



CSE 4088
Introduction to Machine Learning
Project

CREDIT CARD FRAUD DETECTION

Aybüke ÖZKAN - 150115005

Celal BAYRAK - 150114044

Abstract

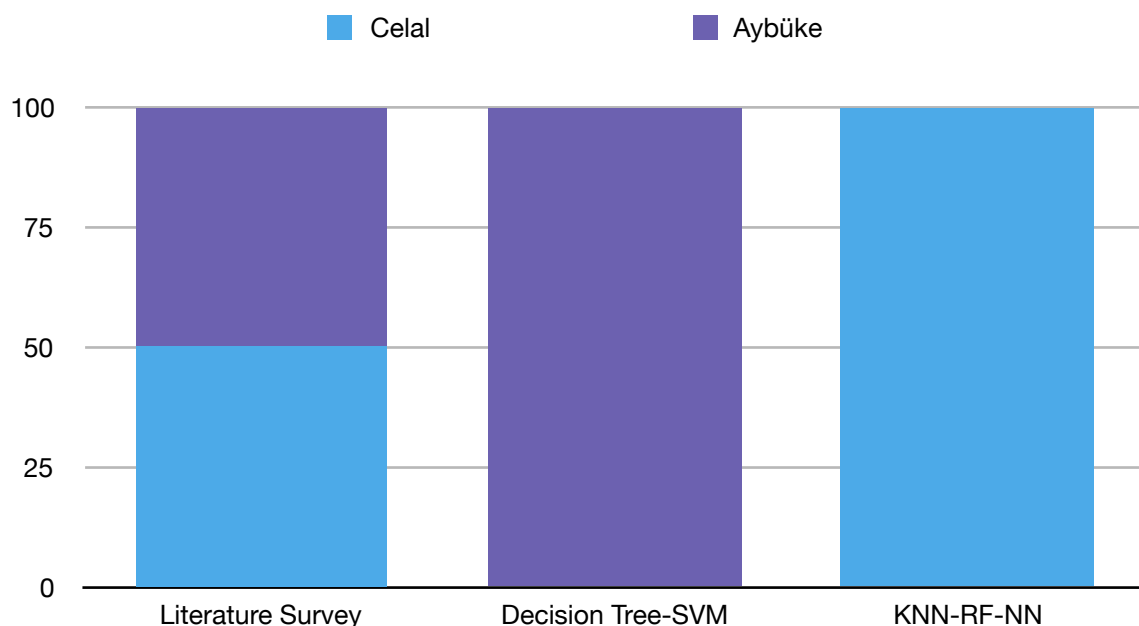
According to the World Payments Report [1], in 2016 total non-cash transactions increased by 10.1% from 2015 for a total of 482.6 billion transactions. This also means that fraudulent transactions are on the rise as well [2]. There is a very high amount of money lost from credit card fraud and there are a lot of methods to prevent that.

In this project, we compared the performances of different algorithms that detects whether the usage of credit card is fraud. We aimed to measure the performances of these algorithms on unbalanced dataset.

We used Kaggle Credit Card Fraud Detection Dataset [3] for our project. It contains 285,000 rows of data.

Overview

- Literature survey about credit card fraud (November 22, 2019): We looked for the previous researches and papers that is related to credit card fraud issue.
- Searching for different algorithms (November 22, 2019): We searched about the different algorithms and selected 5 algorithms: Decision Tree Classifier, K-Nearest-Neighbor, Random Forest Classifier, Support Vector Classifier, and Neural Networks.
- Preparing the midterm report (November 24, 2019)
- Implementation and training of different algorithms (January 6, 2020): We implemented and trained the algorithms mentioned above. We measured and compared their success by using ROC-AUC metric.
- Final presentation (January 7, 2020)
- Preparing the final report (January 8, 2020)



Accomplishments

Credit Card Fraud

Development of modern technology makes the credit card fraud a major topic. Credit card fraud costs consumers and the financial companies billions of dollars annually, and fraudsters continuously try to find new rules and tactics to commit illegal actions [4].

In our research, we encountered some challenges about the detection of credit card fraud. One of them was that the real data sets are not available for researchers. Because of the privacy reasons, banks and financial institutions do not want to share their customer transaction data.

The other challenge was the size of the data. There are millions of credit card transactions processing every day. This reveals an enormous amounts of data which requires competent techniques and computing power.

