The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients

A Project report submitted as a part of the course

Statistics 1

in

B.Math 1st year

Indian Statistical Institute

Banglore Centre

Submitted by:

Aditya Garg (bmat2201)

Amit Kumar Baistha (bmat2203)

Bhavesh Pandya (bmat2215)

Pragalbh Kumar Awasthi (bmat2228)

Abstract

In this project we have tried to replicate the results obtained in the paper 'The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients' authored by I-Cheng Yeh and Che-hui Lien.Apart from the models used by the original authors we have applied some additional algorithms and tried to compare their accuracy in predicting default of credit card clinents.

Contents

1 Introduction

In 2006, card-issuing banks in Taiwan over-issued cash and credit cards wanting to increase market share. However, these cards ended up being overused and the card holders, failing to make even the minimum payment due over extended periods, accumulated heavy credit-card debt. This caused a major blow to consumer finance confidence and banks became less willing to extend new credit.

Typically, when a cardholder is unable to make the minimum payment due for six months in a row, the account defaults and the bank cancels its agreement with the cardholder. Such defaulted credit cards tend to negatively impact banks by reducing their profit. Therefore, in a robust financial system, the identification of cardholders who are at a high risk of default is essential to avoid situations of crisis both for cardholders and the banks.

We analyzed the payment data of 30,000 credit card holders of an important bank in Taiwan out of which 6636 (22.12%) were cardholders with default payments. For each cardholder, 23 features (X_1 to X_{23}) and a binary variable Y for default payment (Yes =1, No =0) were recorded. To deal with the problem of class imbalance, an oversampling technique called SMOTE is implemented. The data was used to train, test and compare various machine-learning binary classifier models that predict the risk probability of a given cardholder based on the values of feature variables. Out of many, one of the ways used to assess the performance of a classifier model was based on the novel "Sorting Smoothing Method"[1] which estimated real probabilities for a model. We then compared the predicted probabilities against these estimated real probabilities for each model through linear regression. Following is a brief outline of our report:

- Literature Review
- Description of Data
- Methodology
- Results
- Analysis of Sorting Smoothing Method

2 Literature Review

2.1 Description of Models

2.1.1 k-Nearest Neighbours

k-nearest neighbours is one of the simplest non-parametric learning methods used for both classification and regression problems. Let P be the target point to be classified for which values of various features are known. k nearest neighbours of P are found from the dataset based on some fixed metric defined on the feature space, and P is assigned the most frequent class among these k neighbours. Since the algorithm is sensitive to the local structure of data, the choice of metric and the neighbour order (i.e. the value of k) based on the sample

size n, become highly significant. Generally, larger values of k reduce the effect of noise or irrelevant features on the prediction, but make boundaries between classes less distinct. The proper divergence of $\{k_n\}$, and convergence of $\{\frac{k_n}{n}\}$ to zero ensure strong consistency. The basic technique of "majority voting" is not good enough in situations of extreme class imbalance because the points of the majority class dominate the prediction merely because they tend to be common among the k neighbours due to their large number.

2.1.2 Naive Bayes Classifier

Naive Bayes is a classifiaction algorithm based on the bayes theorem that assumes independence of features involved in the classification problem.Let $x_1, x_2, \ldots, x_n$ be d-dimensional feature vectors in the training set with output vectors y_i taking values 0 or 1.Our main goal is to find $\mathbb{P}(y|x)$ for any given vector x by using the training data.For that purpose the Bayes theorem is used

$$\mathbb{P}(y|x) = \frac{\mathbb{P}(x|y)\mathbb{P}(y)}{\mathbb{P}(x)}$$

Now $\mathbb{P}(x)$ is independent of y and can be estimated from the training set and we can estimate $\mathbb{P}(y)$ by $\hat{P}(Y = y) = \frac{\sum_{i=1}^{n} I(y_i=y)}{n}$.Now suppose the features are categorical and the j^{th} coordinate of x_i is $x_{i,j}$.Also for a given vector x let $x(j)$ be its j^{th} component.So

$$\hat{P}(x(j) = a_j|y = c) = \frac{\sum_{i=1}^{n} I(x_{i,j} = a_j)I(y_i = c)}{\sum_{i=1}^{n} I(y_i = c)}$$

.Sometimes a smoothing term like Laplace smoothing is added to avoid zero denominator scenario.This gives us the prediction

$$P(y = c|x) \propto \hat{P}(y = c) \prod_{j=1}^{d} \hat{P}(x_{i,j} = a_j|y = c)$$

