Prediction of Credit Card Default and Fraud using Machine Learning Techniques

Report submitted in fulfillment of the requirements
for the Exploratory Project of

Second Year B.Tech.

by

Karthik Kumar,Pranjal Jain, Rathi Saurabh Bhagirath, Vinay Jaisinghani

Under the guidance of

Dr.S.K.Singh

Department of Computer Science and Engineering
INDIAN INSTITUTE OF TECHNOLOGY (BHU) VARANASI
Varanasi 221005, India
May 2017

Dedicated to

Our parents, teachers, friends and

our love for Computer Science.

Declaration

We certify that

1. The work contained in this report is original and has been done by ourself and the general supervision of our supervisor.

2. The work has not been submitted for any project.

3. Whenever we have used materials (data, theoretical analysis, results) from other sources, we have given due credit to them by citing them in the text of the thesis and giving their details in the references.

4. Whenever we have quoted written materials from other sources, we have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Place: IIT (BHU) Varanasi

Date:

Karthik Kumar,Pranjal Jain, Rathi Saurabh Bhagir
Vinay Jaisinghani
B.Tech. Students
Department of Computer Science and Engineering,
Indian Institute of Technology (BHU) Varanasi,
Varanasi, INDIA 221005.

Certificate

*This is to certify that the work contained in this report entitled "**Prediction of Credit Card Default and Fraud using Machine Learning Techniques**" being submitted by **Karthik Kumar,Pranjal Jain, Rathi Saurabh Bhagirath, Vinay Jaisinghani** (**Roll No. 15075022, 15075038, 15075041, 15075056**), carried out in the Department of Computer Science and Engineering, Indian Institute of Technology (BHU) Varanasi, is a bona fide work of our supervision.*

Place: IIT (BHU) Varanasi
Date:

Dr. S.K.Singh
Department of Computer Science and Engineering,
Indian Institute of Technology (BHU) Varanasi,
Varanasi, INDIA 221005.

Acknowledgment

We would like to express our sincere gratitude to Dr.S.K.Singh, our families and our friends for their constant support and motivation.

Place: IIT (BHU) Varanasi

Date: **Karthik Kumar,Pranjal Jain, Rathi Saurabh Bhagirath, Vinay Jaisinghani**

Abstract

The evaluation of credit risk is an important problem in the field of banking in today's world. The research is aimed at comparing various data mining techniques to predict default of credit card clients. This will help credit providers identify fraudulent customers before providing credit. In this project we used six state-of-the-art methods namely Logistic Regression, K-nearest neighbours, Neural Networks, Naive Bayesian, Decision Trees and Support Vector Machines and proposed a technique to generate more and better features, given enough features. Results have been evaluated for two data sets, one of Taiwan from 2005 and other of Europe from 2013.

Contents

List of Figures

Chapter 1

Introduction

1.1 Overview

In the recent times, there has been too much involvement of technology in every sphere of life. Be it banks, finance, quantitive finance, biology, etc, the amount of data is immense and the use of this data to predict few properties has become essential. Banking is one such field, where from years to years amount of data has been collected and now all these data is being used to predict factors like default of credit cards, default of loans, prediction of share markets for investment, corporate banking, etc.

We took the project of using machine learning methods for predicting the default of credit cards. We analysed different machine learning techniques and compared them with each other. Our work included an algorithm proposed by us which gives better accuracy.

1.2 Motivation of the Research Work

Main motivation for our research work has been our fascination for Machine Learning and we wanted to do something for the greater good. Our work will help banks reduce

giving loans to people who will take money from them and not return to them, thus not wasting money.

Chapter 2

Project Work

2.1 Data Description

In our project, we have used two datasets. One being the details of customers of a bank in Taiwan which has 24 attributes like the amount of given credit, gender of the person, education, marital status, age and history of past payments. The dataset has 30000 instances and was collected in 2005. [2]

The second dataset consists of transactions made by credit cards in September 2013 by European credit card holders. There are over 284 thousand instances out of which 492 were frauds. The dataset consists of only numerical input variables which is a result of Principal Component Analysis and only two attributes could not be changed which are time and amount. [3]

2.2 Data Analysis

The first dataset on analysis showed that approximately 70 % data corresponds to non default transactions and 30% corresponded to defaulted transactions (shown in Fig 1). This is a bit skewed data but it can be neglected and our algorithms can be applied.

The second dataset on the other hand was highly skewed with only 492 out of

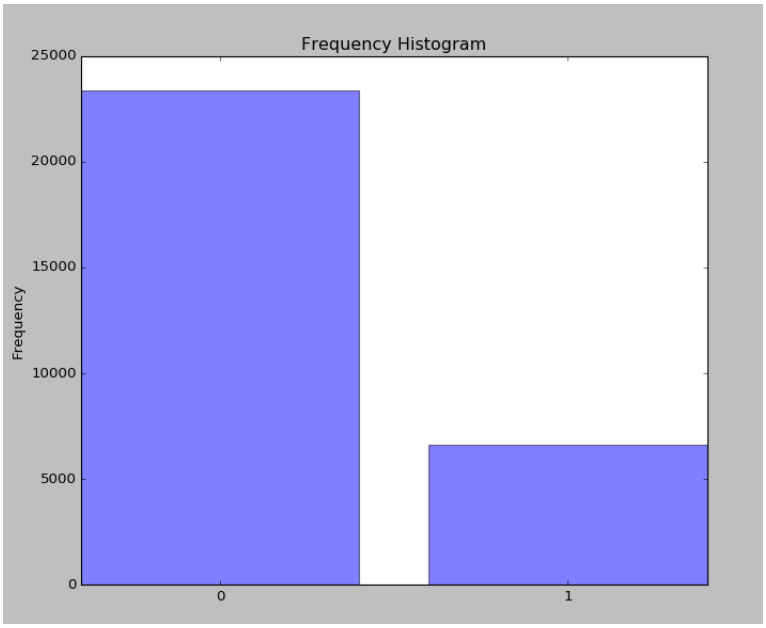

Figure 2.1 Frequency Historgram of first dataset

2,84,000 being defaulted transactions. That amounts to .2% of data being "1" and others being "0"(shown in Fig 2). This much skewedness had to be handled as just predicting "0" gives us an accuracy of 99.8% but we have predicted all defaulted transations wrong which is a very big problem. To address this skewedness, we need to come up with few techniques. The techniques which we can use to resample are