The main challenge of credit card fraud detection is unbalanced data set. Unbalanced data set is a data set in which classes are not evenly distributed, there is not approximately the same number of data for each class when classifying. In credit card fraud detection data sets, they have more of legitimate data and a few of fraudulent data. In general, real cases have a 98% of legal transaction and only 2% of fraud transaction [5]. Many machine learning algorithms do not take into account the uneven distribution of classes and may not yield reliable results. We have to handle the unbalanced data set to get a reliable result.

Firstly, we had to determine the appropriate evaluation parameters for unbalanced data. Accuracy, which is used to measure the performance of many classifications and trained algorithms, was not a reliable metric to measure model performance when using unbalanced data. Therefore, we worked with metrics like precision, ROC, AUC.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Another method for dealing with the unbalanced data is resampling. First method that can be used is over-sampling, which is to increase the data belonging to the minority class by various methods to obtain classes with equal number of data. For example, Naive Random Over-Sampling algorithm provides balance by randomly selecting and replacing existing data belonging to the minority class. SMOTE (Synthetic Minority Over-sampling Technique) and ADASYN (Adaptive Synthetic Sampling Method) algorithms provide balance by producing synthetic data by interpolation method.

The other resampling method is under-sampling, which is used to obtain a balanced data set by subtracting the data from the majority class from the data set. Random Under-sampling, ClusterCentroids and NearMiss methods are used for this purpose. However, if the dataset is not big enough under-sampling may cause data loss.

Adjusting class weights is another method that can be used to handle with the unbalanced data. Many classifiers have a parameter called “class_weight”. By increasing the weight assigned to the minority class in proportion as imbalance, we can increase the error rate caused by the algorithm's misclassification of minority data. By this way, the algorithm that tries to reduce the overall error rate will take into account the minority class and the performance will be improved.

Methods like penalized-SVM, penalized-LDA or logistic regression are used to increase the weight of the minority class, by giving extra cost on misclassified minority class data.

1. Decision Tree Classifier [10]

Decision Tree is a simple representation for classifying examples. It is a Supervised Machine Learning where the data is continuously split according to a certain parameter. There are two main types of Decision Trees: Classification Trees and Regression Trees.

- Classification trees (Yes/No types): Where the outcome was a variable like ‘fit’ or ‘unfit’. Here the decision variable is categorical/discrete. Such a tree is built through a process known as binary recursive partitioning. This is an iterative process of splitting the data into partitions, and then splitting it up further on each of the branches.
- Regression trees (Continuous data types): Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees. (e.g. the price of a house, or a patient’s length of stay in a hospital)

Decision Tree Classifier

- Using the decision algorithm, we start at the tree root and split the data on the feature that results in the largest information gain (IG) (reduction in uncertainty towards the final decision).
- In an iterative process, we can then repeat this splitting procedure at each child node until the leaves are pure. This means that the samples at each leaf node all belong to the same class.
- In practice, we may set a limit on the depth of the tree to prevent overfitting. We compromise on purity here somewhat as the final leaves may still have some impurity.

2. Support Vector Machine (SVM)

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N is the number of features) that distinctly classifies the data points.

To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e. the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence [6].

