Credit Card Fraud Detection

Akash Barnwal
RU ID: 171004409

Abstract

The terminology “Credit card fraud” is a broad term used for theft and fraud committed using payment card such as credit card, debit card, as a fraudulent source of funds in a transaction. According to the **United States Federal Trade Commission**, while the rate of identity theft had been holding steady during the mid-2000s, it increased by 21 percent in 2008. In this paper we try to predict the number of fraud transaction based on the given set of data. The analysis is centered on logistic regression, support vector machine, k nearest neighbor, Random Forest, Neural Net. All these methods have been used to forecast the fraudulent data and to check the model accuracy. Using logistic regression, the best model is predicted and then various measures such as Concordance-Discordance, AUC, Precision Recall and Hosemar Lameshow method has been implemented to get the model accuracy for logistic regression.

Keywords: Credit Card, Support Vector Machine, Random Forest, K nearest Neighbor, Logistic Regression, AUC-RUC, Concordance-Discordance, Neural Net

1. Introduction

Credit Card Fraud is a term widely used for any sort of fraudulent use of a credit card account through the theft of the account holder's card number, card details and personal information, through a wide variety of methods in order to perform unauthorized transactions from the compromised account.

In the year 2016, the numbers of fraudulent transaction done through credit card were $16.31 billion on a total card sales volume of $28.844 trillion. According to the report of Credit Union Times, the number of fraud increased by 19 percent in 2014, while the overall volume of card sales only grew by 15 percent. Hence it’s a very crucial problem to look into the issue and try detecting solution to it.

The project revolves around with the idea of predicting the number of fraudulent transaction in a dataset. The data has been drawn from Machine Learning Group (http://mlg.ulb.ac.be). The data was then analyzed using various ensemble machine learning methods and logistic regression.

The paper proceeds as follows: Section 2 includes a short Literature Review about type of Credit Card Crimes; Section 3 describes the business problem, Section 4 explains the project objective, Section 5 discusses the Data Understanding, Section 6 discusses the Data Preparation and Cleaning Step, Section 7 explains Logistic Regression, Section 8 explains how to evaluate a logistic regression model, Section 9 discusses about SVM Model, Section 10 discusses about Random Forest, Section 11 discusses about K Nearest Network, Section 12 discusses about Neural Network, Section 13 discusses the conclusion of the paper.

2. Type of Credit Card Crimes

Schemes to gain control of credit card accounts are constantly evolving in order to evade law enforcement, but the most common crimes can be characterized in one of two ways. “Card present” crimes are those in which the victim’s physical credit card has been stolen. This category also includes schemes to apply for new credit cards in the victim’s name, or to change the address on the victim’s account and then request replacement cards. However it is accomplished, the thief comes into possession of the card itself, which is then used for purchases or cash advances.

All other types of credit card fraud fall under the category of “card not present” crimes. As the name suggests, these schemes do not require the thief to gain access to the victim’s physical card, or if they do, the card is returned to the victim without raising suspicion. The fraud is accomplished by recording the credit card number and other identifying information so it can be used online, over the phone, or in some other way that does not require the card to be swiped at the point of sale.

Card not present schemes can be the most difficult to detect, as the thief and the victim may be thousands of miles apart at the time the credit card information is stolen. Skimming machines placed onto ATMs and gas pump terminals are one example. These devices go unnoticed by customers, and secretly capture card data when the customer slides the card through the machine. Phishing websites are another example. These are fake websites that are built to appear like legitimate sites of companies that people trust with their financial information. Victims are tricked into logging into the fake site and entering their credit card number.

3. Business Problem:

Nowadays, enterprises and public institutions have to face a growing presence of frauds and consequently need automatic systems able to support fraud detection and fight. These systems are essential since it is not always possible or easy for a human analyst to detect fraudulent patterns in transaction datasets, often characterized by a large number of samples, many dimensions and online update.

In real life, fraudulent transaction are scattered with genuine transactions and simple pattern matching Techniques are not often sufficient to detect those frauds accurately. Outlier detection is a data mining technique commonly used for fraud detection [1] [2][3][4]. Outliers are data points that are inconsistent with the reminder of the dataset or deviate so much from other observations so as to arouse suspicion that they were generated by different mechanism. The outlier detection can be achieved through techniques like neural network, KNN, Random Forest etc. [5]

4. Project Objectives:

Design, assess and validate a machine learning framework able to calibrate in an automatic, real-time and adaptive manner the fraud detection strategy. The goal is to provide a set of learning tools to be integrated within the credit card fraud detection process in order to improve its robustness, performance and accuracy.

The properties of a good fraud detection system are:

1) It should identify the frauds accurately
2) It should detecting the frauds quickly
3) It should not classify a genuine transaction as fraud

In this paper, a comprehensive review of various fraud detection methods has been performed [6] [7][8][9][10] [11].

