**Application for credit card fraud detection: Based on decision tree and K-Fold cross validation.**

**Abstract:**

The scope of this document is about a proposed classifier to detect which customers are engaged in credit card fraud activity. As credit card fraud is increasing considerably with the development of modern technology and the global superhighways of communication. Credit card fraud costs consumers and the financial companies billions annually, also the fraudsters continuously try to find new rules and tactics to commit illegal actions. In proposed system, creating a classification model based on decision tree and k-fold algorithms and classify rules for credit card fraud detection. By the implementation of this approach in fraud detection systems, financial losses due to fraudulent transactions can be decreased more.

1. **Introduction:**

From the moment the [e-commerce payment systems](https://spd.group/ecommerce-solutions/e-payment-systems/) came to existence, there have always been people who will find new ways to access someone’s finances illegally. In present days credit card fraud causes great losses in money for merchants and individuals, as a result many business investments are in danger. On one hand, preventing the unauthorized persons from getting access to credit cards and increasing the security of credit cards is a must. On the other hand, excessive prevention of unauthorized use of a credit or debit card to obtain money or property fraudulently is hard to overstate. So, it is important that credit card companies are able to recognize fraudulent credit card transactions and define rules to deal with this detection.

**Dataset information:**

This dataset presents transactions that occurred in 2019 & 2020. A total of 13280 records and 22 features are collected for use in this experiment. There are some enhancements have been done on this dataset to prepare it to work in the proposed algorithms perfectly.

1. **Dataset preprocessing steps:**

Check the null values, and remove the duplicates in the dataset. Convert categorical values into numerical values by using the TargetEncoder class from category\_encoders preprocessing library. Create a correlation on the dataset using Pearson method and define the heatmap using Seaborn library to select the needed features in the proposed system.

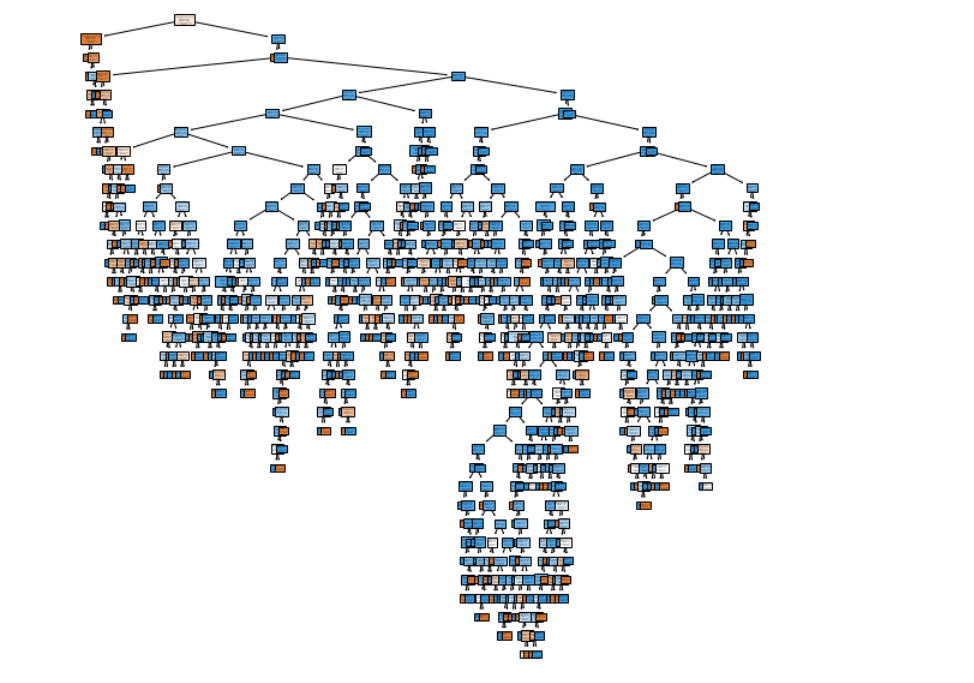
Based on the correlation and heatmap results, Drop the unnecessary columns (street, trans\_num, first, last, dob, merch\_lat, merch\_long) as they won’t affect on the result and to improve the model’s ability to classify quickly.

