tree Classifier**

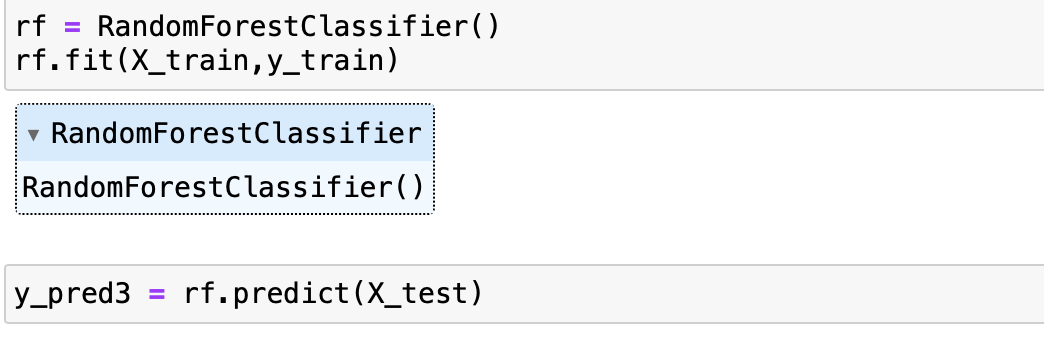
**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface, text, application, email