5. Data Understanding:

The dataset contains the transaction made by European cardholders in two days of September2013. The data contains 492 frauds out of 2, 84,807 transactions. The dataset is highly unbalanced as the no of fraud counts for 0.172% of all the transactions.

It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, the original features and the background information about the data is not disclosed. 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-dependent cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

1	Time	Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset
2	V1 – V28	These are the features which are obtained with using PCA.
3	Amount	Amount is the transaction amount.
4	Class	Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

6. Data Preparation and Cleaning:

- The data set contains 284807 records containing 492 fraudulent records and remaining non fraudulent records.
- The initial step of analysis or data cleaning involves checking of null values. We can see that the data contains zero null values. In case of existence of null values, we do missing value treatment.
- The next step is to check for unique values in each column.

No. of Unique Values In Each Variables											
Time	**V1**	**V2**	**V3**	**V4**	**V5**	**V6**	**V7**	**V8**	**V9**	**V10**	**V11**
124592	275663	275663	275663	275663	275663	275663	275663	275663	275663	275663	275663
V12	**V13**	**V14**	**V15**	**V16**	**V17**	**V18**	**V19**	**V20**	**V21**	**V22**	**V23**
275663	275663	275663	275663	275663	275663	275663	275663	275663	275663	275663	275663
V24	**V25**	**V26**	**V27**	**V28**	**Amount**	**Class**					
275663	275663	275663	275663	275663	32767	2					

- The data is positively skewed with fraud transactions very less as compared to the non-fraud ones.
- Time features shows the chronological order of the transaction hence it's not a significant feature to be kept.
- The data is divided into test and training data with 70% of data as training set and 30% of the data as test data.
- We see the distribution of target variables which is "Class" in training and test dataset to understand the distribution of them.

7. Logistic Regression:

Binary Logistic Regression is a special type of regression where binary response variable is related to a set of explanatory variables, which can be discrete and/or continuous. Binary Logistic Regression models how binary response variable Y depends on a set of k explanatory variables, X=(X1, X2 ... Xk).

$$\text{logit}(\pi) = \log(\pi/1-\pi) = \beta 0+\beta xi+\ldots+\beta 0+\beta xk'$$

which models the log odds of probability of "success" as a function of explanatory variables.

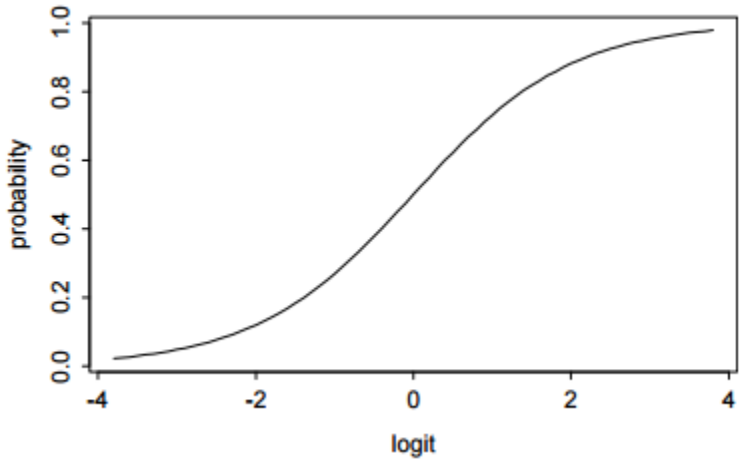
Random component: The distribution of Y is assumed to be Binomial (n, π), where π is a probability of "success".

Systematic component: X's are explanatory variables (can be continuous, discrete, or both) and are linear in the parameters, e.g., $\beta 0 + \beta xi + \ldots + \beta 0 + \beta x_k$. Again, transformation of the X's themselves are allowed like in linear regression; this holds for any GLM.

Link function:

$$\eta=\text{logit}(\pi)=\log(\pi_1-\pi)\eta=\text{logit}(\pi)=\log(\pi_1-\pi)$$

More generally, the logit link models the log odds of the mean, and the mean here is π. Binary logistic regression models are also known as logit models when the predictors are all categorical. The logit transformation is one-to-one. The inverse transformation is sometimes called the anti logit, and allows us to go back from logits to probabilities. [12]



Result After Running the Logistic Regression Model:

- According to the model output, we can see that the significant variables are V4, V8, V10, V13, V14, V20, V21, V22, V27 and v28.
- The negative coefficient for this predictor suggests that all other variables being equal, the variables with negative coefficient are less likely to have fraud values.
- The negative coefficient variables are V6 –V11, V13-V16, V18, V20, V23, and V25 -V28.