1. **ML Fraud Detection System Implementation Steps:**

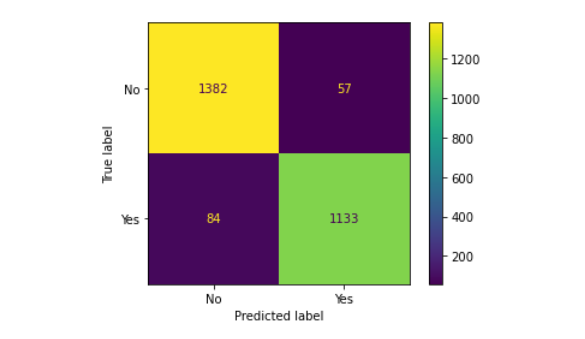
Building the proposed system requires a combination between decision tree algorithm and K-Fold algorithm.

* 1. In order to build a classification model using decision tree algorithm, import the DecisionTreeClassifier class from Sklearn.tree library, create an instance of DecisionTreeClassifier with the default attribute criterion (criterion=’entropy’) for the information gaining, and the strategy used to choose the split at each node is random, then train it on the dataset using Fit function, and check its mean accuracy.

To visualize the decision tree, import the tree class from Sklean, and pyplot class from matplotlib.



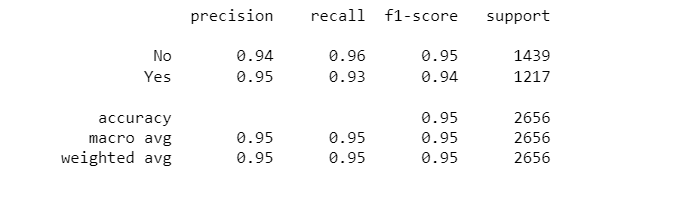
* 1. **Performance Estimation:**
     1. To estimate the performance of the created decision tree, use the hold-out method which splits the dataset into training and testing sets by importing the train\_test\_split function from Sklearn.model\_selection library with randomize split (80% for training, 20% for testing), and pass the parameters (training data, testing data, test size), then build the decision tree from the training data, and use testing data for prediction, and finally compare the accuracy of decision tree on training dataset and test dataset.
     2. Also, the estimation performance can be made by generating the confusion matrix by import scikit-learn metrics module for accuracy calculation, and visualize it by matplotlib library.



By comparing the performance scores on the test and training datasets, there is no overfitting or underfitting.

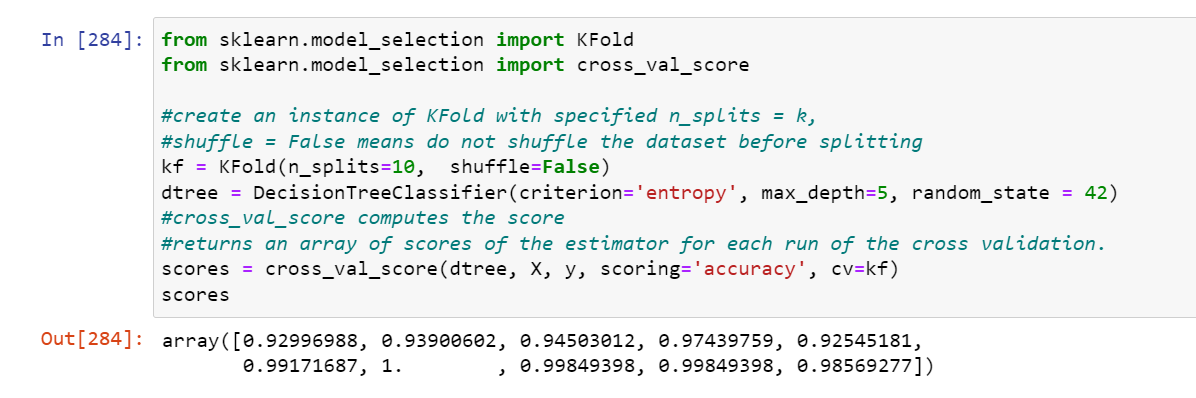
* + 1. **Classification report**

To generate the classification report, compute precision, recall, F-score, and support by using the metrics.classification\_report function which returns the result in a dictionary.