If the features are continuous the model assumes $P(x(j)|y = c) = \mathcal{N}(\mu_{j,c}, \sigma^2_{j,c})$.The parameters of the normal distribution is estimated as[13]

$$\mu_{j,c} = \frac{\sum_{i=1}^{n} I(y_i = c)x_{i,j}}{\sum_{i=1}^{n} I(y_i = c)}$$

$$\sigma^2_{j,c} = \frac{\sum_{i=1}^{n} I(y_i = c)(x_{i,j} - \mu_{j,c})^2}{\sum_{i=1}^{n} I(y_i = c)}$$

2.1.3 Logistic Regression

Let S be the sample and $x_1, x_2, \ldots, x_n$ be the d-dimensional feature vectors in the sample whose values are known for each card holder. Let $\mathcal{C}_0$ and $\mathcal{C}_1$ be the set of non-defaulters and defaulters in the sample respectively. Whether each of the card holders in our sample is defaulters ($Y = 1$) or non-defaulters ($Y = 0$) is known. Based on this, we wish to model p,

the risk probability for any cardholder, as a function of its various feature values. Logistic Regression is used to find the best fit for p of the form given by the following equation:

$$p_\theta(x) = \frac{1}{1 + e^{-\theta^t x}} \tag{1}$$

In (1), θ is a vector of coefficients to be determined through regression.Sometimes a constant b called bias is also added to p.Let y_i be the output variable corresponding to each x_i in the sample, taking values 0 or 1.We define

$$\mathbb{P}(y_i = 1|x_i; \theta) = p_\theta(x)$$

$$\mathbb{P}(y_i = 0|x_i; \theta) = 1 - p_\theta(x)$$

So

$$\mathbb{P}(y|x; \theta) = [p_\theta(x)]^y \, [1 - p_\theta(x)]^{1-y}$$

The regression is carried out to maximize the Likelihood function of the parameter θ given by

$$L(\theta) = \prod_{i=1}^{n} \mathbb{P}(y_i|x_i; \theta)$$

Intuitively, the likelihood function measures the probability of occurrence of our sample values in the dataset given the parameters θ, which makes it clear why maximizing it should give the best fit for our model. We start with random values for θ and update it by the following rule repeatedly for a maximum number of times until the desired tolerance is reached.

$$\theta_{new} = \theta_{old} + \alpha(y - p_\theta(x))x$$

where α is the step size.Smaller value of α ensures higher accuracy The resulting θ value is then plugged into (1), and the best fit is obtained. Though it doesn't deal with the non-linear and interactive effects of the explanatory variables, the major advantage of logistic regression is that it gives a simple probabilistic formula for classification.

2.1.4 Linear Discriminant Analysis

Linear Discriminant Ananlysis is a classification algorithm first proposed by Fisher in 1936.It can be used for problems involving both binary and multiple classes.But in our case we shall only require the binary one as our outpt variable is binary(default or not-default)

The main idea behind this algorithm is to project the data points into a line such that it maximizes the distance between the means of the two classes whille taking into account the variance within each class.Let $x_1, x_2, \ldots x_n$ be d-dimensional feature vectors which belong to two classes $\mathcal{C}_0$ and $\mathcal{C}_1$ with each class containing n_0 and n_1 data points ($n = n_0 + n_1$).Let w be the vector onto which the data is to be projected.Let m_i and M_i (i=0,1) be the means of the two data before and after projection.The data points after projection are $w^t x_i$ $(i = 1, 2, \ldots, n)$.So $M_i = w^t m_i \;\; (i = 0, 1)$.Further let

$$S_b = (m_0 - m_1)(m_0 - m_1)^t$$

$$S_0 = \sum_{x_i \in \mathcal{C}_0} (x_i - m_0)(x_i - m_0)^t$$

$$S_1 = \sum_{x_i \in \mathcal{C}_1} (x_i - m_1)(x_i - m_1)^t$$

$$S_w = S_0 + S_1$$

$$J(w) = \frac{(M_0 - M_1)^2}{\sum_{x_i \in \mathcal{C}_0}(w^t x_i - M_0)^2 + \sum_{x_i \in \mathcal{C}_1}(w^t x_i - M_1)^2} = \frac{w^t S_b w}{w^t S_w w}$$

The optimal vector w must maximize $J(w)$.So differentiating $J(w)$ with respect to w we get the equation $S_b w = \lambda S_w w$.This can be solved using different techniques like LU decomposition.

Further if S_w is invertible we get the following eigenvalue-eigenvector problem $S_w^{-1} S_b w = \lambda w$ where λ is some eigenvalue.