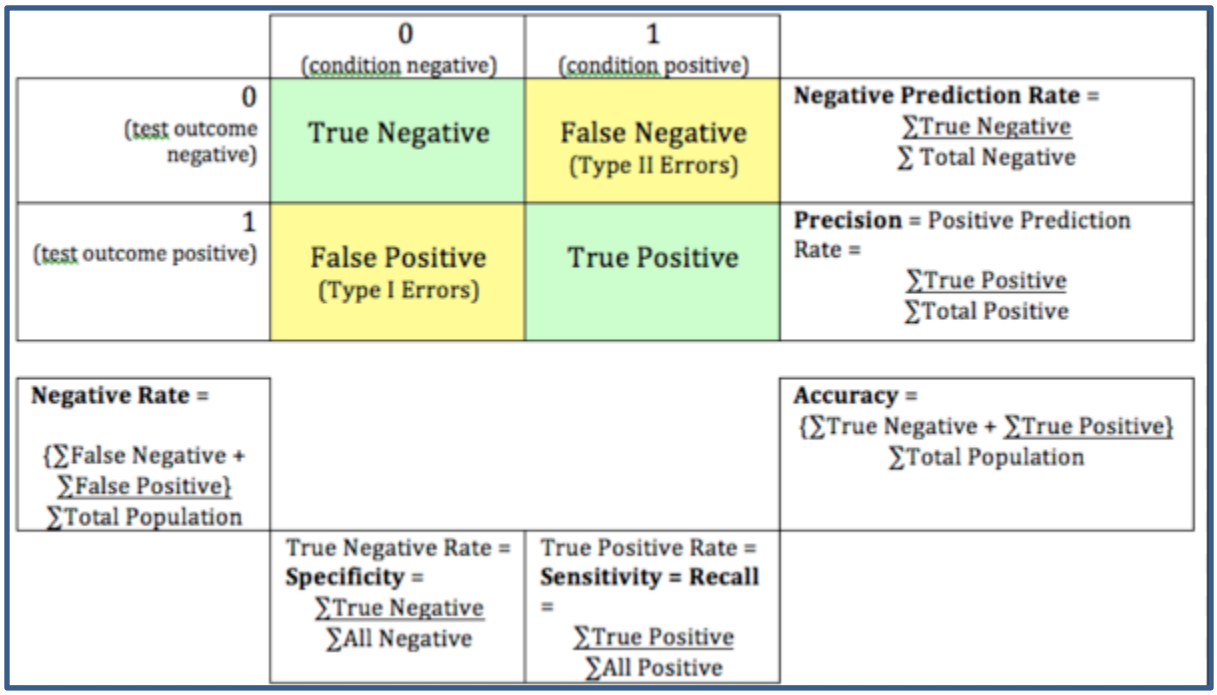
8. Evaluating a Model:

There are several different techniques which can be used for evaluating a model.

I. Confusion Matrix

Confusion matrix is a terminology used in machine learning, specifically in the problems related to statistical classification. It is also known as error matrix which is a table layout that allows visualization of the performance of an algorithm, typically a supervised learning one. The final table of confusion would contain the average values for all the classes combined.

Let us define an experiment from P positive instances and N negative instances for some condition. The four outcomes can be formulated in a 2×2 confusion matrix, as follows:



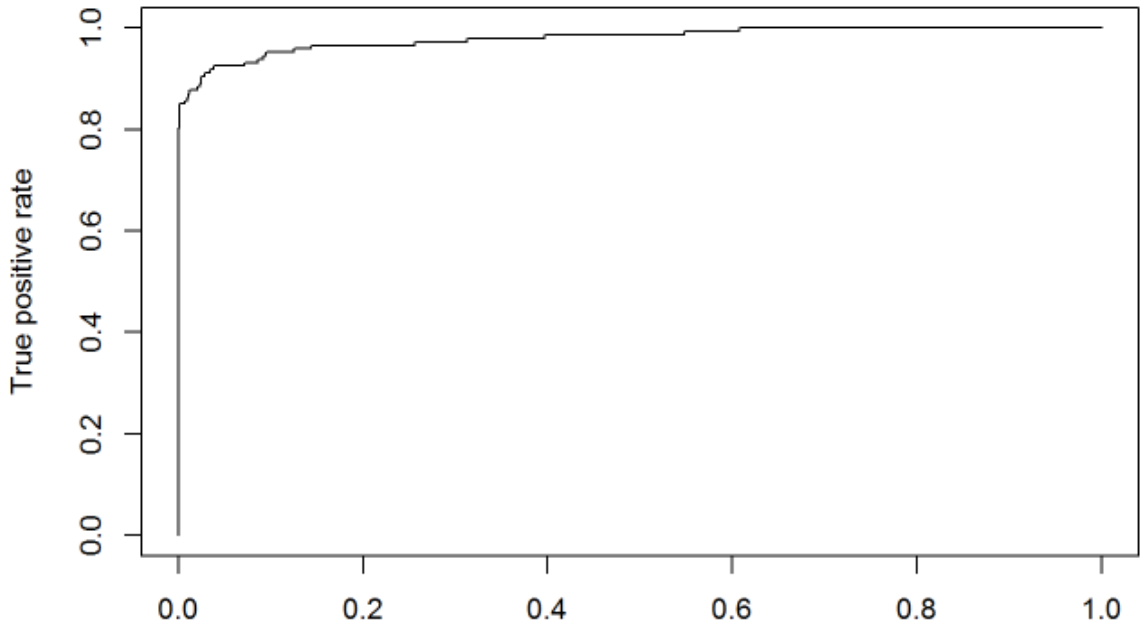
II. Gain and Lift Chart

- Lift is a measure of the effectiveness of a predictive model calculated as the ratio between the results obtained with and without the predictive model.
- Cumulative gains and lift charts are visual aids for measuring model performance
- Both charts consist of a lift curve and a baseline
- The greater the area between the lift curve and the baseline, the better the model