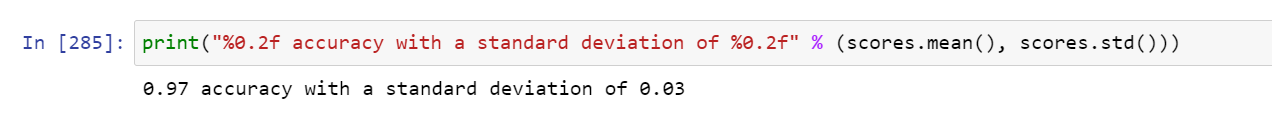


* 1. In order to build the K-Fold cross validation algorithm, use the sklearn.model\_selection to import the KFold, and cross\_val\_score functions, create an instance of KFold with specified number of splits (which in this case was 10 splits) and perform the 10-fold cross validation with decision tree. Using the ten as number of splits is because the extensive experiments have shown that this is the best choice to get an accurate estimation.

For training and testing the data, the cross\_val\_score is used to compute the accuracy scores for each iteration and it returns an array of scores of the estimator for each run of the cross validation.

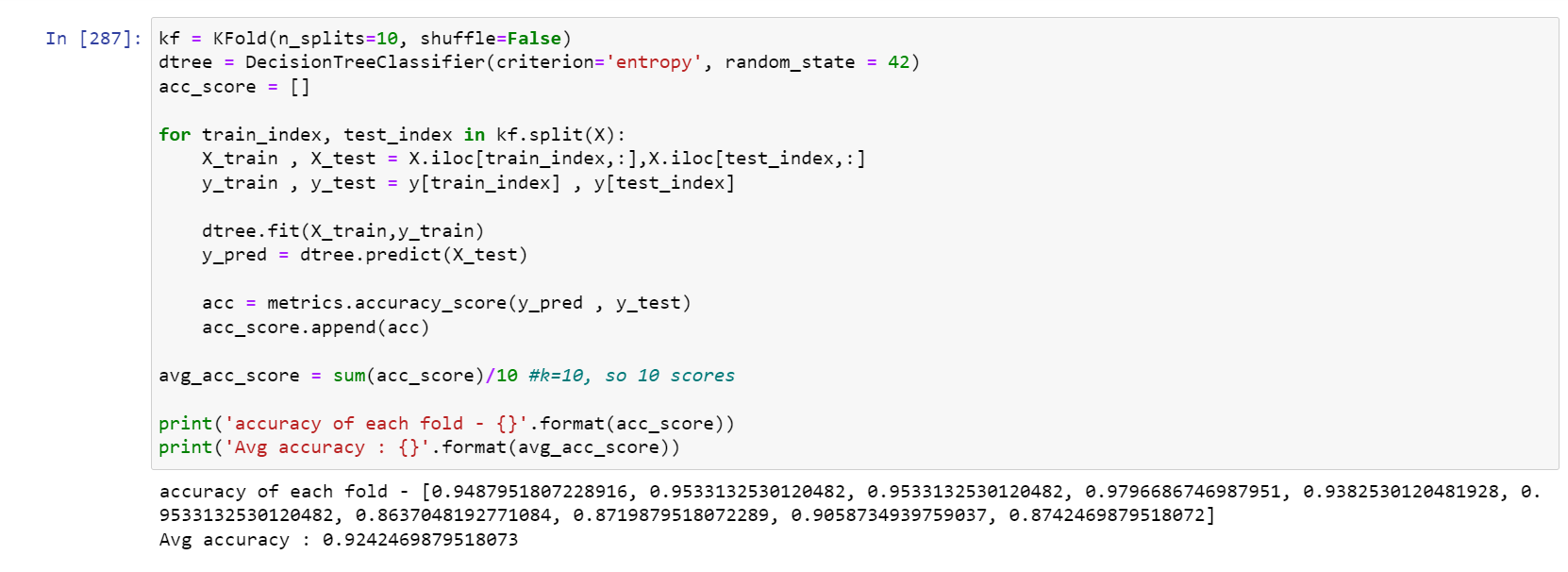


* 1. To estimate the performance of the created 10-fold cross validation algorithm, compute the average score for all the 10 scores with the standard deviation.