1. Collecting more data (which is not possible now)
2. Over-Sampling, which is adding copies of underrepresented class
3. Under-Sampling, which is removing copies of overrepresented class
4. SMOTE- Synthetic Minority Over-Sampling Technique which is a mixture of both Under and Over Sampling.

We tried Under-Sampling wherein we randomly removed the "0"s and reduced the total instances to around 1350. Now the "1"s represented approximately 35% of dataset [Refer Fig 3 for clearer analysis].

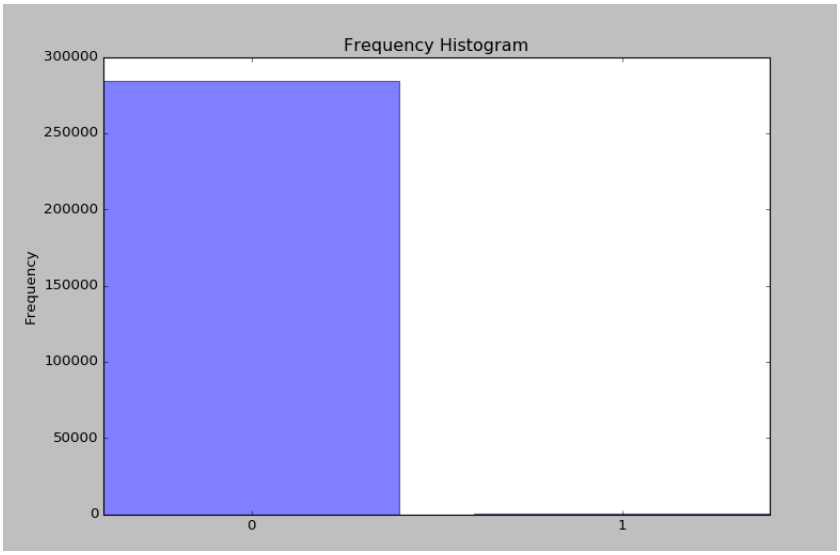

Figure 2.2 Frequency Historgram of Second dataset before resampling

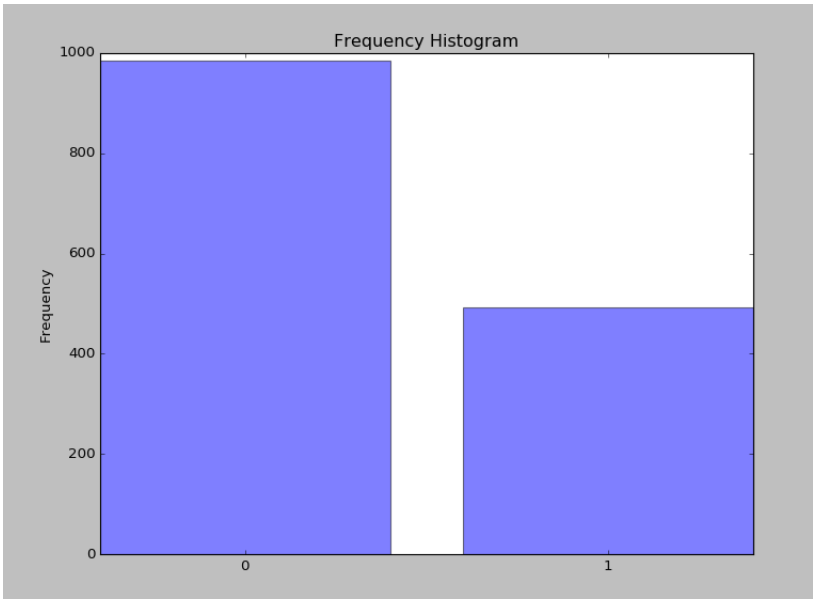

Figure 2.3 Frequency Histogram of Second dataset after resampling

Thus after resampling, we can apply our machine learning algorithms for the second dataset and find our results. To analyse further, we decided to apply our machine learning algorithms on the dataset before Under-Sampling so that we check our efficiency in predicting the "1"s. Now we'll discuss the algorithms and techniques we applied and compare them for both datasets.For every algorithm, we have used 70% of training data for training and rest 30% for the testing of our model.

2.3 Logistic Regression

In Machine Learning, logistic regression is a statistical method wherein we try to approximate the hypothesis by minimising the cost associated with the hypothesis.In this form of regressive analysis, we approximate the hypothesis to be a linear model of the features (independent variables). Logistic Regression is used for predicting binary dependent variables rather than a continuous outcome. Here in our project the binary dependent variable is that the credit has been defaulted or not. If not defaulted the binary variable is '0' else the binary variable is '1'. [4]

In this algorithm, here x_{ij} are the features and y_i is the result for the corresponding training data (i ranges from $[1, \text{Number of instances}]$ and j ranges from $[1, \text{Number of attributes}]$). Now we assume

$$p_\theta(x) = \Sigma\theta_i x_i \quad [5]$$
$$p_\theta(x) = \theta^T x \quad [5]$$