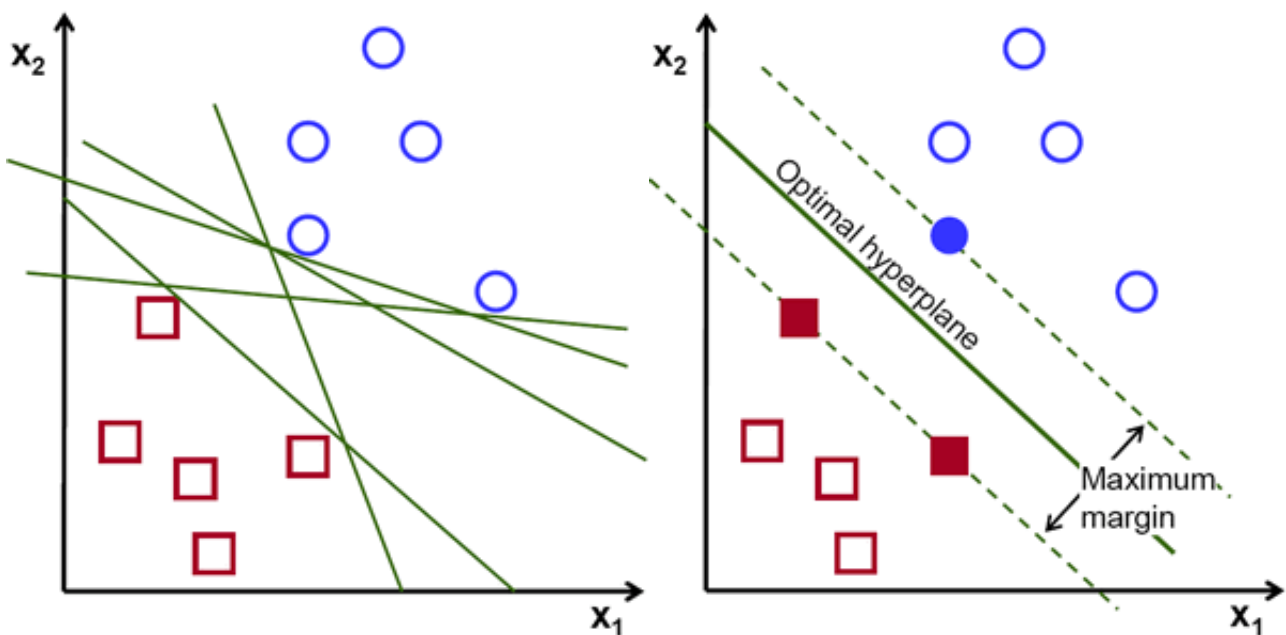


Fig. 1. Possible hyperplanes [6]

3. Random Forest (RF)

Random forest is a classifier that evolves from decision trees. It actually consists of many decision trees. To classify a new instance, each decision tree provides a classification for input data; random forest collects the classifications and chooses the most voted prediction as the result. The input of each tree is sampled data from the original dataset. In addition, a subset of features is randomly selected from the optional features to grow the tree at each node. Each tree is grown without pruning. Essentially, random forest enables a large number of weak or weakly-correlated classifiers to form a strong classifier [7].

4. K-Nearest Neighbor (KNN)

The KNN approach to classification is a relatively simple approach to classification that is completely nonparametric. Given a point x_0 that we wish to classify into one of the K groups, we find the k observed data points that are nearest to x_0 . The classification rule is to assign x_0 to the population that has the most observed data points out of the k -nearest neighbors. Points for which there is no majority are either classified to one of the majority populations at random, or left unclassified.

The advantage of nearest-neighbor classification is its simplicity. There are only two choices a user must make: (1) the number of neighbors, k and (2) the distance metric to be used. Common choices of distance metrics include Euclidean distance, Mahalanobis distance, and city-block distance. The number of neighbors is usually selected by either cross-validation or testing the quality of the classifier on a second, test data set [8].

5. Neural Networks (NN)

Neural network is a computational learning system that uses a network of functions to understand and translate a data input of one form into a desired output, usually in another form. The concept of the artificial neural network was inspired by human biology and the way neurons of the human brain function together to understand inputs from human senses [9].

Implementation of the Algorithms

The dataset we used for fraud detection is heavily imbalanced. The method which we used to handle this was SMOTE. SMOTE algorithm provides balance by producing synthetic data by interpolation method. After we used SMOTE to generate synthetic data, the distribution of classes became equal as shown as in Fig.3.