2.1.5 Support Vector Machine

Support Vector Machine is another classification algorithm which uses a hyperplane to separate the two classes in a classification problem.So for a point x we want to find two vectors w, b such that

$$y = w^t x + b > 0 \implies y \in \mathcal{C}_0 \text{ and}$$

$$y = w^t x + b < 0 \implies y \in \mathcal{C}_1$$

Let $x_1, x_2, \ldots, x_n$ be vectors in the feature space which are taken as training data.So the separating hyperplane is given by $\mathcal{H} := w^t x + b = 0$.Now let $\mathcal{H}_0 := w^t x + b = -\delta$ be the hyperplane parallel to $\mathcal{H}$ such that no point is there between the two planes and $\mathcal{H}_0$ is the hyperplane closest to the elements of $\mathcal{C}_0$. Similarly define $\mathcal{H}_1 := w^t x + b = \delta$.We can take $\delta = 1$ as we can always scale w to $\frac{w}{\delta}$.Let $d :=$ distance between $\mathcal{H}_0$ and $\mathcal{H}_1$.

Let $y_i = \begin{cases} 1 & , \text{if } x_i \in \mathcal{C}_1 \\ -1 & , \text{if } x_i \in \mathcal{C}_0 \end{cases}$

We want to maximize d.Let x_+ be a point in $\mathcal{H}_1$ and x_- be a point in $\mathcal{H}_0$.

So $d = |\frac{(x_+ - x_-)w^t}{|w|}| = \frac{1+b-(b-1)}{|w|} = \frac{2}{|w|}$.In order to maximize d we need to minimize w subject to the constraint $y_i(w^t x_i + b) \geq 1$ which can easily be done using numerical methods

Sometimes it is not possible to separate the two classes using a hyperplane directly.Then a transformation is applied to the space in order to make the data points separable.It appears during solving the optimization problem that the transformation itself is not required if we know how to compute the inner product in the transformed space . So a kernal function $\mathcal{K}$ is a function corresponding to a transformation ϕ such that $K(x, y) = \langle \phi(x), \phi(y) \rangle$.Some popular kernal functions are polynomial kernal,gaussian kernal,etc

2.1.6 Classification Tree

Classification tree,also known as decision tree, is a model that uses a set of if-then logical (split) conditions that permit accurate prediction or classification of cases.Classification 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.

Initially, all objects are considered as a single group. the group is split into two subgroups using a creteria, say high values of a feature for one group and low values for the other. The two subgroups are then split using the values of a second feature. The splitting process continues until a suitable stopping point is reached. The features that influence splitting can be ordered or unordered categories.

The procedure begins with a training set of records that have already been pre-classified. Building a tree that differentiates between the classes is the aim. Let the classes be $\mathcal{C}_0$ and $\mathcal{C}_1$.For simplicity we can assume the splits to be binary.Any multi-way partitioning can be accomplished with repeated binary splits, and the splitting criterion is easily generalizable to many classes. The algorithm takes into account each input field in turn to choose the optimum splitter at a node. The optimal split is the one that results in the greatest reduction in the variety of the categorization label within each partition after all potential splits have been explored and taken into account.This process is continued in the next node and a full tree is obtained.To determine the best splitting criteria different criterions like Gini Impurity,Sum of Squares Error,Information Gain,etc are used.

Let p_i be the probability of Class i.The Gini Index of a node is defined as $p_0(1-p_0)+p_1(1-p_1)$.Smaller value of the Gini Index of a node corresponds to increased purity of the node.

Sometimes to avoid overfitting of data pruning is done by removing leaves and branches so that the decision tree performs well when moved from the training data to real world predictions

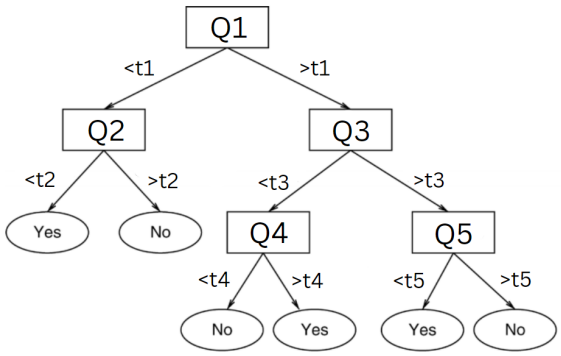


Figure 1: Model of a Classification Tree

2.1.7 Random Forest

A random forest consists of multiple random decision trees .Two types of randomnesses are built into the trees.First, each tree is built on a random sample from the original data. Second, at each tree, a subset of features are randomly selected to generate the best split.