III. AUC – ROC

In statistics, a receiver operating characteristic curve or ROC curve is a plot that explains the performance of a binary classifier as its discrimination threshold is varied. The curve is created by plotting the true positive rate against the false positive rate. The true-positive rate is also known as sensitivity, recall or probability of detection[1] in machine learning. The false-positive rate is also known as the fall-out or probability of false alarm and can be calculated as (1 − specificity).

The AUC –ROC curve for the given set of data is as followed:



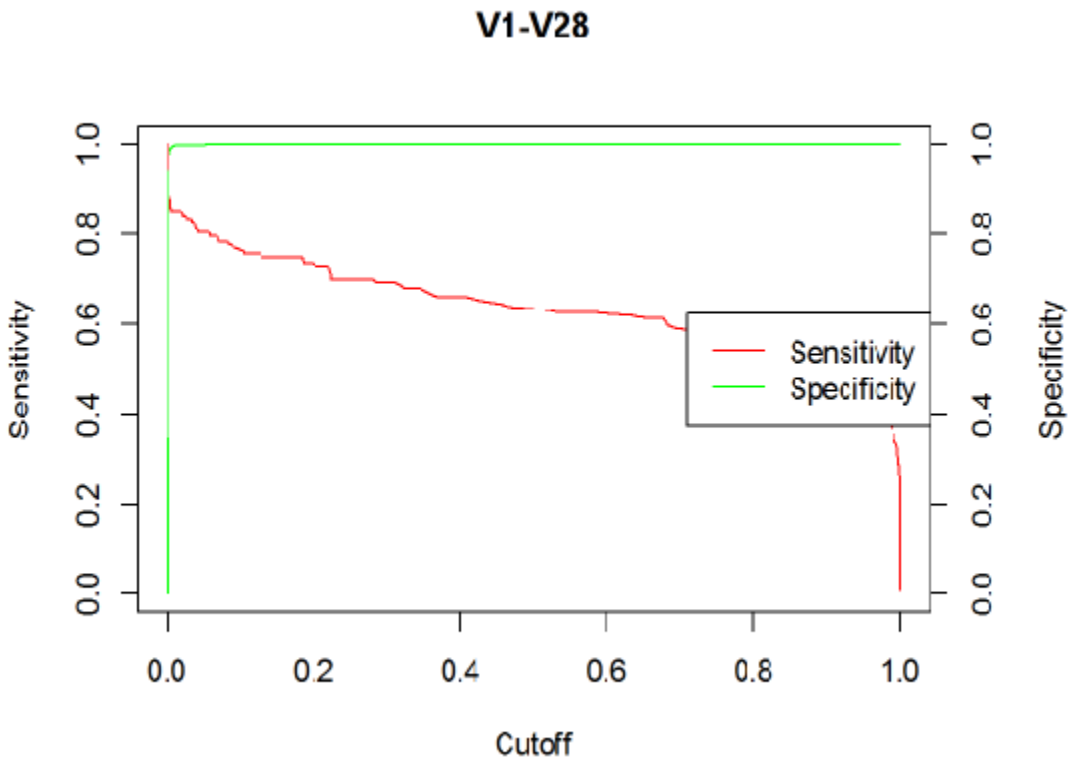
The area under the curve for the model gives an accuracy of 97.84% indicating a good fit.

IV. Sensitivity and specificity

Sensitivity and specificity are statistical measures of the performance of a binary classification test, also known in statistics as classification function:

- Sensitivity (also called the true positive rate, the recall, or probability of detection[1] in some fields) measures the proportion of positives that are correctly identified as such (i.e. the percentage of sick people who are correctly identified as having the condition).
- Specificity (also called the true negative rate) measures the proportion of negatives that are correctly identified as such (i.e., the percentage of healthy people who are correctly identified as not having the condition).