(ie) linear function of x_i where $p_\theta(x)$ is the prediction . Now in our classification problem y can only be 1 or 0. So to correct the y, we introduce a Sigmoid function

$$q(x) = \frac{1}{1+e^{-x}} \quad [5]$$

. The speciality of the function is it's range is $[0, 1]$ and with the value of 0.5 at $x = 0$. Now is we replace $p_\theta(x)$ as

$$p_\theta(x) = q(\theta^T x) \text{ [5]}$$

We initialise all θ_i to 0 initially. Now let the cost function

$$W(y|x;\theta) = (p_\theta(x))^y(1-p_\theta(x))^{1-y} \quad [5]$$

. This function $W(y)$ tries to convey the difference between our prediction for the training data and the actual value. Now to fit the θ , we will maximise the likelihood

$$J(\theta) = \Pi_{i=1}^m W(y|x;\theta) \quad [5]$$

We know it is better to maximise the log likelihood. Thus taking log

$$log(B(\theta)) = \Sigma_{i=1}^m y^{(i)} log(p(x^{(i)})) + (1-y^{(i)}) log(1-p(x^{(i)}))$$

To maximise the log likelihood, we use the **Gradient Descent** algorithm which is Repeat until convergence {

$$\theta_j := \theta_j + \alpha \frac{\partial log(B(\theta))}{\partial \theta_j} \text{ (for every } j)$$

}

Here α is called learning rate and by differentiating we get,

$$\theta_j := \theta_j + \alpha(y - p_\theta(x))x \text{ (for every } j) \quad [5]$$

Implementing this algorithm, we get results(classification report) as shown below for dataset 1

	Precision	Recall	f1-score
0	0.83	0.98	0.90
1	0.72	0.21	0.33
avg	0.81	0.82	0.78

The accuracy is 82.25

The results for the dataset 2 before resampling

	Precision	Recall	f1-score
0	1.0	1.0	1.0
1	0.75	0.56	0.65
avg	1.0	1.0	1.0

The accuracy is 92.92

This table has to be analysed carefully. It says that the f1-score is 1.0 but the main thing is the data set is highly skewed. Instead from this table,if we consider the the f1-score of predicting "1"s , it gives us an f1-score of 0.65 which seems to be a bit legitimate score and we can consider this to be our result.

Next the results for the second dataset after Under-Sampling

	Precision	Recall	f1-score
0	0.96	0.99	0.97
1	0.97	0.92	0.94
avg	0.96	0.96	0.96

The accuracy is 95.17

After application of Under-Sampling, the absolute values of f1-score has decreased(which had to happen) and we get a f1-score of .98 which is very good score for this problem.

2.4 Neural Networks

Neural Networks is machine learning algorithm used for non-linear classification. Basically, if we want to take into consideration two degree terms(e.g. $X_1^2, X_1X_2, X_1X_3, X_2^2$) as features, we will have to consider n^2 two degree terms. This will lead to tremendous increase in number of features and applying gradient descent algorithm to minimise cost function will be very costly(in terms of computations). In Neural Networks we consider hidden layers between input and output layer, for the each unit in the layer

there is a mapping from each unit of previous layer having a particular weightage. [6]

In this algorithm, we randomly initialise the mapping terms so called θ_{ij}^{l} terms (l is the layer from which mapping is done, i is the cell from which mapping is done and j is the cell to which mapping is done in $(i+1)^{th}$ layer). Compute the terms in next layer and so on until the output layer. We calculate the cost function using the predicted output. The error in output helps in calculating errors from previous layer which further helps in calculating error from previous layers. These errors are used while calculating gradient of cost function with θ_{ij}^{l}. Gradient Descent method is used to optimise cost function. In each such iteration of forward and backward propagation we improve the values of θ_{ij}^{l} and so the output. [6] Implementing this

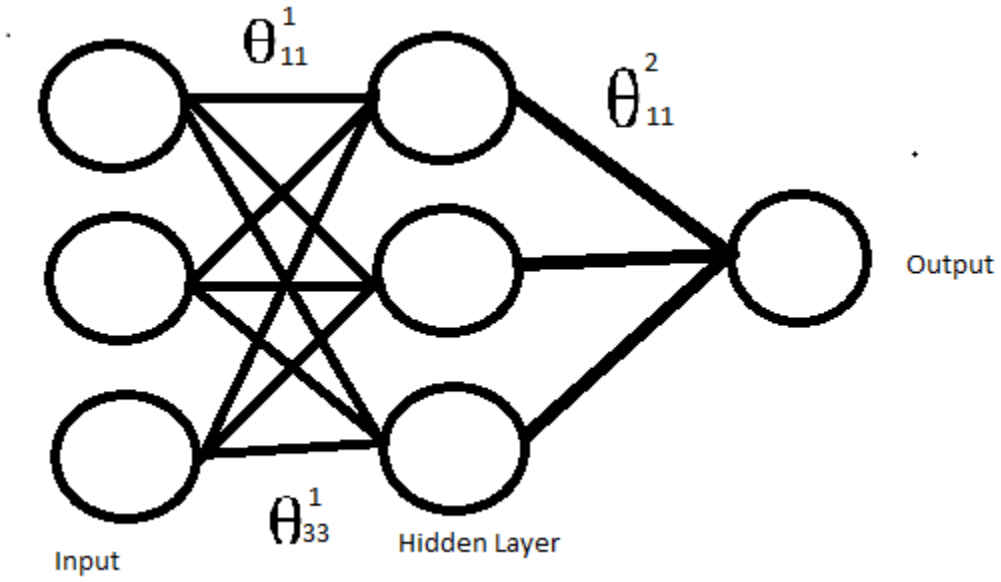

Figure 2.4 Neural Networks

algorithm, we get results(classification report) as shown below for dataset 1

	Precision	Recall	f1-score
0	0.82	0.98	0.90
1	0.72	0.17	0.27
avg	0.80	0.82	0.77

The accuracy is 81.66

The results for the dataset 2 before resampling

	Precision	Recall	f1-score
0	1.0	1.0	1.0
1	0.96	0.70	0.81
avg	1.0	1.0	1.0

The accuracy is 99.95

This table has to be analysed carefully. It says that the f1-score is 1.0 but the main thing is the data set is highly skewed. Instead from this table,if we consider the the f1-score of predicting "1"s , it gives us an f1-score of 0.65 which seems to be a bit legitimate score and we can consider this to be our result.