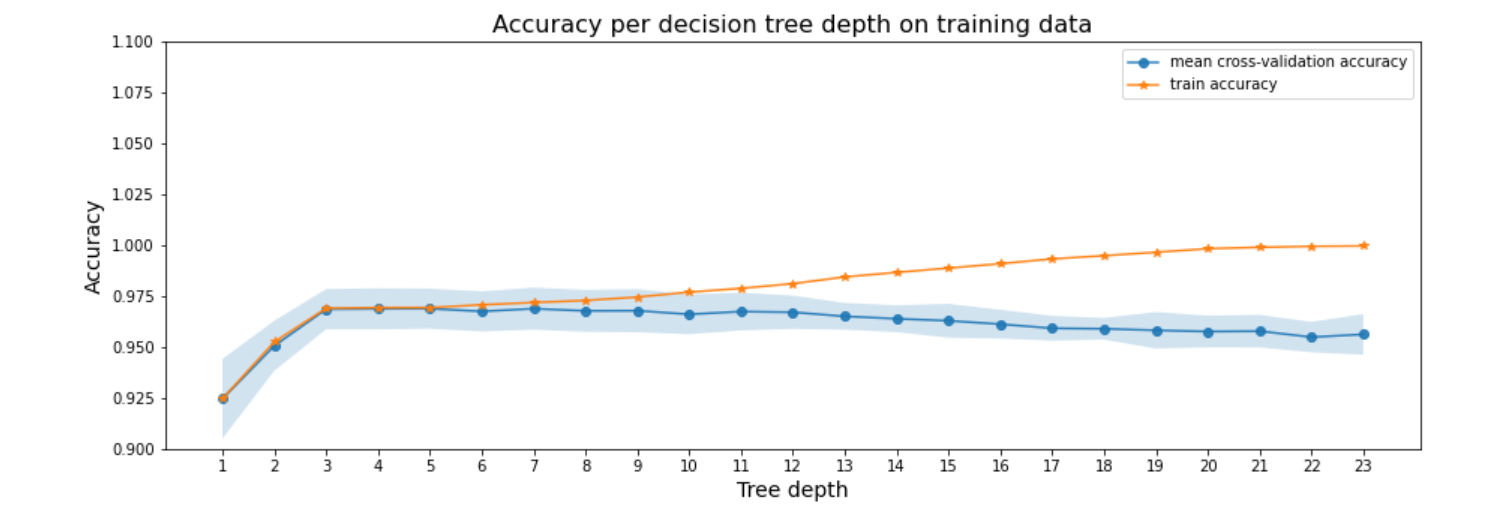
* 1. **Joining the decision tree with k-fold cross validation.**

Making an iteration over the different depths of decision tree with 10-fold cross validation by implement the cross\_val\_score function with kf.split function, then print a list for the accuracy of each fold and the mean accuracy score.