Suppose $x_1, x_2, \ldots, x_n$ be d-dimensional feature vectors in the training set S.Now a new sample S_1 from S is created by sampling n vectors randomly and independently with re-

placement.From the d features d_1 features are selected independently at random and a classification tree is constructed with S_1 as sample and with the d_1 features.This process is then repeated(with the value of d_1 fixed).Now for a new data point the prediction of all the classification trees are taken into consideration and the final target variable is predicted.

One of the major factors in this process is the value of d_1 taken and it is necessary to take optimal value of d_1 to get better prediction.For that consider the samples $S_1, S_2, \ldots, S_n$ and the trees $T_1, T_2, \ldots, T_n$ created during the process of production of tree.Define the out of bag dataset for each of these samples as $\mathcal{O}_i = S \setminus S_i$.Now consider $x \in \mathcal{O}_j$ for some j.Let $1 \leq i_1 < i_2 < \cdots < i_k \leq n$ be such that $x \in \mathcal{O}_j$ for $j = i_1, i_2, \ldots, i_k$ but $x \notin \left(\bigcap_{j=i_1}^{i_k} \mathcal{O}_j\right)^c$.$x$ is then is sent through each of the trees $T_{i_1}, T_{i_2}, \ldots, T_{i_k}$ and the error rate is calculated.It is standard to take the initial value of d_1 to be $\lfloor\sqrt{d}\rfloor$ and then the value is increased/decreased to find the optimal value.

In a similar fashion the number of trees created in the random forest is determined.Also the trees created can be both fully grown or pruned.

2.1.8 Multi-Layer Perceptron

An MLP classifier is a feedforward neural network consisting of an input layer of nodes which store the feature values of the target point, several hidden layers and an output layer of neurons. The value stored in each node is called its activation. Each neuron in one layer contributes with a certain weight (obtained from training the model) to the activation of neurons in the subsequent layer. More precisely, the activation of a neuron in the following layer is computed by evaluating the activation function ϕ at a weighted mean of activation of neurons in the current layer. Mathematically, if y_i's are the activations of the neurons in the current layer and w_{ij}'s are their connection weights with the neuron q_j in the next layer, then

$$A(q_j) = \phi\left(\sum w_{ij} y_i\right)$$

This is repeated until all the neurons in the output layer are activated. Finally, the activation of neurons corresponding to different classes in the output layer gives the class of the target point.

The classifier model is trained through gradient descent to determine the optimal weights in the network that minimize the error

$$E = \sum_{\text{output node k}} e_k^2$$

where e_k is the error in k^{th} output node. After each piece of training data is processed, connection weights are changed based on the amount of error compared to the ideal result. Under the same setup as before, the change in w_{ij} is given by

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial v_j} y_i$$

, where v_j is the weighted sum of input connections to q_j, and η is the learning rate. Using ϕ', $\frac{\partial E}{\partial v_j}$ can be computed for hidden layers recursively in terms of changes to weights in subsequent layers. Therefore, this algorithm for modifying the weights represents backward propagation of the activation function.

2.2 Synthetic Minority Oversampling Technique (SMOTE)

In our dataset, 77.88% of cardholders are risk-free. This gives rise to a high class imbalance. Since we aim to achieve the best results in predicting the default risk of a given cardholder, i.e. the probability of the cardholder being in the minority class, such a huge bias towards non-risky cardholders can considerably degrade the performance of any classifier model. Furthermore, in situations of high class imbalance, the accuracy of the model no longer remains a reliable measure of its performance: if most of the cardholders in the population are generally risk-free (say 80%), even if the model predicts every cardholder to be risk-free, it will be right roughly 80% of the time.

SMOTE[5] is an oversampling technique used to overcome such problems in binary classifier models. Set M to be the set of data points in the minority class. Suppose a total of N synthetic data points (i.e. the oversampling observations) are required per data point in M to reduce the class imbalance to the desired extent. Generally, N is taken to be roughly the difference between the number of data points in the two classes divided by the number of data points in M so that the resulting binary class distribution is 1: 1.

1. For each x in M, K of its nearest neighbours in M are identified by using some distance metric defined on the feature space (by default, $K = 5$).

2. N of these neighbours of x are randomly selected. Call them $x_1, x_2, \ldots, x_N$.

3. For each x, a random number r between 0 and 1 is generated. Then $y_i := x + r(x_i – x)$ is added to M, and these steps are repeated for each x in M. In the end, N times the number of points originally in M is added to M as synthetic data points.