Next the results for the second dataset after Under-Sampling

	Precision	Recall	f1-score
0	0.96	0.96	0.96
1	0.93	0.93	0.93
avg	0.95	0.95	0.95

The accuracy is 95.21

2.5 Naive Bayesian Algorithm

It is a Machine Learning technique based on Bayes theorem with an assumption of independence among predictors. Naive Bayesian technique assumes that all the features are independent of each other. Bayes theorem mathematically implies

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad [7]$$

where

$P(A|B)$ is the probability that event A occurs given event B

$P(A)$ is the probability of occurence of event A

$P(B)$ is the probability of occurence of event A

$P(A|B)$ is the probability that event B occurs given event A

Now here A will be the event that y be "1" or "0". and B be the events that each of the features have taken the value x_{ij} (ie) for each instance in the test data, we will find the probability of it being 1 and the probability of it being 0 given all the x_{ij} have occured. Then the assign $y = 1$ if probability of "1" is larger ,else "0". The question now arises is how to calculate the probability?

Now if the features take discrete values, we can then find the probability from the formula

$$P(A) = \frac{cnt(A)}{totalcnt}$$

But if the data is continuous instead of discrete, we apply some standard probability functions. Here we will use Gaussian(Normal) Probability distribution function for the prediction of probability.

$$P(x) = \frac{1}{\sqrt{2\sigma^2\pi}} e^{\frac{(x-\mu)^2}{\sqrt{2}\sigma}}$$

Here

$P(x)$ is the probability of x

μ is the Mean of the instances

σ is the Standard Deviation of the instances. [7]

Now we implement this algorithm for dataset 1 and find the following results.

	Precision	Recall	f1-score
0	0.79	0.98	0.90
1	0.72	0.21	0.33
avg	0.81	0.82	0.78

The accuracy is 75.07

Now when we implemented this algorithm for the algorithm for the second dataset before the application of Under-Sampling, we got the following results.

	Precision	Recall	f1-score
0	0.98	1.00	0.99
1	0.80	0.04	0.08
avg	0.97	0.98	0.97

The accuracy is 97.77

What must be viewed here is that the data is skewed and our algorithm is working very poorly in predicting the default conditions.

Now after we apply Under-Sampling, we got the following results.

	Precision	Recall	f1-score
0	0.96	0.94	0.95
1	0.88	0.93	0.90
avg	0.94	0.94	0.94

The accuracy is 94.05

Although the overall average has been decreased but we have succeeded in predicting fraud cases quite well and this model will be quite successful.

2.6 K-Nearest Neighbours

K-nearest neighbours is machine learning algorithm widely used for classification problems. As the name suggests, to classify an test example, we compute it's distance from all the training examples. And divide the distances into different sets corresponding to the set which training sample belongs. The set which has least sum of k smallest distances will be the required classification for the sample. Here

$$distance_j^2 = \Sigma_{i=1}^{m}(Train_{ij} - Test_i)^2$$

where $distance_j$ represents distance of test example with the j^{th} training example.
m represents the number of features in the dataset
$Train_{ij}$ represents i^{th} feature of the j^{th} training example
$Test_i$ represents i^{th} feature of test example.

How we choose value k? Initially, error starts decreasing by increasing value of k until a value comes which has the least error and then again error starts increasing with increasing value of k. If we test on training sample itself error will increase on increasing value of k from beginning itself. [8]

Results for Dataset 1.

	Precision	Recall	f1-score
0	0.84	0.97	0.90
1	0.68	0.29	0.40
avg	0.81	0.83	0.80

The accuracy is 82.76

Now when we implemented this algorithm for the algorithm for the second dataset before the application of Under-Sampling, we got the following results.

	Precision	Recall	f1-score
0	1.00	1.00	1.00
1	0.96	0.73	0.83
avg	1.00	1.00	1.00

The accuracy is 99.96

What must be viewed here is that the data is skewed and our algorithm is working very poorly in predicting the default conditions.

Now after we apply Under-Sampling, we got the following results.

	Precision	Recall	f1-score
0	0.92	0.99	0.96
1	0.98	0.84	0.90
avg	0.94	0.94	0.94

The accuracy is 94.01

Although the overall average has been decreased but we have succeeded in predicting fraud cases quite well and this model will be quite successful.

2.7 Decision Trees

Decision trees build regression models based on a structure of the tree. In Machine Learning, the decision is about a single predictor (attributes). In this algorithm, the main concept is to choose an attribute and make decisions on that and distribute the dataset according to the decision. Then recursively on the distributed datasets apply decisions on the other attributes and finally we reach a point that the dataset becomes completely homogeneous in one category.

The task is tough because we don't know upon which attribute the first decision has to be taken. Thus we solve this greedily. Our final destination is smaller and more

homogeneous datasets. Thus at every step, we will try to maximise the homogeneity of the resulting datasets. Here the measure of homogeneity mathematically is the entropy of the dataset. In order to increase the homogeneity, we need to reduce to entropy of the datasets. Thus we have entropy of a random variable X [9]

$$Entropy(X) = -\sum P(x_k) \log_2 P(x_k) \quad [9]$$

where x_k are the values of the random variable X.

$P(x_k)$ is the probability of the occurence of x_k for the variable X.

Thus we take the initial dataset, find it's entropy. Divide the dataset according to every attribute available and then find the resulting entropy. The attribute which results with the least entropy is then chosen and the dataset is divided accordingly.Then the datasets which are formed from the above said action are solved recursively.