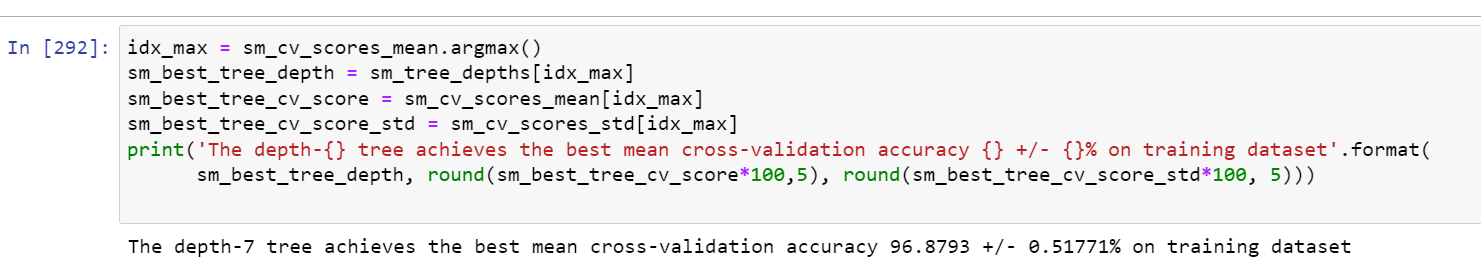


* 1. **Fitting decision tree of various depths on the training data to find optimal depth via K-fold cross validation.**

Choose a range of tree depths to evaluate the estimated performance for each depth and plot the cross-validation results.

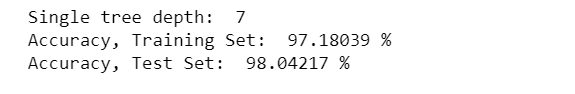


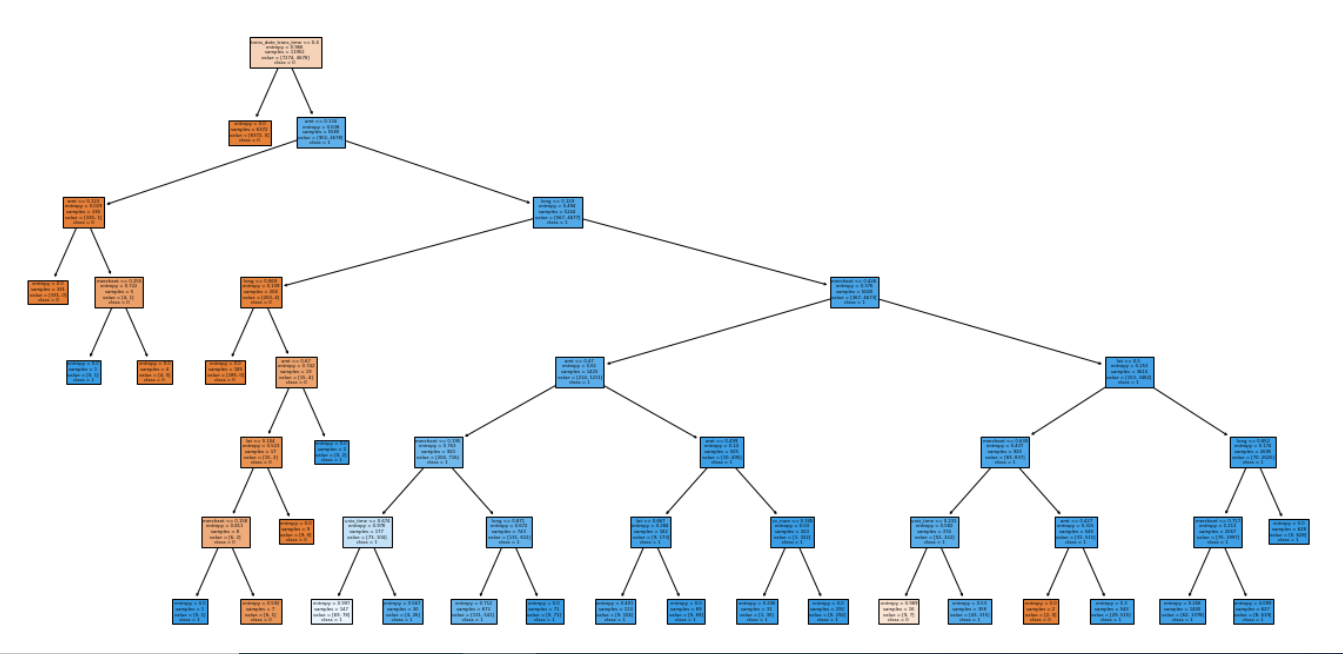
The method selects tree depth 7 because it achieves the best average accuracy on the training data using cross-validation with size 10.