The basic intuition is that we are moving points in M towards each other to generate more points inside M. As opposed to other under and oversampling techniques in which a subset of the majority class is chosen or points already inside M are added to M to reduce class imbalance, in SMOTE, no valuable information from the data is lost and no duplicate points are generated.

3 Description of the Data

The data in this study has been taken from an important Taiwanese bank that issues cash and credit cards. Among the 30,000 data points 6,636 were defaulters(22.12%). The study used a binary variable -default payment, which takes values as 0 or 1, as the response variable and used the following 23 variables as feature variables:

- **X1**: Amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her family (supplementary) credit.
- **X2**: Gender (1 = male; 2 = female).
- **X3**: Education (1 = graduate school; 2 = university; 3 = high school; 4 = others).
- **X4**: Marital status (1 = married; 2 = single; 3 = others).

- **X5**: Age (year).
- **X6** - **X11**: History of past payment. We tracked the past monthly payment records (from April to September, 2005) as follows: **X6** = the repayment status in September, 2005; **X7** = the repayment status in August, 2005; . . . ;**X11** = the repayment status in April, 2005. The measurement scale for the repayment status is: -1 = pay duly; 1 = payment delay for one month; 2 = payment delay for two months; . . .; 8 = payment delay for eight months; 9 = payment delay for nine months and above.
- **X12**-**X17**: Amount of bill statement (NT dollar). **X12** = amount of bill statement in September, 2005; **X13** = amount of bill statement in August, 2005;. . . ; **X17** = amount of bill statement in April, 2005.
- **X18**-**X23**: Amount of previous payment (NT dollar). **X18** = amount paid in September, 2005; **X19** = amount paid in August, 2005;. . . ;**X23** = amount paid in April, 2005.

4 Methodology

We first divided the data set int two groups,one for training and the other for testing with train to test ratio 8:2.Although it is a common practice to take the error rate as a measure of the classification accuracy of models ,in a dataset which is biased towards one output variable it is not the best solution as in such cases error rates becomes insensitive to classification accuracy of the models. In such cases the area ratio under the lift chart[4][7](or cumulative gains chart) can offer a better solution for comparing the performance of various models.

In the lift chart,the horizontal axis represents the number of total data points and the vertical axis represents the cumulative number of target data points.The basic idea is to rank the data points in terms of decreasing order of predicted probabilities(of the target class) given by the model and then look for the cumulative number of target class data. There are three curves : model curve,theoretically best curve and the diagonal baseline curve.The area ratio is given by

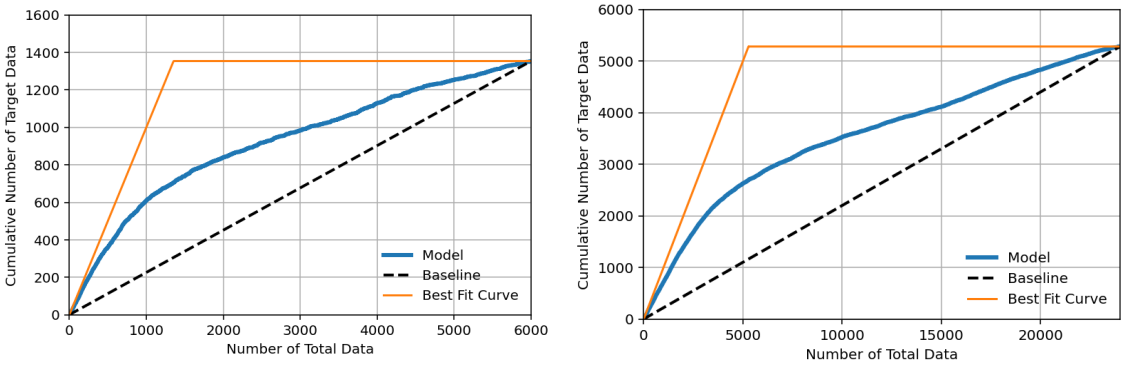
$$\text{Area ratio} = \frac{\text{area between model curve and baseline curve}}{\text{area between theoretically best curve and baseline curve}}$$

Also we implemented the technique called SMOTE which is used in data with high class imbalance to make the ratio of defaulters and non-defaulters in the train set to 1:1 and compared the results of the models both with and without SMOTE.

