## About Dataset

## Context

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.

## Content

The dataset contains transactions made by credit cards and contains 31 columns (variables) and 284807 rows. Variables include Time, V1 – V28, Amount & Class.

‘Class’ is the dependent variable denoted by 0 (non-fraudulent transactions) and 1 (fraudulent transactions)

## Steps:

* Set the working directory and read the data
* Data Cleaning: Check for missing values, duplicates, data types. Convert dependent variable "Class" into a factor variable from character vector.
* Run logistic regression
* Run step wise function for step wise variable selection. Variables V2, V3, V11, V12, V17, V18, V19, V24, V25, V26 have been removed.
* Check for multicollinearity through variance inflation factor
* Variables V1, V6, V7, V10, V20, Amount have absolute values more than 5. Let us drop ‘Amount’ first as it has the highest VIF. We also drop the variables which were removed after running the step wise function
* Run logistic regression again with the reduced number of variables and run the VIF again. Variable ‘V10’ has an absolute value of 6.79. Let's drop ‘V10’
* Run the VIF again and now the absolute values for all the variables are below 5.
* Split the data into train and test by using ‘createDataPartition’ function in 80/20 ratio
* Run logistic regression on train data and use step wise function
* Predict the test data
* We find out the Accuracy (99.92%), Sensitivity (60.2%), Specificity (99.98%) through Confusion Matrix
* We check for data imbalance and find out the data is heavy imbalanced with non-fraudulent transactions accounting for 227452 and fraudulent being only 394. To overcome this imbalance, we use the library (ROSE) and ovun.sample function by doing oversampling, undersampling and both sampling
* Confusion Matrix provide the following results:
* Accuracy:97.24%, Sensitivity:85.71%, Specificity:97.25% (Logistic Regression for oversampling)
* Accuracy:95.6%, Sensitivity:89.79%, Specificity:95.61% (Logistic Regression for undersampling)
* Accuracy:97.15%, Sensitivity:86.73%, Specificity:97.16% (Logistic Regression for both sampling)
* We use a different model “DECISION TREE” for prediction purpose for all three
* We set the seed and use library(rpart)
* Confusion Matrix provide the following results:
* Accuracy:96.21%, Sensitivity:85.71%, Specificity:96.22% (Decision Tree for over sampling)
* Accuracy:95.77%, Sensitivity:85.71%, Specificity:95.79% (Decision Tree for under sampling)
* Accuracy:96.21%, Sensitivity:85.71%, Specificity:96.22% (Decision Tree for both sampling)
* We use a different model “RANDOM FOREST” for prediction purpose for all three
* We set the seed and use library(randomForest)
* Confusion Matrix provide the following results:
* Accuracy:99.95%, Sensitivity:72.44%, Specificity:99.99% (Random Forest for over sampling)
* Accuracy:95.18%, Sensitivity:87.75%, Specificity:95.18% (Random Forest for under sampling)
* Accuracy:99.95%, Sensitivity:75.51%, Specificity:99.99% (Random Forest for both sampling)
* While confusion matrix is a good method for dealing with imbalanced datasets, ROC and AUC are a better indicator.
* We use library(pROC) to find the AUC for all three

|  |  |  |  |
| --- | --- | --- | --- |
|  | Area Under Curve | | |
| **Model** | Oversampling | Undersampling | Both |
| Logistic Regression | 97.00% | 96.60% | 97.00% |
| Decision Tree | 95.20% | 90.50% | 95.20% |
| Random Forest | 96.50% | 97.50% | 96.20% |

* **Conclusion: We should go for both sampling under Logistic Regression as it has the highest AUC**