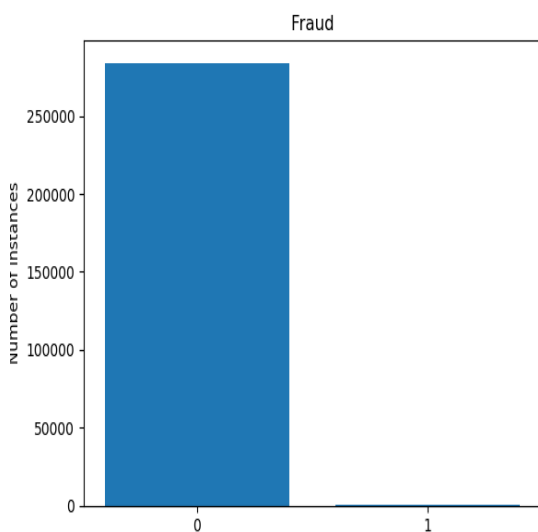


Fig. 2. Class distribution of the original dataset

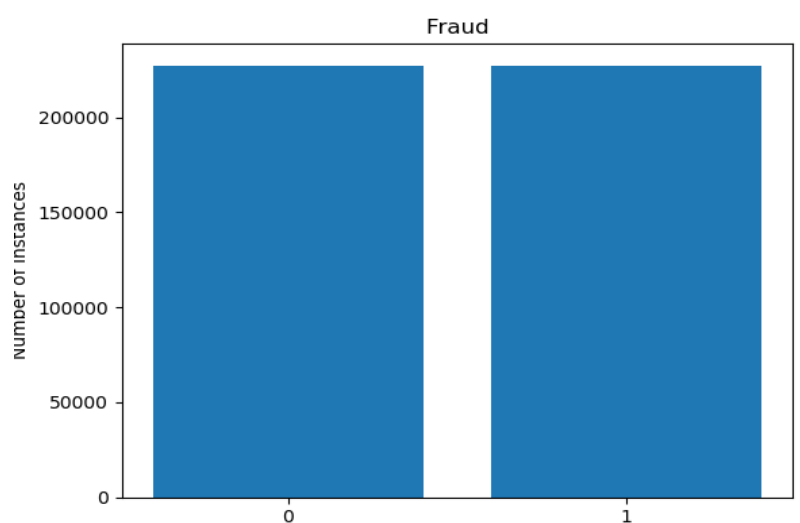


Fig. 3. Class distribution of the synthetic dataset

We have used ROC-AUC score for measuring the successes of algorithms. We did not use accuracy metric because the test set is highly imbalanced. For example if we use accuracy metric and if the model predicts all of data as '0', the accuracy score will be over 99%. So it will be illusive. ROC-AUC score is the mean of True Positive Rate and True Negative Rate. True Negative predictions and True Positive predictions are shown in Fig.4.

		Predicted	
		Negative	Positive
Actual	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

Fig. 4. Confusion matrix

True Positive Rate is $\text{True Positive} / (\text{True Positive} + \text{False Negative})$ and True Negative Rate is $\text{True Negative} / (\text{True Negative} + \text{False Positive})$. So ROC-AUC score gives more realistic and logical scores compared to accuracy.

KNN Results

The validation curve of K-NN is shown in Fig.6. Due to the curve we chose `n_neighbor` parameter as 5. Fig.5 represents the confusion matrix of predictions. The ROC-AUC Score of K-NN is 0.7707.