Now for the results of this technique.

	Precision	Recall	f1-score
0	0.96	0.85	0.90
1	0.34	0.67	0.45
avg	0.89	0.83	0.85

The accuracy is 82.25

Now when we implemented this algorithm for the algorithm for the second dataset before the application of Under-Sampling, we got the following results.

	Precision	Recall	f1-score
0	1.00	1.00	1.00
1	0.64	0.67	0.65
avg	1.00	1.00	1.00

What must be viewed here is that the data is skewed and our algorithm is working very poorly in predicting the default conditions.

Now after we apply Under-Sampling, we got the following results.

	Precision	Recall	f1-score
0	0.95	0.93	0.94
1	0.88	0.91	0.89
avg	0.93	0.93	0.93

Although the overall average has been decreased but we have succeeded in predicting fraud cases quite well and this model will be quite successful.

2.8 Support Vector Machines

SVM is a supervised learning algorithm. It is used to classify our data in positive and negative sets. Given a set of data, we need a separating plane that divides the data in two sets, specifically our classes.(Refer Fig 5)

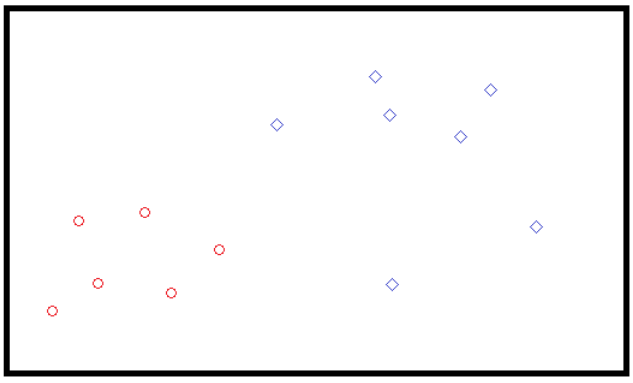

Figure 2.5 Positive and Negative data sets

A hyperplane divides the given data into two classes. It can be a point, line, plane or hyperplane, depending upon whether the data is in one dimension, two dimension, three or more.(Refer Fig 6) It seems very clear that there may be many

such hyperplanes separating our data. So our task lies down at choosing one such hyperplane which gives best results. [1]

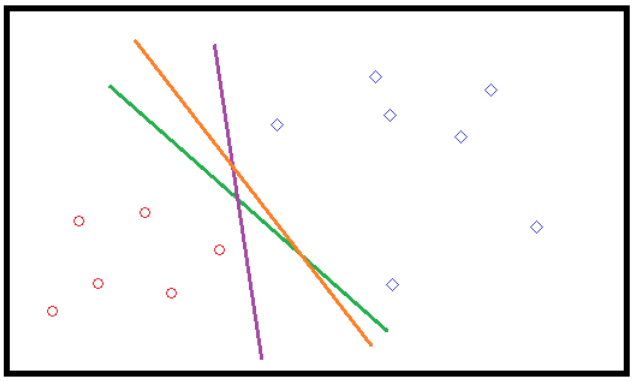

Figure 2.6 Separating Hyperplanes [1]

The distance between the hyperplane and the closest point to the hyperplane is defined as the Margin. Clearly, no point lies inside the margin.(refer Fig 7)

The hyperplane which has the largest margin is chosen and it eventually gives the best result.

As we saw in logistic regression, we chose a decision boundary by our constraint

$$\min_{\theta} \sum (y_i \log(h_\theta(i)) + (1 - y_i) \log(1 - h_\theta(i)))$$

.

If we penalise our parameters, our equation modifies as following

$$\sum (y_i \log(h_\theta(i) + (1 - y_i) \log(1 - h_\theta(i)))) + \frac{\lambda}{2m} \sum (\theta_i^2)$$

Lets call our cost function $A + \lambda B$.

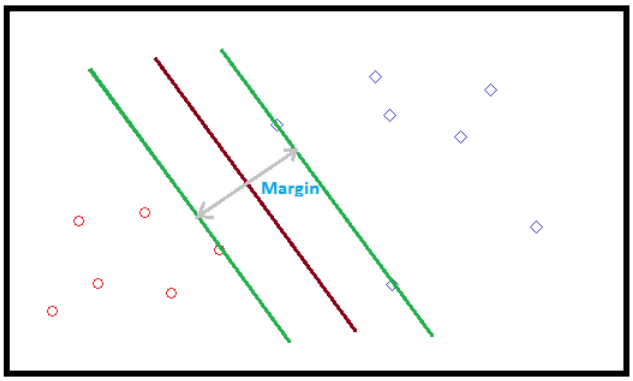

Figure 2.7 Margin [1]

In SVM, our cost function becomes $\alpha A + B$. Instead of $\log(h(\theta_i))$, we use the term CostA, and instead of $\log(1 - h(\theta_i))$ we use CostB.

Finally, we are required to minimize $\alpha \sum (y_i CostA_i + (1 - y_i) CostB_i) + \sum \theta_i^2$

Now, we keep our α large enough such that the term αA in our cost function does not contribute much to the error. So we are ultimately left at minimizing Logistic Regression $\sum \theta_i^2$ for some θ, such that, θ_i belongs to theta

$$||\theta|| = \sqrt{\sum \theta_i^2}$$

$$||\theta||^2 = \sum \theta_i^2$$

Hence, We are to minimize $||\theta||^2$ for some θ, i.e., we need to minimize $||\theta||$

The equation of our hyperplane is $\theta X = 0$ where $x_0 = 1$ and $x_1, x_2, x_3, \cdots x_n$ are the variables. Further, θ_is are our parameters.