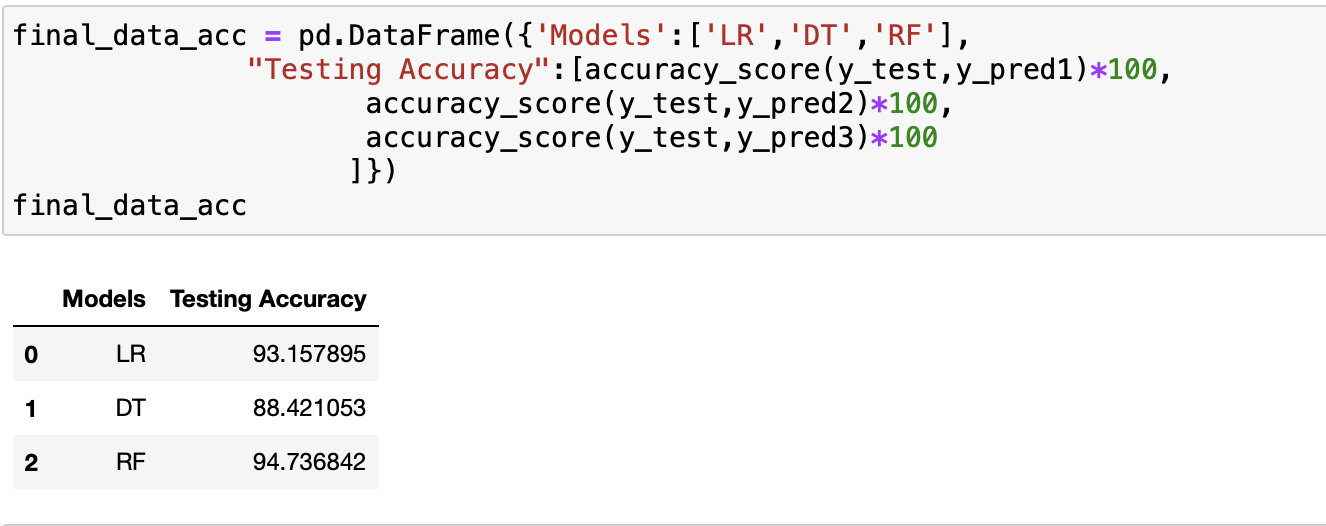
Description automatically generated**

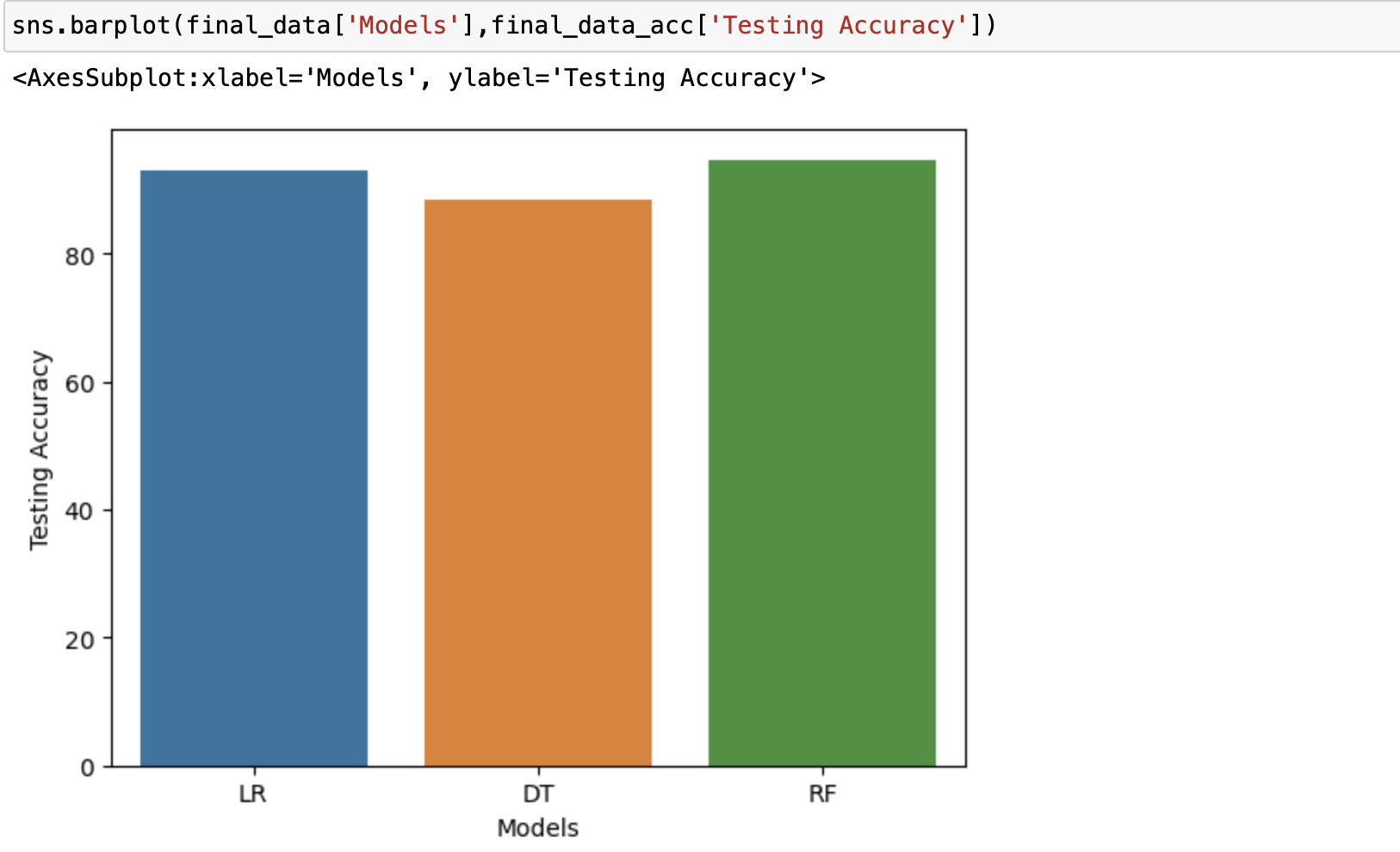
**Random Forest