* 1. **Building a decision tree with the selected depth using the training dataset.**

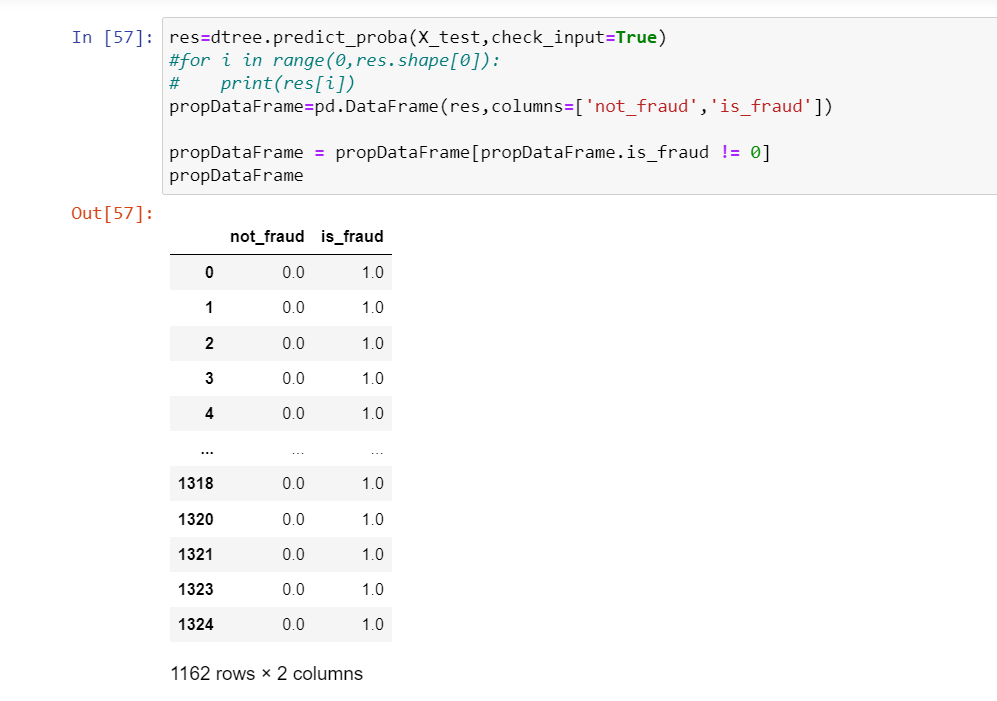
The selected tree depth 7 will help in avoiding the overfitting and gives a better chance to reproduce the accuracy and generalize the model on test data.