The sensitivity- specificity curve for the given dataset is as followed:



V. Concordance – Discordance Ratio:

- Concordant- Discordant is an important concept to understand whether a model is good or not. It's a technique used to test the goodness of fit in logistic model. The various parameters to look into **Concordant – Discordant Ratio** are**:**

a) Cutoff Value:

For instance, students are classified as pass (1) or fail (0) depending upon the cutoff passing marks in the examination. The cutoff marks vary depending upon the requirements of the different examination. Let us consider the cutoff passing marks as 60 out of 100; therefore, all the students getting marks greater than 60 will be considered pass (1) else fail (0).

b) Cutoff Value for Logistic Regression:

Now, let us consider a binary logistic model in which an event and non-events are classified as (1) and (0) respectively. We get the probability values as an output of the logistic model. We classify the output as event (1) or nonevent (0), depending upon the cutoff probability value (Popular choice of the cutoff probability is 0.5*). Thus, if the cutoff probability is chosen as 0.5, then all the observations with probability value greater than 0.5 will be classified as event (1) else nonevent (0).

Case I: Concordant Pairs:

For example, let us consider pair one (1, 0), which has the probability as (0.9, 0.6) correspondingly. This means that the probability of 0.9 is considered as event (1) and the probability value of 0.6 is considered as non-event (0), in this case. Therefore, for an observation to be classified as an event the cutoff value should be less than 0.9, whereas for an observation to be classified as a non-event cutoff should be more than 0.6. For the coexistence of both the cases, the cutoff value of the probability should be less than 0.9 and greater than 0.6. In case of this pair, cutoff value can vary between 0.6 to 0.9. Similarly, can have another example of a pair having probability values (0.3, 0.2); this pair can have cutoff values ranging between 0.3 and 0.2. Such pairs are termed as concordant pairs.

Case Ii: Discordant Pairs:

Let us take another pair of one and zero (1, 0), with probability value as (0.70, 0.75) respectively. What can be said about the cutoff value, in this case?? In this case the probability of event is less than the probability of non-event. In this case, for an observation to be classified as an event the cutoff value of the probability should be less than 0.70 , whereas for an observation to be classified as an nonevent the cutoff value should be greater than 0.75. Such cases cannot coexist because this is contrary to our definition of the event and non-events based upon the cutoff value. Therefore, such pairs are termed as discordant pairs.

Case Iii: Tied Pairs:

Let us take one more pair of one and zero (1, 0), with probability value as (0.65, 0.65) respectively. What can be said about the cutoff value, in this case?? In this case the probability of event is equal to the probability of nonevent. Looking at event's probability of 0.65, we can say the cutoff value should be less than 0.65, on the other hand , looking at the nonevent's probability , we can say that the cutoff value should be greater than 0.65. Big Problem!!!! Solution-> These two cases can coexist if and only if the cutoff value is equal to 0.65. Such pairs are called Tied pairs.

Interpretation:

The total number of concordant pairs are counted and divided by the total number of pairs. This will give us the value of concordance ratio. The higher the concordance ratio, the better is the model.

VI. Anova Model:

The mathematical model that describes the relationship between the response and treatment for the one way Anova is given by

$$Y_{ij}=\mu+\tau_i+\epsilon_{ij},$$

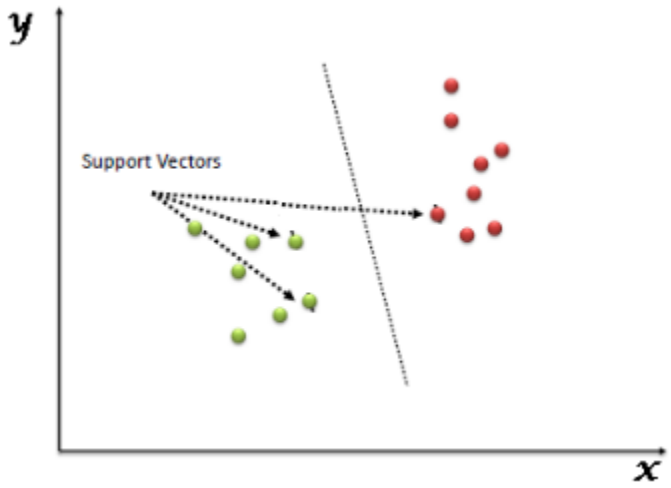
where Y ij represents the jth observation (j=1,2,…, ni)on the ith treatment (i=1,2,…,k levels). So, Y23 represents the third observation using level 2 of the factor. μ is the common effect for the whole experiment, τi represents the ith treatment effect, and ϵij represents the random error present in the j^{th} observation on the i^{th} treatment.

The inferences drawn from running the Anova model on the credit card data are as followed: The difference between the null deviance and the residual deviance shows how our model is doing against the null model (a model with only the intercept). The wider this gap, the better.

- A large p-value here indicates that the model without the variable explains more or less the same amount of variation.

9. SVM Model:

Support Vector Machine is a supervised machine learning algorithm which can be used for both classification and regression challenges with major focus on classification problems. In this algorithm, we plot each data item as a point in n dimensional space with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well.