	0	1
0	54732	2135
1	40	55

Fig. 5. Confusion matrix of K-NN

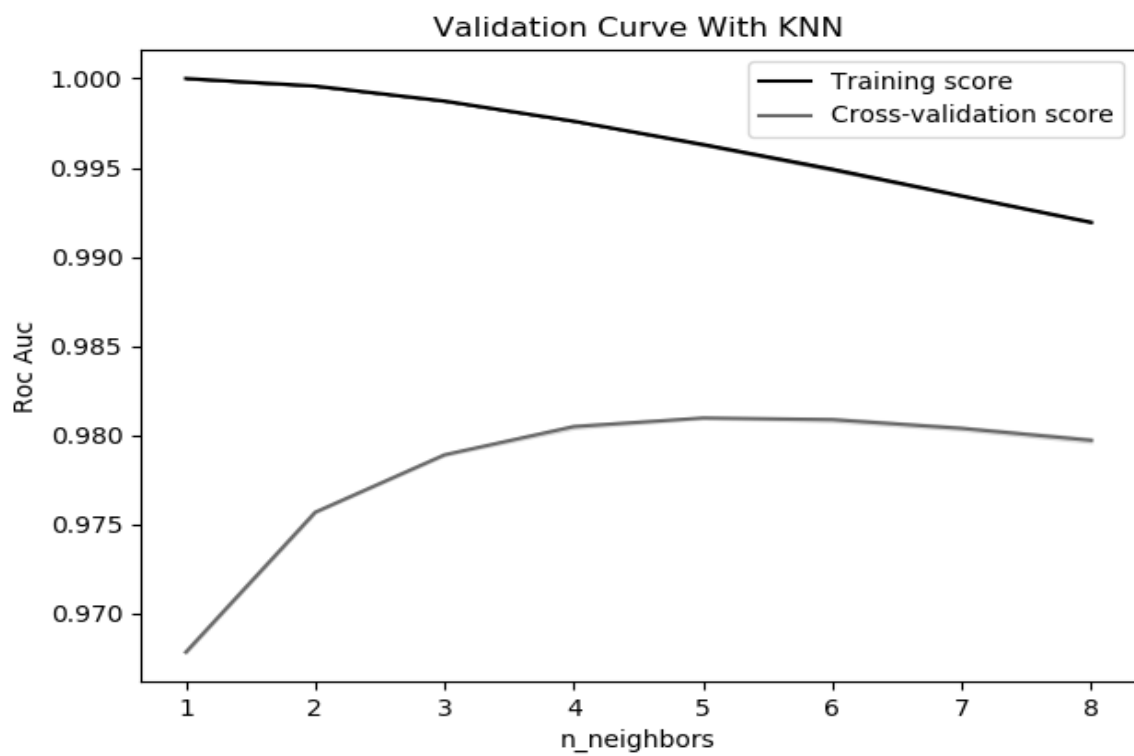


Fig. 6. Validation curve of K-NN

Decision Tree Classifier Results

The validation curve of Decision Tree Classifier is shown in Fig.8. The values of x axis represents the class weight of class ‘1’. Due to the curve we chose 0.9 the class weight of class ‘1’. Fig.7 represents the confusion matrix of predictions.The ROC-AUC Score of Decision Tree Classifier is 0.8836.

	0	1
0	56802	65
1	22	73

Fig. 7. Confusion matrix of Decision Tree

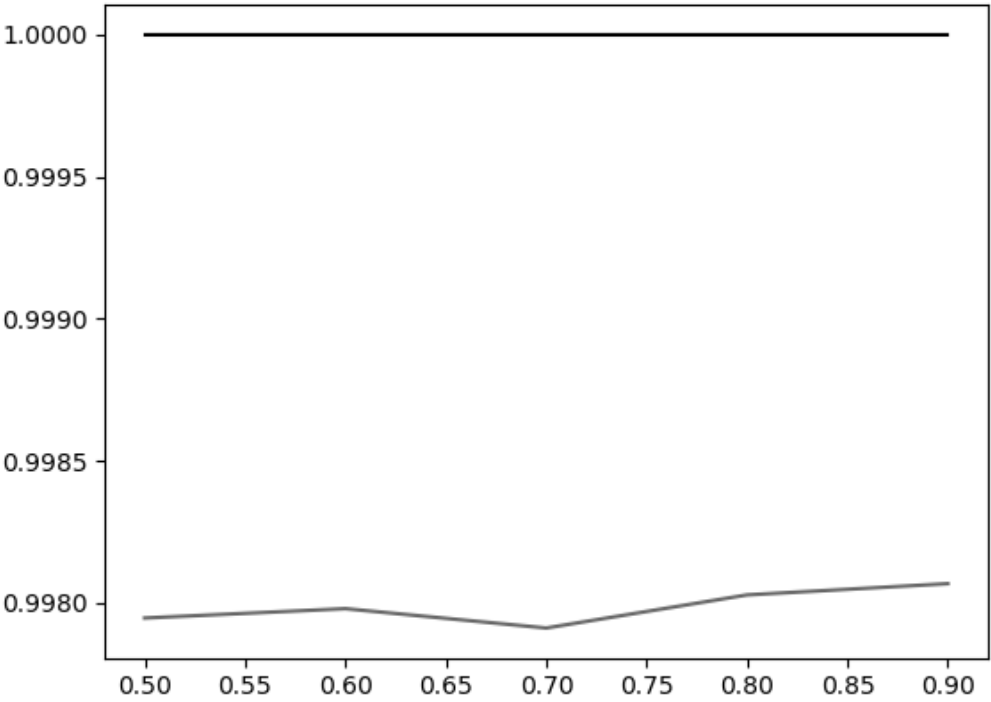


Fig. 8. Validation curve of Decision Tree

Random Forest Classifier Results

The validation curve of Random Forest Classifier is shown in Fig.10. Due to the curve we chose `n_estimator` parameter as 16. Fig.9 represents the confusion matrix of predictions. The ROC-AUC Score of Random Forest Classifier is 0.8841.