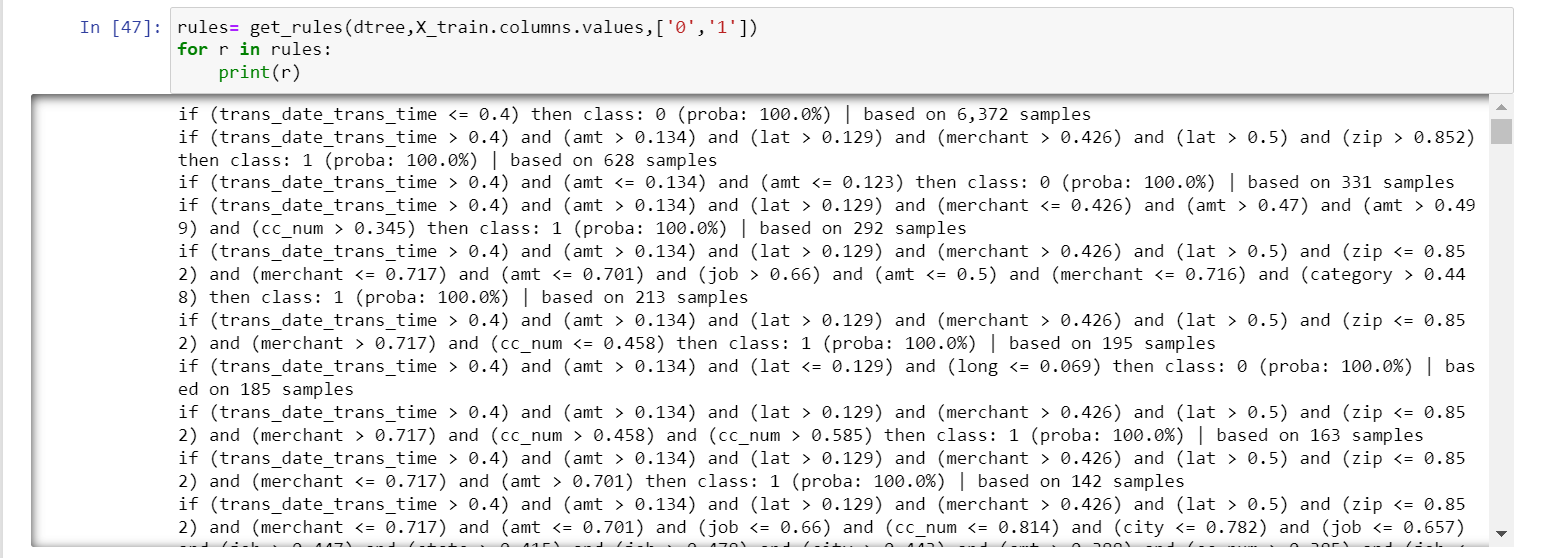
* 1. **The probability of a transaction to be fraud.**

By applying the predict\_propa() function from sklearn library to predict the probability of a credit card transaction to be fraud or not, there are 1162 transactions are fraud.



* 1. **Converting the decision tree into set of rules.**

Finally, generating the rules from the decision tree that will help the classifier in classifying the credit card fraud detection.



# References

Ajitesh, K., 2020. *Visualize Decision Tree with Python Sklearn Library.* [Online]   
Available at: https://vitalflux.com/visualize-decision-tree-python-sklearn-library/  
[Accessed 30 11 2021].

Brownlee, J., 2018. *A Gentle Introduction to k-fold Cross-Validation.* [Online]   
Available at: https://machinelearningmastery.com/k-fold-cross-validation/  
[Accessed 7 12 2021].

Brownlee, J., 2019. *How to Choose a Feature Selection Method For Machine Learning.* [Online]   
Available at: https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/  
[Accessed 28 11 2021].

Chuprina, R. & Kovalenko, O., 2021. *Credit Card Fraud Detection: Top ML Solutions in 2021.* [Online]   
Available at: https://spd.group/machine-learning/credit-card-fraud-detection/  
[Accessed 4 12 2021].

Gaikwad, J. R. et al., 2014. Credit Card Fraud Detection using Decision Tree. *International Journal of Innovative Technology and Exploring Engineering (IJITEE),* 4(6), p. 4.

Kana, M., 2020. *How To Find Decision Tree Depth via Cross-Validation.* [Online]   
Available at: https://towardsdatascience.com/how-to-find-decision-tree-depth-via-cross-validation-2bf143f0f3d6  
[Accessed 7 12 2021].

Sharmila, P., 2020. *CREDIT CARD FRAUD DETECTION WITH CLASSIFICATION ALGORITHMS IN PYTHON.* [Online]   
Available at: https://dataaspirant.com/credit-card-fraud-detection-classification-algorithms-python/  
[Accessed 29 11 2021].

Sidelov, P., 2021. *Credit Card Fraud Detection. Big Players’ Experience.* [Online]   
Available at: https://sdk.finance/credit-card-fraud-detection-big-players-experience/  
[Accessed 9 12 2021].

Zareapoor, M. & Sahmsolmoali, P., 2015. *Application Of Credit Card Fraud Detection: Based On Bagging Ensemble Classifier.* India , Procedia Computer Science.