The basic idea of SVM classification algorithm is to construct a hyper plane as the decision plane which making the distance between the positive and negative mode maximum [30]. The strength of SVMs comes from two important properties they possess - kernel representation and margin optimization. Kernels, such as radial basis function (RBF) kernel, can be used to learn complex regions. A kernel function represents the dot product of projections of two data points in a high dimensional feature space. In SVMs, the classification function is a hyper-plane separating the different classes of data. The basic technique finds the smallest hyper sphere in the kernel space that contains all training instances, and then determines on which side of hyper sphere a test instance lies. If a test instance lies outside the hyper sphere, it is confirmed to be suspicion. This algorithm finds a special kind of linear model, the maximum margin hyper plane, and it classifies all training instances correctly by separating them into correct classes through a hyper plane. The maximum margin hyper plane is the one that gives the greatest separation between the classes. The instances that are nearest to the maximum margin hyper plane are called support vectors. There is always at least one support vector for each class, and often there are more. In credit card fraud detection, for each test instance, it determines if the test instance falls within the learned region. Then if a test instance falls within the learned region, it is declared as normal, else it is declared as anomalous. This model has been demonstrated that it possess a higher accuracy of detection compared with other algorithms. It also has a better time efficiency and generalization ability [49][58]. Performance evaluation of SVM with BPN in credit card fraud detection shows that when the data number is small, SVM can have better prediction performance than BPN in predicting the future data. But in large data BPN has a good performance.

Tuning support vector regression model

In order to improve the performance of the support vector regression we will need to select the best parameters for the model. Tuning parameters value for machine learning algorithms effectively improves the model performance. The various parameters which need to be looked into for tuning are **epsilon** (ϵ), **cost** parameter which is used to avoid over fitting. The process of choosing these parameters is called hyper parameter, or **model selection**.

10. Random Forest

Random forests or Random Decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of over fitting to their training set.

It's a versatile machine learning method capable of performing both regression and classification tasks. It also undertakes dimensional reduction methods, treats missing values, outlier values and other essential steps of data exploration, and does a fairly good job. It is a type of ensemble learning method, where a group of weak models combine to form a powerful model.

How does it work?

In Random Forest, we grow multiple trees as opposed to a single tree in CART model To classify a new object based on attributes, each tree gives a classification and we say the tree "votes" for that class. The forest chooses the classification having the most votes (over all the trees in the forest) and in case of regression, it takes the average of outputs by different trees.

It works in the following manner. Each tree is planted & grown as follows:

1. Assume number of cases in the training set is N. Then, sample of these N cases is taken at random but with replacement. This sample will be the training set for growing the tree.
2. If there are M input variables, a number m<M is specified such that at each node, m variables are selected at random out of the M. The best split on this m is used to split the node. The value of m is held constant while we grow the forest.
3. Each tree is grown to the largest extent possible and there is no pruning.
4. Predict new data by aggregating the predictions of the ntree trees (i.e., majority votes for classification, average for regression).

11. K nearest neighbor:

KNN can be used for both classification and regression predictive problems. However, it is more widely used in classification problems in the industry.

To evaluate any technique we generally look at 3 important aspects:

1. Ease to interpret output

2. Calculation time

3. Predictive Power

In KNN, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression:

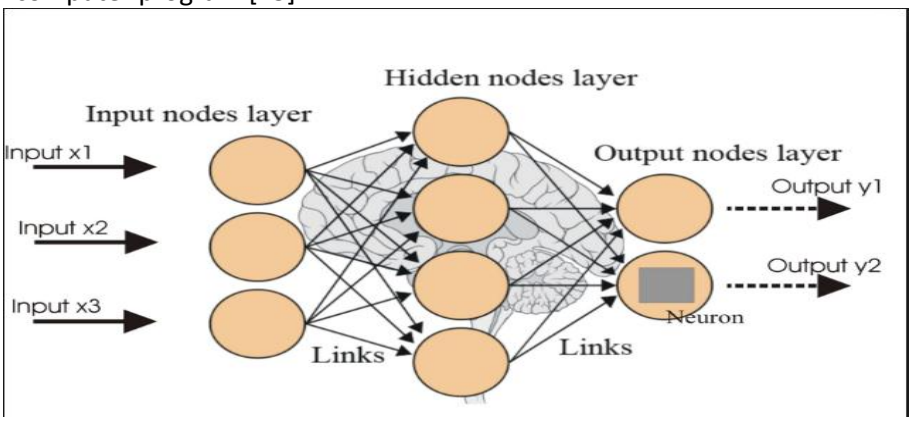
- In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor.
- In k-NN regression, the output is the property value for the object. This value is the average of the values of its k nearest neighbors.

Both for classification and regression, it can be useful to assign weight to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of 1/d, where d is the distance to the neighbor.
A shortcoming of the k-NN algorithm is that it is sensitive to the local structure of the data. The algorithm is not to be confused with k-means, another popular machine learning technique.