	0	1
0	56857	10
1	22	73

Fig. 9. Confusion matrix of Random Forest

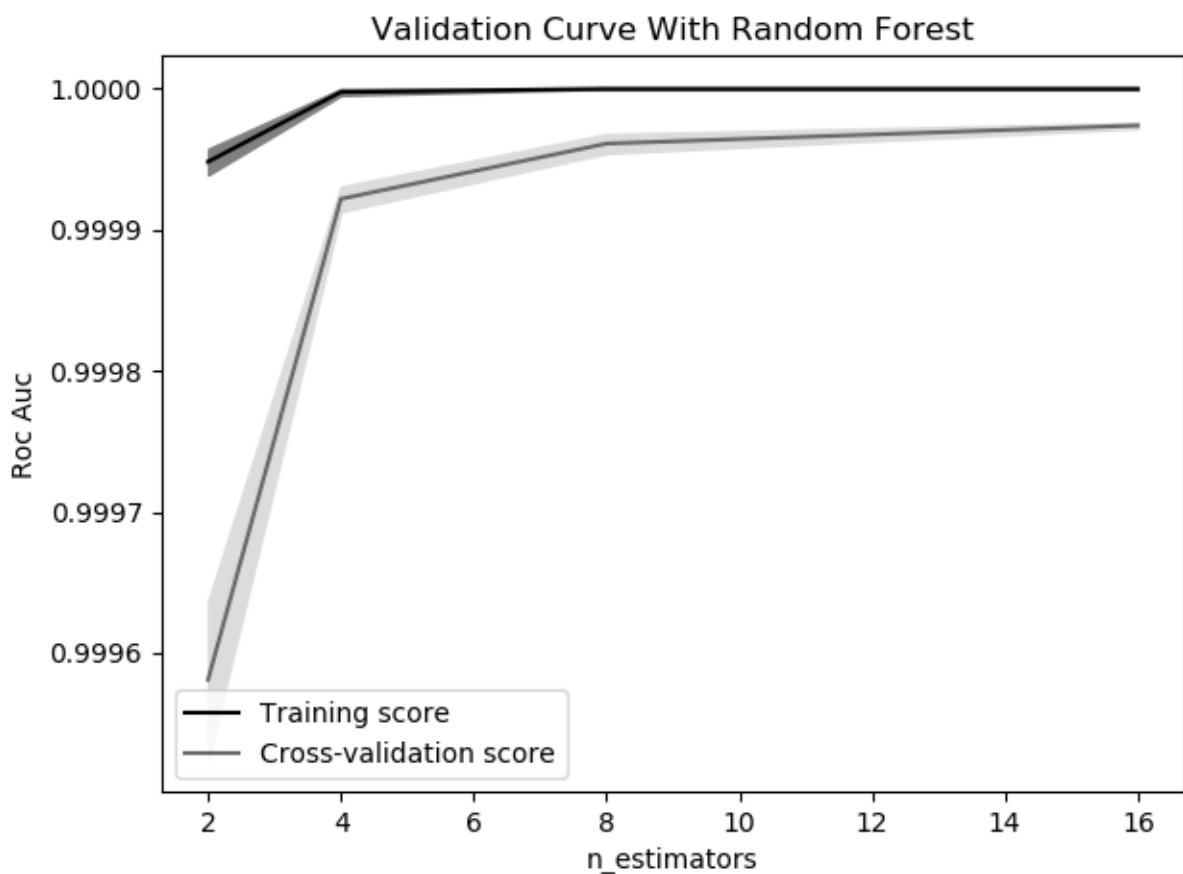
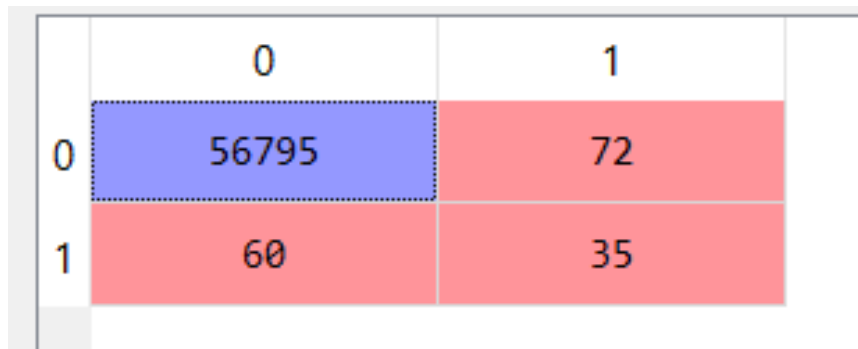