In case of logistic regression and SVM, we may have less number of features making our model prone to under fitting. In order to overcome under fitting, one way is to increase the features. Now we implement this algorithm for data set 1 and find the

following results.

	Precision	Recall	f1-score
0	0.85	0.97	0.90
1	0.72	0.32	0.45
avg	0.82	0.84	0.81

The accuracy is 83.57

Here now we could not apply this technique to the data set 2 before Under-Sampling because of the length of computation. It would take days to complete so we did not use this technique.

Now after we apply Under-Sampling, we got the following results.

	Precision	Recall	f1-score
0	0.94	0.99	0.96
1	0.98	0.88	0.93
avg	0.95	0.95	0.95

The accuracy is 95.21

Although the overall average has been decreased but we have succeeded in predicting fraud cases quite well and this model will be quite successful

2.9 Proposed Methodology

. .0 We have considered the already given feature tuples as special points. That is, the values Xs of a training data is a tuple, and we have m such tuples because there are a total of m training samples.

Now in order to have more features, say k, we have applied k-means algorithm to find the centroids of k clusters formed out of these tuples.

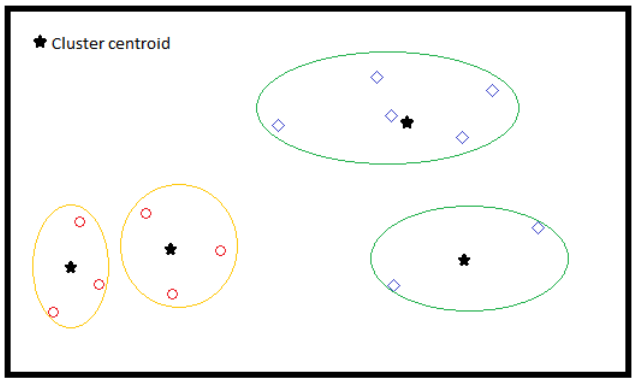

Figure 2.8 Clusters

We have defined positive and negative clusters by having only positive points in positive cluster and negative points in negative cluster. We have proportionate number of positive and negative clusters to as there are number of positive and negative data points. If there are k_1 cluster centroids of positive data points and k_2 of negative, then $k = k_1 + k_2$. [6]

These cluster centroids can now define the closeness of our given sample points. Each centroid now functions as a feature which says how close is a sample from that centroid. We use the function

$$e^{-\frac{||x-l||^2}{2\sigma^2}}$$

Our k features can now be defined as

$$f_i = e^{-\frac{||x-l_i||^2}{2\sigma^2}}$$

where l_i is the cluster centroid obtained by k-means algorithm on the tuples that we got from the Xs of the input data.

This technique not only reduces the risk of underfitting by increasing the number of features, but also deals with the trouble of having a non-separable data in case of SVM. SVM technique is supposed to be applied on linearly separable data. This technique makes our data linearly separable. Although kernels can be used in case of

SVM, this, infact, is a special case of kernels.

We applied this technique alongside SVM and Logistic Regression. For SVM with this new technique on Dataset 1, we got the following results.

	Precision	Recall	f1-score
0	0.85	0.96	0.90
1	0.69	0.32	0.43
avg	0.81	0.83	0.81

The accuracy is 83.11

For Logistic Regression with this new technique on Dataset 1, we got the following results.

	Precision	Recall	f1-score
0	0.84	0.97	0.90
1	0.70	0.30	0.42
avg	0.81	0.83	0.80

The accuracy is 83.03

For SVM with this new technique on Dataset 2 after Under-Sampling, we got the following results.

	Precision	Recall	f1-score
0	0.96	0.98	0.97
1	0.96	0.92	0.94
avg	0.96	0.96	0.96

The accuracy is 95.60

For Logistic Regression with this new technique on Dataset 2 after Under-Sampling, we got the following results.