5 Results

The lift charts of the eight models are shown below.Of all the models Classification Tree and Random Forest outperformed most of the models both in terms of accuracy and area ratio.Naive Bayes Classifier performed the worst in terms of accuracy although it performed relatively well in terms of area ratio.As the testing data is the effective data set used to measure the classification accuracy of the models , we can conclude that Random Forest is the best model among the eight used.

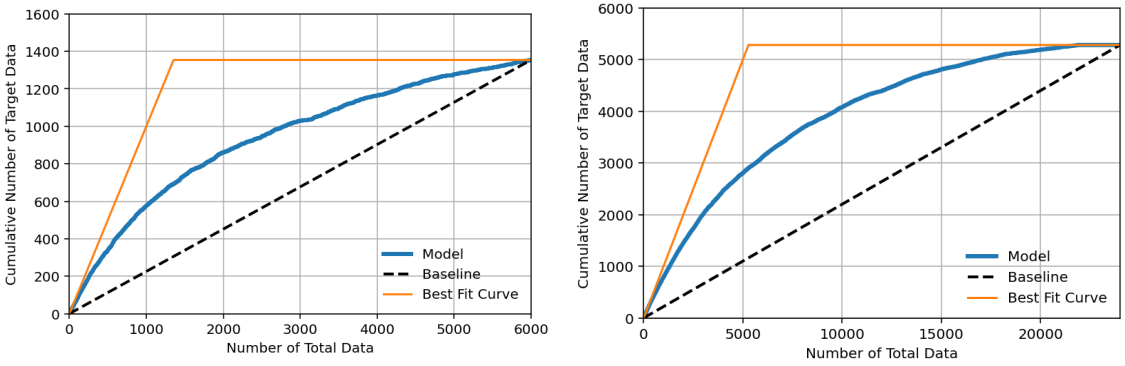
When using area ratio of testing data the eight models can be ranked as :Random Forest ,Classification Tree,K-Nearest Neighbour,Naive Bayes Classifier,Logistic Regression,Multi-Layer Perceptron,Linear Discriminant Analysis,Support Vector Machine



(a) Testing Data

(b) Training Data

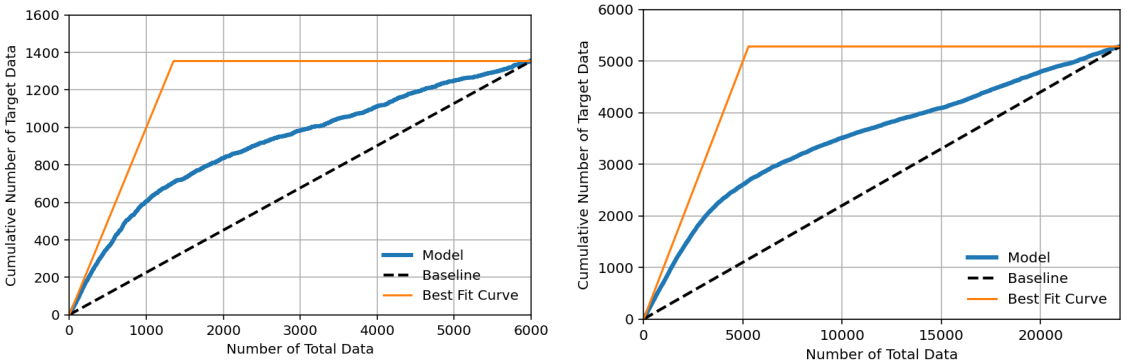
Figure 2: Lift Chart for Logistic Regression