Fig. 10. Validation curve of Random Forest

Support Vector Classifier Results

Fig.11 represents the confusion matrix of predictions of Support Vector Classifier. Training of SVM takes long time. So we couldn't tune any parameter of SVM and we have trained with only 100000 rows of dataset. Because of that SVM's predictions are not accurate compared to other algorithms. The ROC-AUC Score of SVM is 0.6835.

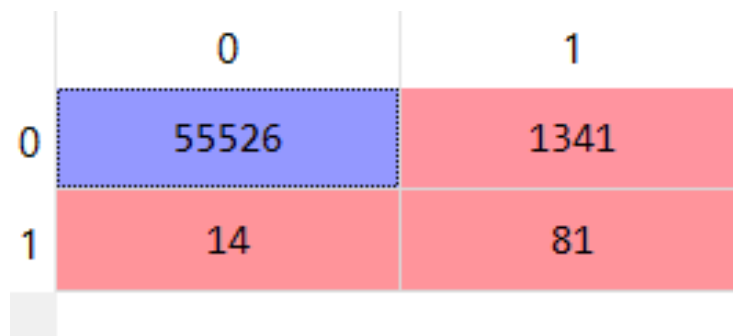


	0	1
0	56795	72
1	60	35

Fig. 11. Confusion matrix of SVM

Neural Network Results

Fig.12 represents the confusion matrix of predictions of Neural Networks. We have trained the networks with 20 epochs. Neural Networks gave the best result in our experiments The ROC-AUC Score of Neural Networks is 0.9145.



	0	1
0	55526	1341
1	14	81

Fig. 12. Confusion matrix of NN

Summary

We trained 5 different algorithms with the equally distributed dataset, but in test dataset we have not used synthetic data because using synthetic data for testing may be illusive. The algorithms we have used are K-NN, Decision Tree Classifier, Random Forest Classifier, Support Vector Classifier and Neural Networks. Testing models with real data gives more realistic results. We get the best results by using Neural Network.

References

- [1] World Payments Report, 14th Edition. 2018. [<https://worldpaymentsreport.com/wp-content/uploads/sites/5/2018/10/World-Payments-Report-2018.pdf>]
- [2] Macaraeg, R. Credit Card Fraud Detection. 2019. [<https://towardsdatascience.com/credit-card-fraud-detection-a1c7e1b75f59>]
- [3] Kaggle Credit Card Fraud Detection Dataset. [<https://www.kaggle.com/mlg-ulb/creditcardfraud>]
- [4] Zareapoor, M., Shamsolmoali, P., Application of credit card fraud detection based on bagging ensemble classifier. Procedia Computer Science, 48, 2015, pp: 679-685. [<https://www.sciencedirect.com/science/article/pii/S1877050915007103>]
- [5] J. Piotr., A.M. Niall, J.D. Hand, C. Whitrow, J. David (2008). Off the peg and bespoke classifiers for fraud detection. Computational Statistics and Data Analysis, 52, pp: 4521-4532.
- [6] Gandhi, R. Support Vector Machine — Introduction to Machine Learning Algorithms. 2018. [<https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>]
- [7] Mao, W., Wang, F. Y., Cultural modeling for behavior analysis and prediction. New Advances in Intelligence and Security Informatics, 2012, pp: 91-102. [<https://www.sciencedirect.com/science/article/pii/B9780123972002000087>]
- [8] Neath, R.C., Johnson, M.S., Discrimination and classification, International Encyclopedia of Education (Third Edition), 2010, pp: 135-141. [<https://www.sciencedirect.com/science/article/pii/B9780080448947013129>]
- [9] DeepAI. Neural Network-What is a Neural Network. [<https://deepai.org/machine-learning-glossary-and-terms/neural-network>]
- [10] Chakure, A. Decision Tree Classification. 2019. [<https://towardsdatascience.com/decision-tree-classification-de64fc4d5aac>]