Classifier**

****

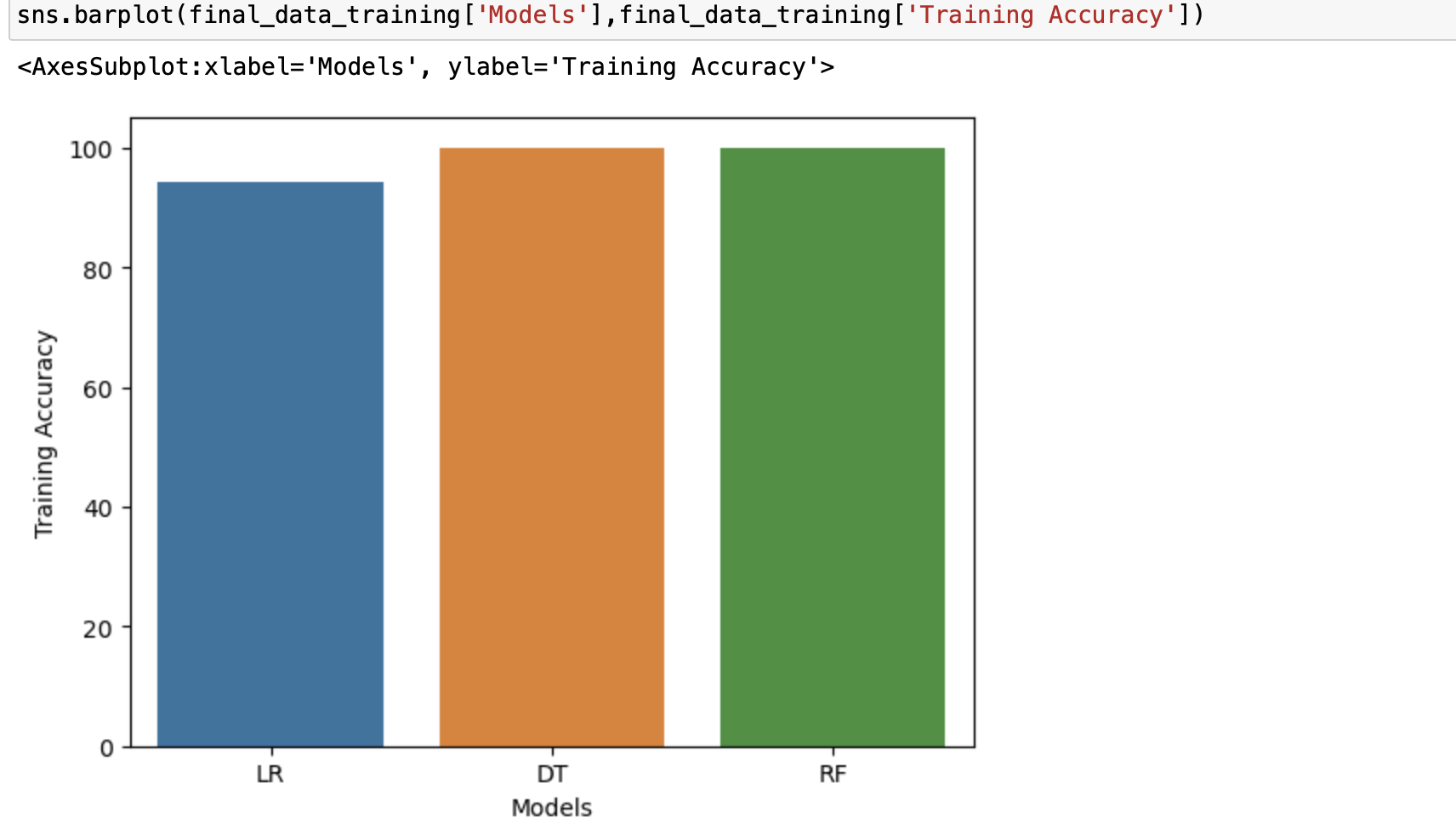
****

**Comparing and evaluating different models**

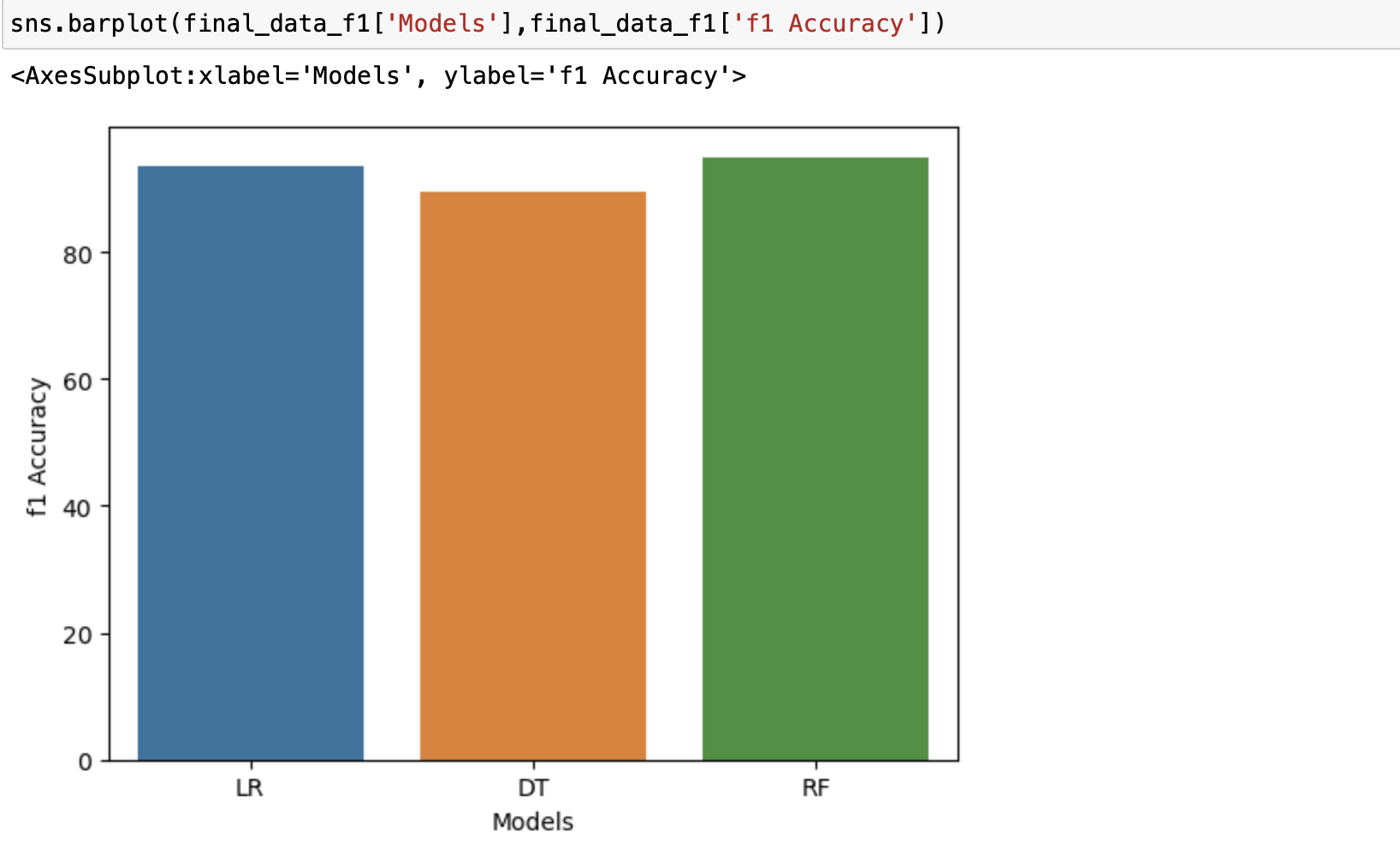