Among the various credit card fraud detection methods of supervised statistical pattern recognition, the K Nearest Neighbor rule achieves consistently high performance, without a priori assumptions about the distributions from which the training examples are drawn. K- Nearest neighbor based credit card fraud detection techniques require a distance or similar the measure defined between two data instances. [70][30]. In process of KNN, we classify any incoming transaction by calculating of nearest point to new incoming transaction. Then if the nearest neighbor be fraudulent, then the transaction indicates as a fraud. The value of K is used as, a small and odd to break the ties (typically 1, 3 or 5) [40]. Larger K values can help to reduce the effect of noisy data set. In this algorithm, distance between two data instances can be calculated in different ways. For continuous attributes, Euclidean distance is a good choice, [70][65]. For categorical attributes, a simple matching coefficient is often used. For multivariate data, distance is usually calculated for each attribute and then combined [65]. The performance of KNN algorithm can be improved by using a genetic algorithm for optimizing the distance metric. This technique required legitimate as well as fraudulent samples of data for training. It is fast technique along with high false alarm [40].

12. Neural net:

Artificial neural networks (ANNs) or connectionist systems are a computational model used in computer science and other research disciplines, which is based on a large collection of simple neural units (artificial neurons), loosely analogous to the observed behavior of a biological brain's axons. Each neural unit is connected with many others, and links can enhance or inhibit the activation state of adjoining neural units. Each individual neural unit computes using summation function. There may be a threshold function or limiting function on each connection and on the unit itself, such that the signal must surpass the limit before propagating to other neurons. These systems are self-learning and trained, rather than explicitly programmed, and excel in areas where the solution or feature detection is difficult to express in a traditional computer program.[13]



13. Conclusions

In this paper, we present a comparative study of five fraud detection methods based on credit card (Logistic Regression, Random Forest, KNN, Neural Network and Support Vector Machine). The main objective of this paper is to review methodology of different detection methods based on credit card. I have considered the most important parameter in different methods such as, accuracy, speed and cost. Comparison table was prepared in order to compare various credit card fraud detection mechanisms. All the techniques of credit card fraud detection described in the table 1 have its own strengths and weaknesses. From the result we can see that Logistic Regression gave the best performance as compared to Random Forest and then KNN.

Methods	Accuracy
Logistic Regression	97.84
Random Forest	97.43
KNN	96.79
Neural Network	93.21
Support Vector Machine	95.90

14. Future Work:

All these techniques of credit card fraud detection discussed in this paper, have its own weaknesses as well as strengths. Thus, this survey enables us to build a hybrid approach for developing some effective algorithms which can perform well for the classification problem with variable misclassification costs and with higher accuracy. Apart from this we can implement decision tree, Hidden Markov Model, Fuzzy Logic Method to check the accuracy of the model.

15. Acknowledgement:

I want to thank Prof Yiyuan She for his incredible support throughout the project.

16. References:

[1] Leila Seyedhossein, mahmoud reza hashemi “Mining Information from Credit Card Time Series for Timelier Fraud Detection”. IEEE-5th International Symposium on Telecommunications. (2010).

[2] Malak Alshawabkeh, Byunghyun Jang, David Kaeli.” Accelerating the Local Outlier Factor Algorithm on a GPU for Intrusion Detection Systems”. ACM; (2010) (104-110).

[3] Raghuveer Kancherla, Ratna Venkata, Anurag Verma “Behavioral Fraud Mitigation through Trend Offsets”. (2008).(1-11).

[4] E.W.T. Ngai, Yong Hu, Y.H. Wong, Yijun Chen, Xin Sun “The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature”. Elsevier-Decision Support Systems(2011). 50; (559–569).

[5]Masoumeh Zareapoor, Seeja. K.R, M. Afshar Alam “Analysis of credit card fraud Detection” International Journal of Computer Applications, August 2012

[6] R.C. Chen, T.S. Chen, C.C. Lin (“A new binary support vector system for increasing detection rate of credit card fraud”. International Journal of Pattern Recognition2006). 20 (2); (227–239).

[7] H. Leggatt, CyberSource “Online fraud to reach $4 billion”. BizReport, December 16. (2008).

[8] Kou, Y., Lu, C.-T., Sirwongwattana, S., Huang, Y.-P “Survey of fraud detection techniques”. In Proceedings of the IEEE International Conference on Networking, Sensing and Control. (2004).