	Precision	Recall	f1-score
0	0.96	0.99	0.97
1	0.97	0.92	0.94
avg	0.96	0.96	0.96

The accuracy is 96.17

Chapter 3

Conclusions and Discussion

3.1 Plots

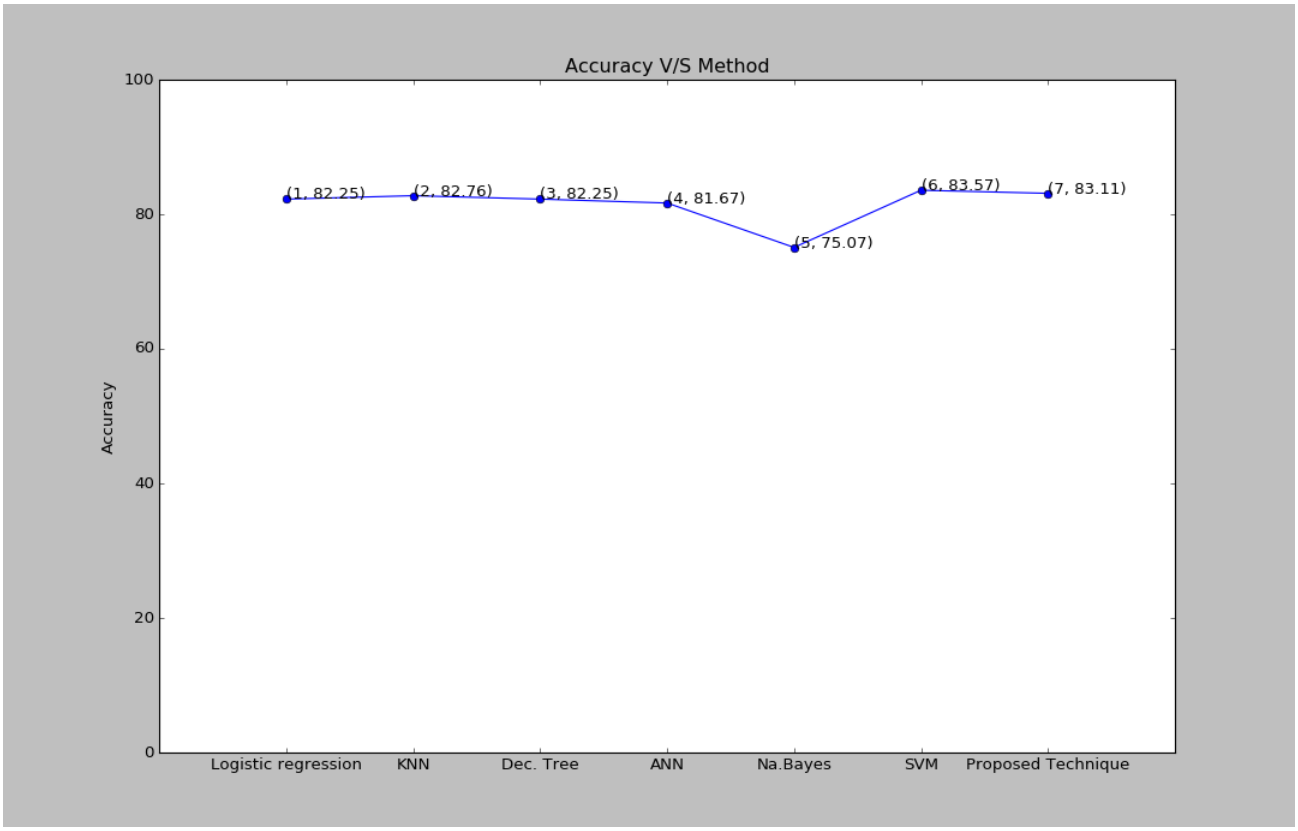

Figure 3.1 Accuracies v/s Methods for Data set 1

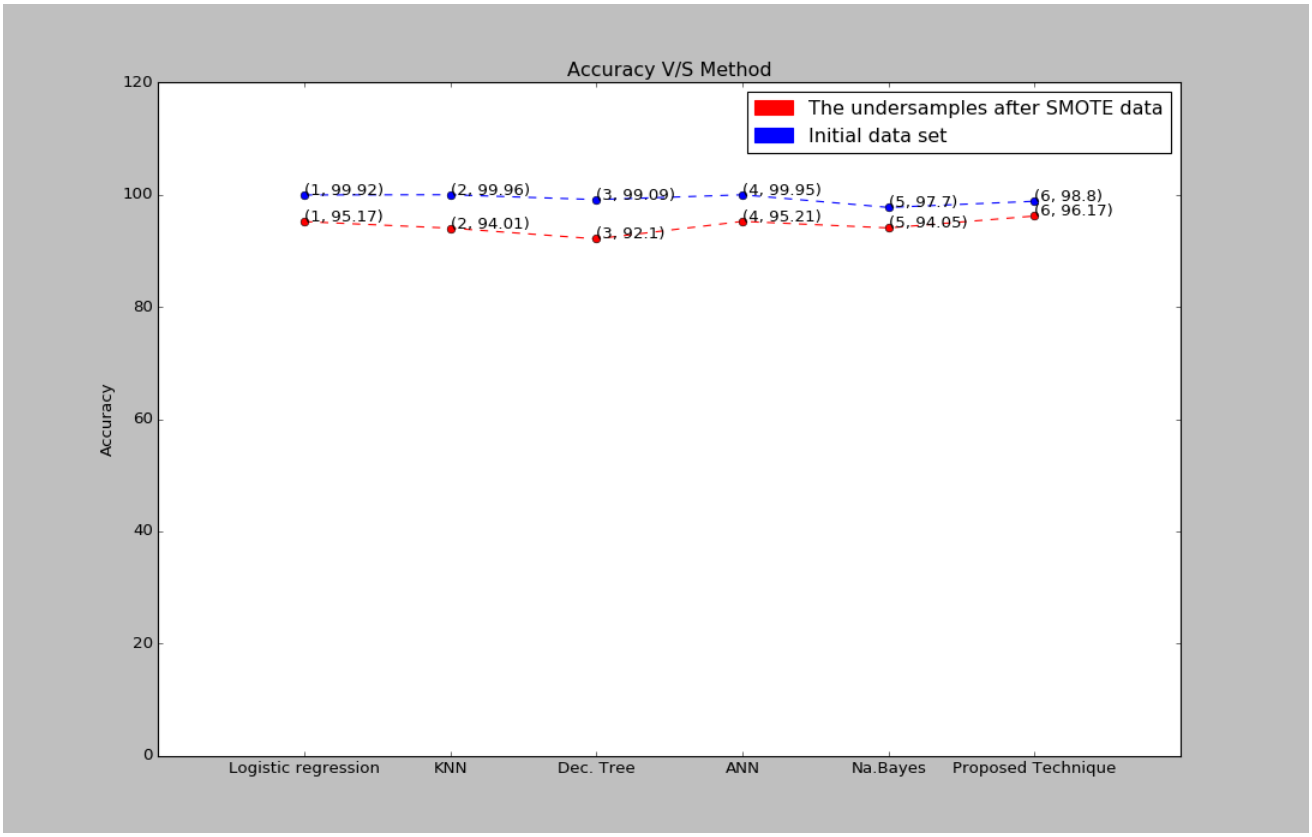

Figure 3.2 Accuracies v/s Methos for Dataset 2

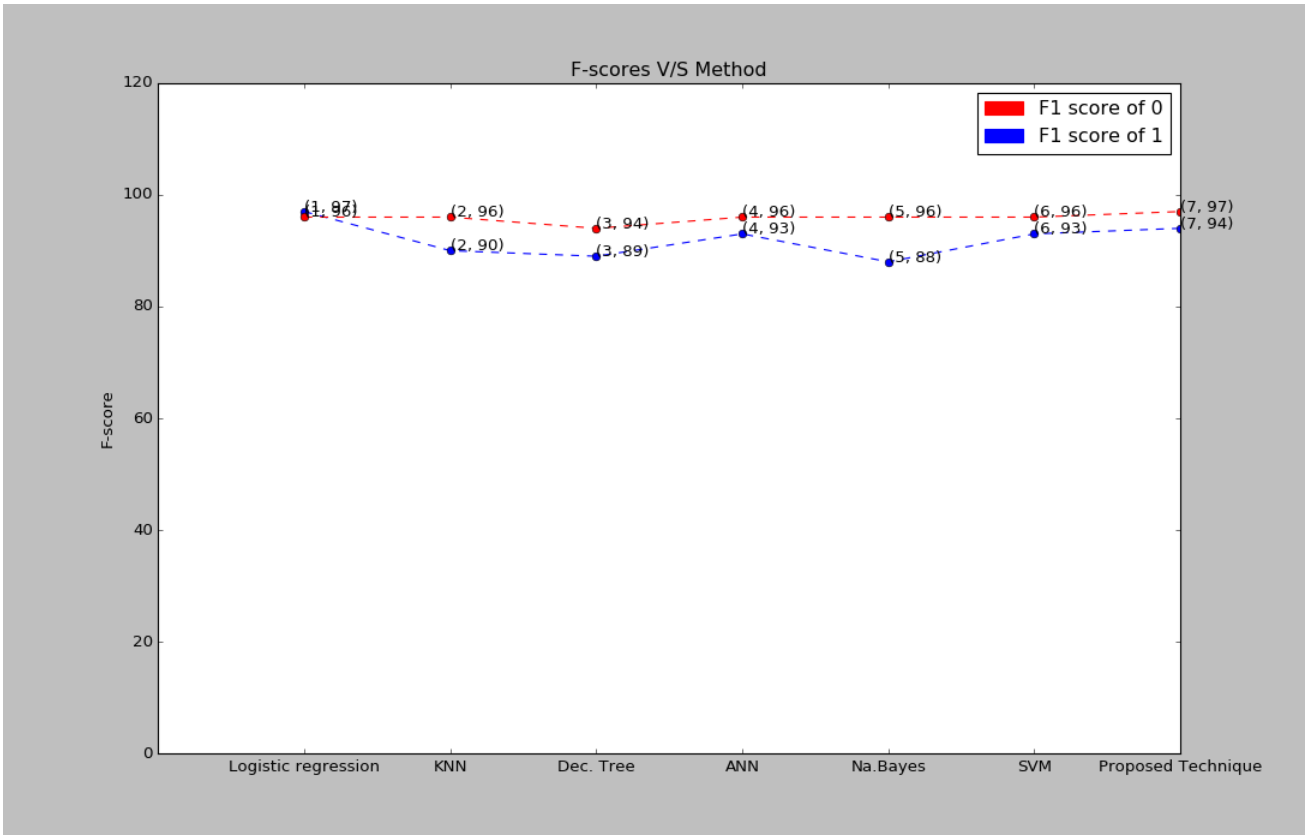

Figure 3.3 F1-Scores of 0 and 1 v/s Methods for Dataset 2

3.2 Conclusions

This report gives rise to a number of important conclusions.

- In our first data set, we applied all techniques directly to the the given data and found SVM giving us the highest accuracy followed by our proposed methodology. The classification reports among all methods showed very less differences in the accuracies.

- In the second data set, we applied the techniques before the re-sampling and found our models not predicting the ones properly. Thus some change had to be done in our model. Thus we under-sampled the data and drastically reduced the data set's size to 1300 and then we applied our machine learning techniques.

- After under-sampling, the overall results were lesser than that of the initial data set but the accuracy and $f1$ score of predicting 1 increased a lot and thus increasing the quality of our models.

- Again in this data set, our proposed technique of applying K-means with SVM performs slightly better than other techniques.

- Thus in our report, we applied all machine learning techniques and then compared them to find the best for the data sets.

3.3 Further Research

In our report, we just applied few Machine Learning Techniques and compared. There are many more Advanced Machine Learning Algorithms like Random forests, Boltzmann Algorithms, etc.