****

****

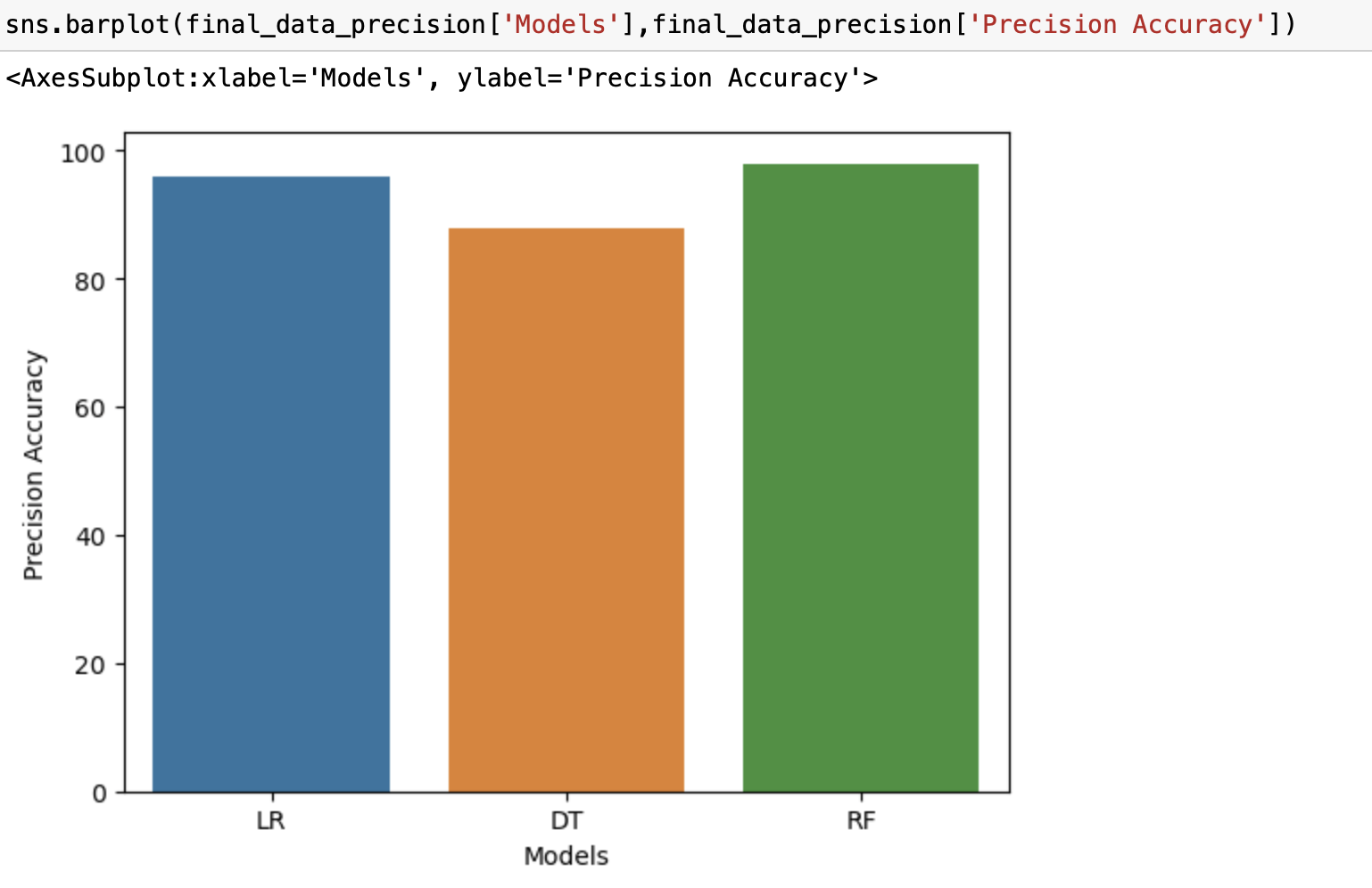
****

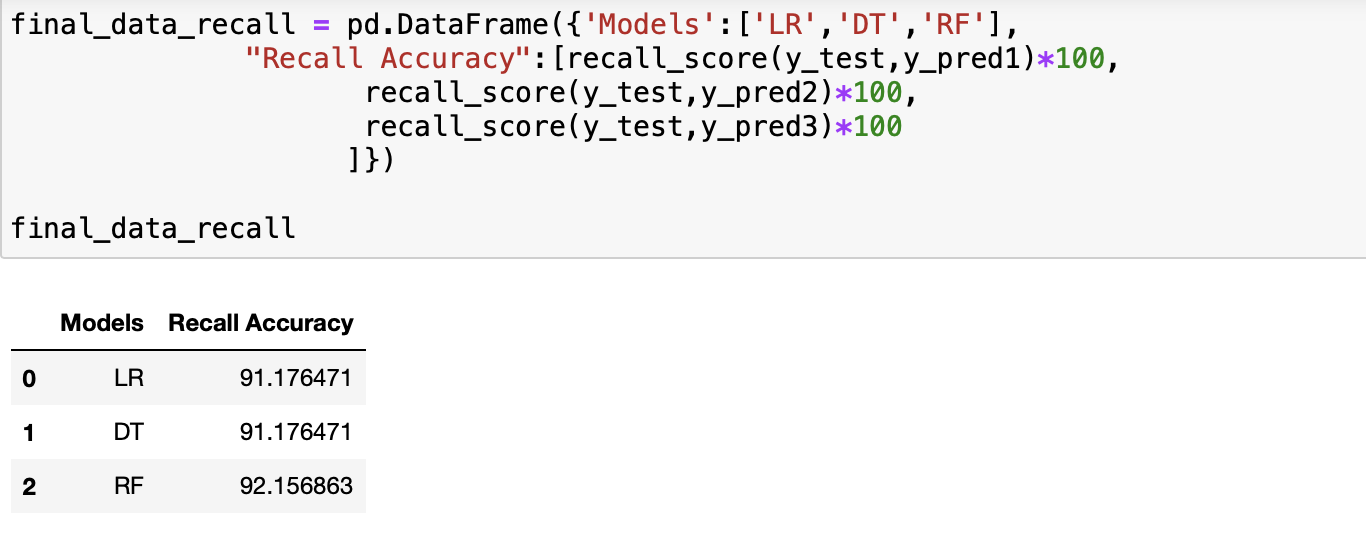
****

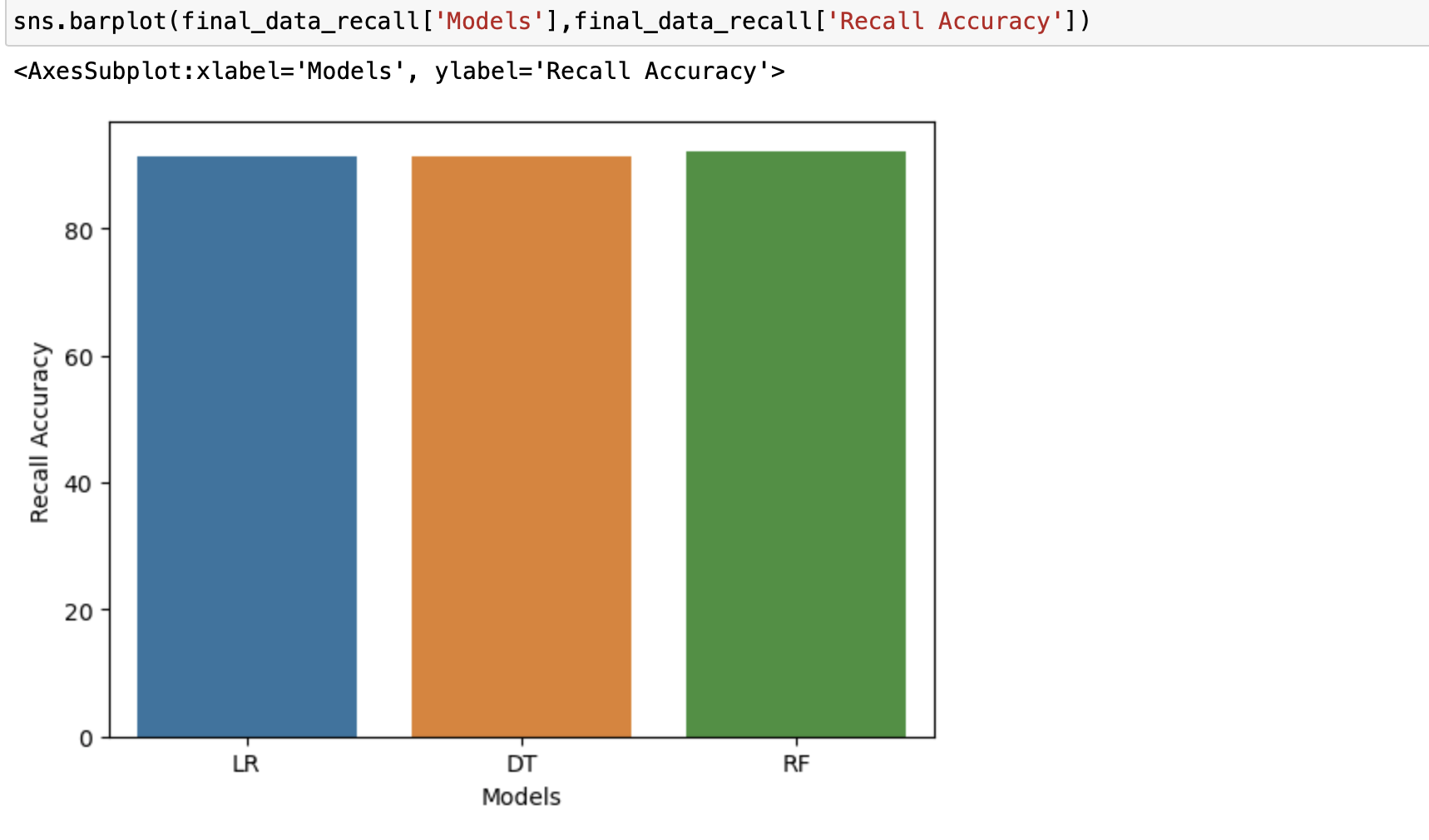
****

****

****





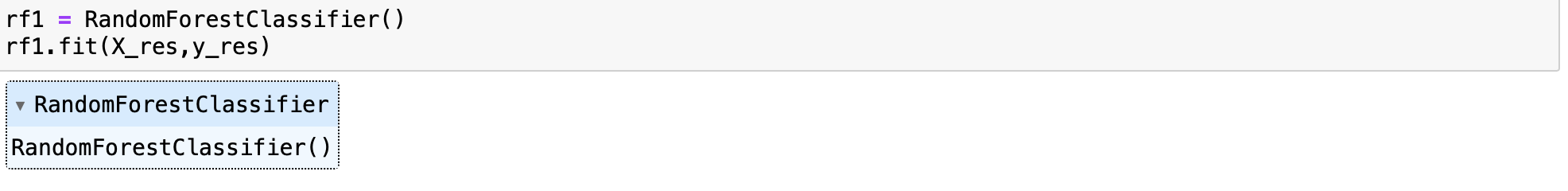


**6. Conclusion**

After comparison between the 3 models - Logical Regression, Decision tree classification and Random Forest classification, it has been found that Random Forest Regression will be the best fit for the model.

**7. Saving the model**

We train the entire model with Random forest classifier.

****

Now using joblib library we predict model.