[9] Phua, C, Lee, V., Smith, K., Gayler, R "A comprehensive survey of data mining-based fraud detection research". Artificial Intelligence Review. (2002).
[10] Serrano-Cinca, C "Self-organizing neural networks for financial diagnosis". Decision Support Systems, (1996). 17; (227–238).
[11] P.K. Chan, W. Fan, A.L. Prodromidis, S.J. Stolfo "Distributed Data Mining in Credit Card Fraud Detection". Data Mining; (1999). (67–74).
[12]Logit Data for Binary Data, http://data.princeton.edu/wws509/notes/c3.pdf
[13] https://en.wikipedia.org/wiki/Artificial_neural_network

Appendix:

Steps Followed:

- The data has 31 features out of which we removed *time* variables since the data didn't have any essence on the result.
- Logistic regression was implemented on the dataset.

```
## Call:
## glm(formula = Class ~ ., family = binomial(link = "logit"), data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.8098  -0.0290  -0.0187  -0.0119   4.6649
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.7896435  0.1890800 -46.486  < 2e-16 ***
## Amount       0.0008529  0.0004523   1.886 0.059313 .
## V1           0.0926755  0.0492430   1.882 0.059836 .
## V2           0.0109477  0.0711481   0.154 0.877711
## V3           0.0151143  0.0543248   0.278 0.780843
## V4           0.7230459  0.0909237   7.952 1.83e-15 ***
## V5           0.1008442  0.0804327   1.254 0.209925
## V6          -0.1066279  0.0916405  -1.164 0.244608
## V7          -0.1179042  0.0802491  -1.469 0.141771
## V8          -0.1797154  0.0351588  -5.112 3.20e-07 ***
## V9          -0.1794709  0.1353247  -1.326 0.184765
## V10         -0.8240539  0.1192371  -6.911 4.81e-12 ***
## V11         -0.0376685  0.0909044  -0.414 0.678600
## V12          0.0446235  0.0997015   0.448 0.654463
## V13         -0.3285088  0.0968672  -3.391 0.000696 ***
## V14         -0.5342260  0.0715685  -7.465 8.36e-14 ***
## V15         -0.1462628  0.1002128  -1.460 0.144421
## V16         -0.1339339  0.1620558  -0.826 0.408539
## V17          0.0082812  0.0809265   0.102 0.918495
## V18         -0.0610133  0.1622178  -0.376 0.706828
## V19          0.1387606  0.1182098   1.174 0.240455
## V20         -0.4319632  0.0993389  -4.348 1.37e-05 ***
## V21          0.4159169  0.0686564   6.058 1.38e-09 ***
## V22          0.6741539  0.1577365   4.274 1.92e-05 ***
## V23         -0.1399163  0.0671080  -2.085 0.037074 *
## V24          0.0255261  0.1777216   0.144 0.885793
## V25         -0.0109381  0.1488794  -0.073 0.941433
## V26         -0.0630006  0.2283904  -0.276 0.782666
## V27         -0.7675468  0.1489691  -5.152 2.57e-07 ***
## V28         -0.2853256  0.1033211  -2.762 0.005753 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5077.4  on 199365  degrees of freedom
## Residual deviance: 1552.6  on 199336  degrees of freedom
## AIC: 1612.6
##
## Number of Fisher Scoring iterations: 12
```

- Anova model was run on the model to check the difference between the null model and the residual model.

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Class
##
## Terms added sequentially (first to last)
##
##
##         Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                   199365     5077.4
## Amount   1     3.27    199364     5074.1 0.0704305 .
## V1       1   567.32    199363     4506.8 < 2.2e-16 ***
## V2       1   531.30    199362     3975.5 < 2.2e-16 ***
## V3       1   654.96    199361     3320.5 < 2.2e-16 ***
## V4       1   787.90    199360     2532.6 < 2.2e-16 ***
## V5       1    23.28    199359     2509.3 1.399e-06 ***
## V6       1    38.74    199358     2470.6 4.842e-10 ***
## V7       1    61.32    199357     2409.3 4.863e-15 ***
## V8       1    58.68    199356     2350.6 1.858e-14 ***
## V9       1    27.51    199355     2323.1 1.559e-07 ***
## V10      1   456.17    199354     1866.9 < 2.2e-16 ***
## V11      1    47.47    199353     1819.4 5.578e-12 ***
## V12      1    30.56    199352     1788.9 3.241e-08 ***
## V13      1    27.36    199351     1761.5 1.687e-07 ***
## V14      1   114.02    199350     1647.5 < 2.2e-16 ***
## V15      1     1.20    199349     1646.3 0.2725863
## V16      1    30.33    199348     1616.0 3.653e-08 ***
## V17      1     1.46    199347     1614.5 0.2273822
## V18      1     0.18    199346     1614.3 0.6706314
## V19      1     0.60    199345     1613.7 0.4380357
## V20      1     6.55    199344     1607.2 0.0104845 *
## V21      1     9.05    199343     1598.1 0.0026312 **
## V22      1    17.90    199342     1580.2 2.327e-05 ***
## V23      1     6.86    199341     1573.4 0.0088171 **
## V24      1     0.00    199340     1573.4 0.9635628
## V25      1     0.04    199339     1573.3 0.8326281
## V26      1     0.19    199338     1573.1 0.6656322
## V27      1    12.55    199337     1560.6 0.0003964 ***
## V28      1     8.00    199336     1552.6 0.0046828 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Sensitivity –Specificity Curve:

- Accuracy of each of the model was compared and we found that logistic regression was the most accurate model of all the methods which was implemented.