In both the data sets, Artificial Neural Networks did not perform very badly. Thus this research work can be extended with the application of Deep Learning Algorithms.

Bibliography

[1] L. B. DURGESH K. SRIVASTAVA, "Data classification using support vector machine," *Journal of Theoretical and Applied Information Technology*, vol. 12, no. 1, 2010.

[2] I.-C. Yeh and C.-h. Lien, "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients," *Expert Systems with Applications*, vol. 36, no. 2, pp. 2473–2480, 2009.

[3] A. Dal Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating probability with undersampling for unbalanced classification," in *Computational Intelligence, 2015 IEEE Symposium Series on*. IEEE, 2015, pp. 159–166.

[4] A. S. Al-Ghamdi, "Using logistic regression to estimate the influence of accident factors on accident severity," *Accident Analysis & Prevention*, vol. 34, no. 6, pp. 729–741, 2002.

[5] S. U. CS229. (2015) CS229 supervised machine learning techniques. [Online]. Available: http://cs229.stanford.edu/notes/cs229-notes1.pdf

[6] S. B. Maind, P. Wankar *et al.*, "Research paper on basic of artificial neural network," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 2, no. 1, pp. 96–100, 2014.

[7] I. Rish, "An empirical study of the naive bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, no. 22. IBM New York, 2001, pp. 41–46.

[8] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "Knn model-based approach in classification," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, 2003, pp. 986–996.

[9] L. Li and X. Zhang, "Study of data mining algorithm based on decision tree," *International Conference on Compute and Data Analysis*, 2010.