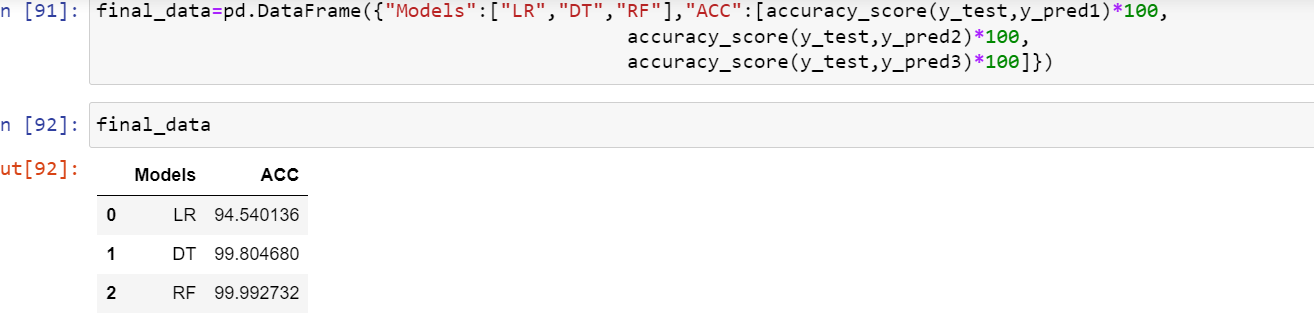


|  |
| --- |
| CREDIT CARD FRAUD DETECTION TITLE  2023 |
|  |
| February 9  Authored by: Nisha Nandal |

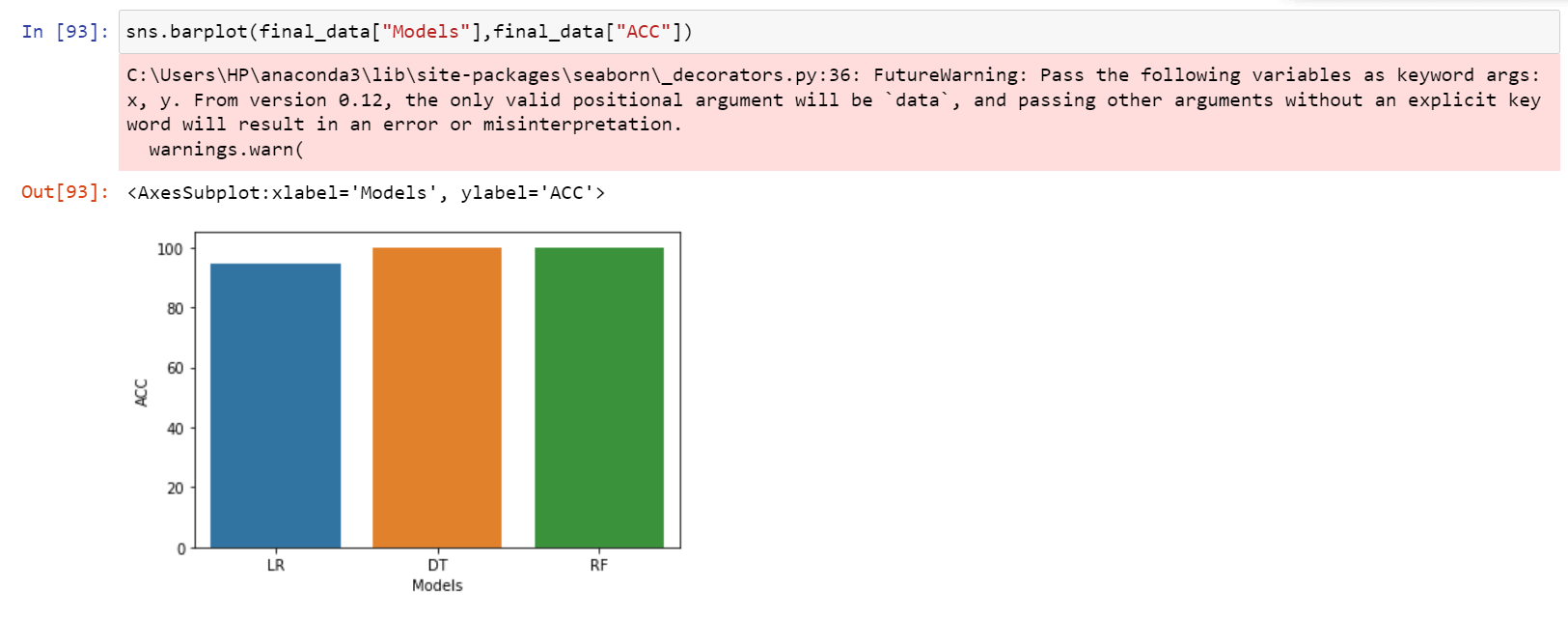
# Credit Card Fraud Detection

|  |
| --- |
| It is important that credit card companies are able to recognize fraudulent credit card transactions so that customers are not charged for items that they did not purchase.  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) accounts for 0.172% of all transactions.  It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, we cannot provide the original features and more background information about the data. Features V1, V2, … V28 is 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-dependent cost-sensitive learning. Feature 'Class' is the response variable and it takes the value 1 in case of fraud and 0 otherwise. |
| <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud> |
|  |
| The challenge is to recognize fraudulent credit card transactions so that the customers of credit card companies are not charged for items that they did not purchase.  The main challenges involved in credit card fraud detection are:   1. Enormous Data is processed every day and the model build must be fast enough to respond to the scam in time. 2. Imbalanced Data i.e most of the transactions *(99.8%)* are not fraudulent which makes it really hard for detecting the fraudulent ones 3. Data availability as the data is mostly private. 4. Misclassified Data can be another major issue, as not every fraudulent transaction is caught and reported. 5. Adaptive techniques used against the model by the scammers.   How to tackle these challenges?   1. The model used must be simple and fast enough to detect the anomaly and 2. classify it as a fraudulent transaction as quickly as possible. 3. Imbalance can be dealt with by properly using some methods which we will talk about in the next paragraph 4. For protecting the privacy of the user the dimensionality of the data can be reduced. A more trustworthy source must be taken that double-checks the data, at least for training the model. We can make the model simple and interpretable so that when the scammer adapts to it with just some tweaks we can have a new model up and running to deploy.   Before going to the code it is requested to work on a jupyter notebook.  Code:  # Import all the necessary packages:    #Loading the data:    #Understanding the data:    **#Describing the data:** |
| #checking null values in data:  No null values found in the data |
| # Standardization of data:    The amount column in the data frame got standardized. |
| #Imbalance in the data:  Here, the imbalance is highly skewed and shows that   * 473 are invalid transactions * 275190 are valid transactions       #Plotting the correlation matrix      #Handling the imbalance data |
| # Separating X and y values    For “*Undersampling”* I have used three algorithms:  #Logistic Regression  #Decision Tree Classifier  #Random Forest Classifier |
|  |
| # Decision Tree classifier |
| #Random Forest Classifier |
| #Final data of undersampling |
| Here, Logistic regression is the best fit model to work with. Now, we will see the effect of oversampling.  For “*Oversampling”* I have used three algorithms:  #Logistic Regression  #Decision Tree Classifier  #Random Forest Classifier    #Logistic regression |
| # Decision Tree Classifier |
| #Random forest Classifier |

Final data of oversampling:



Here , from the graph it can be predicted as Random forest classifier as the best fit.



# Save the model



By following three algorithms we have predicted not only the accuracy score but also the precision score, recall score, and f1 score. Further, for testing any transaction we have set up the model to do a quick check and give results as shown in the last output.