(a) Testing Data

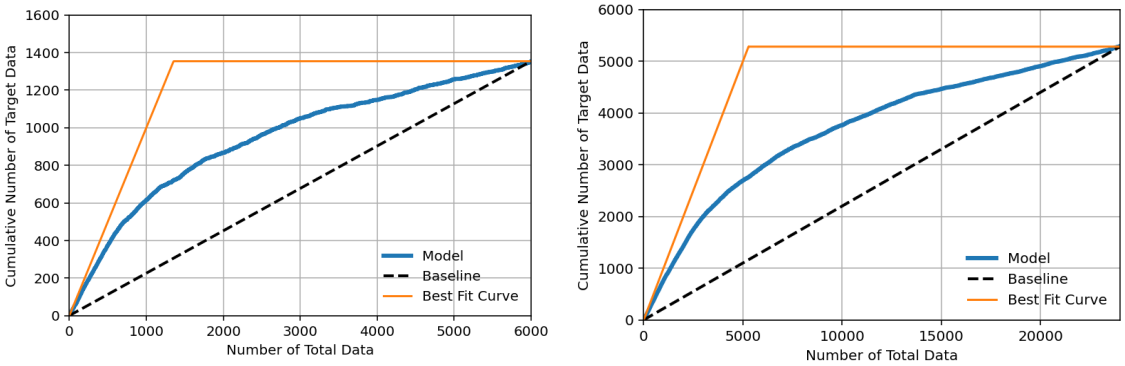
(b) Training Data

Figure 3: Lift Chart for k Nearest Neighbours



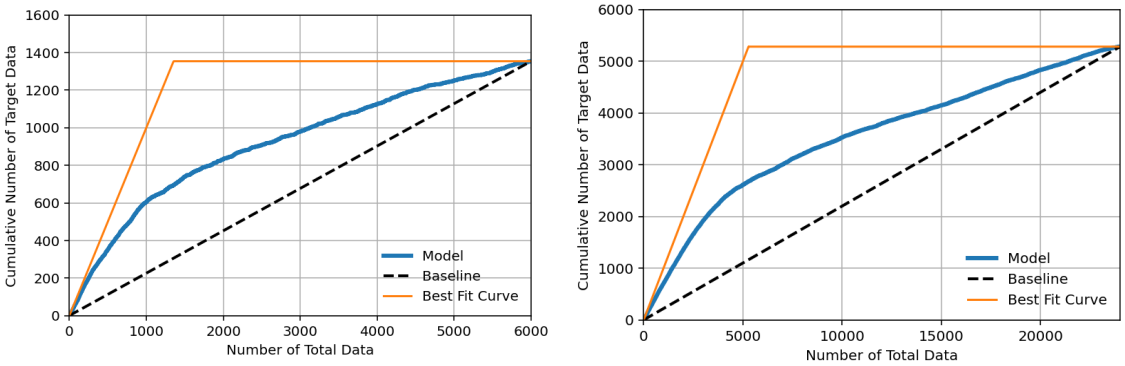
(a) Testing Data (b) Training Data

Figure 4: Lift Chart for Linear Discriminant Analysis



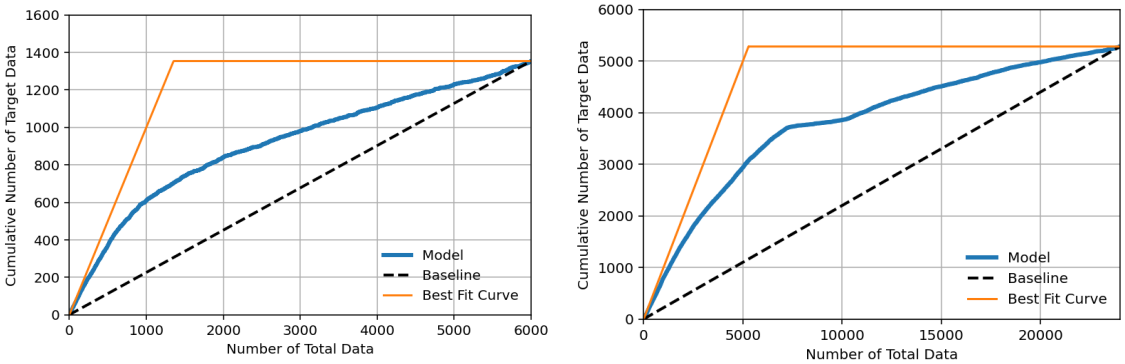
(a) Testing Data (b) Training Data

Figure 5: Lift Chart for Classification Tree



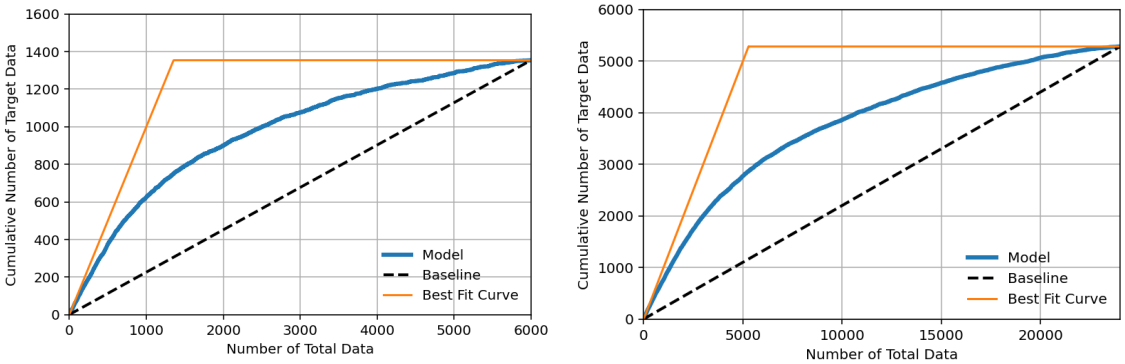
(a) Testing Data (b) Training Data

Figure 6: Lift Chart for Multi-Layer Perceptron



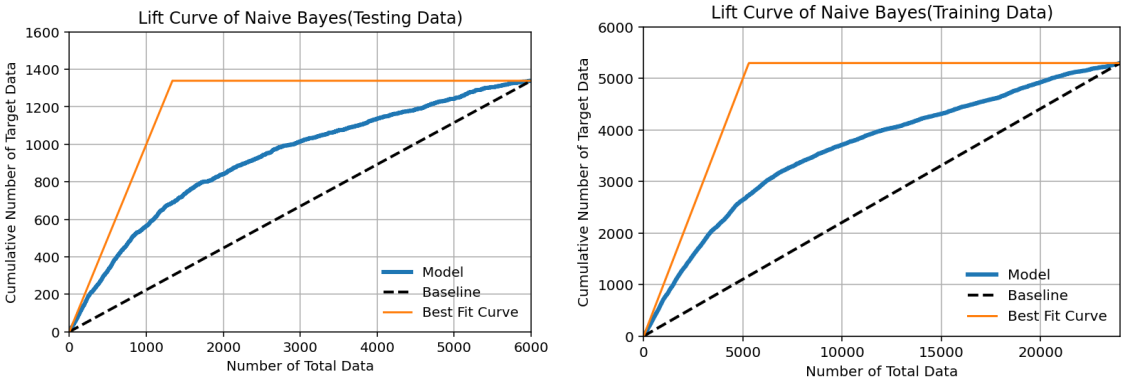
(a) Testing Data (b) Training Data

Figure 7: Lift Chart for Support Vector Machine



(a) Testing Data (b) Training Data

Figure 8: Lift Chart for Random Forest



(a) Testing Data (b) Training Data

Figure 9: Lift Chart for Naive Bayes Classifier

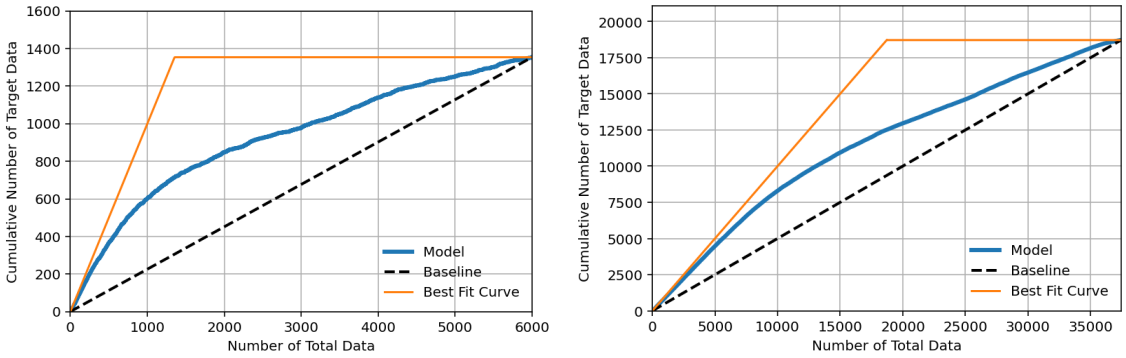
The table of prediction accuracy and area ratio of the models is given below

Model	Accuracy		Area Ratio	
	Training	Testing	Training	Testing
Logistic Regression	0.8100	0.8103	0.4418	0.4720
k-Nearest Neighbours	0.8200	0.8068	0.6235	0.5011
Linear Discriminant Analysis	0.8107	0.8105	0.4298	0.4612
Classification Tree	0.8216	0.8213	0.5173	0.5122
Multi-Layer Perceptron	0.8062	0.8068	0.4398	0.4661
Support Vector Machine	0.8258	0.8205	0.5712	0.4544
Random Forest	0.8207	0.8203	0.5642	0.5669
Naive Bayes Classifier	0.7974	0.7955	0.4821	0.4897

Table 1: Classification accuracy of the models without SMOTE

We then used SMOTE to make the ratio of defaulters and non-defaulters in the training set to 1:1.This makes error rate a reliable measure of classification accuracy of the models.Based on error rate of testing data Support Vector Machine was the best of all the models followed closely by Random Forest whereas Logistic Regression performed the worst

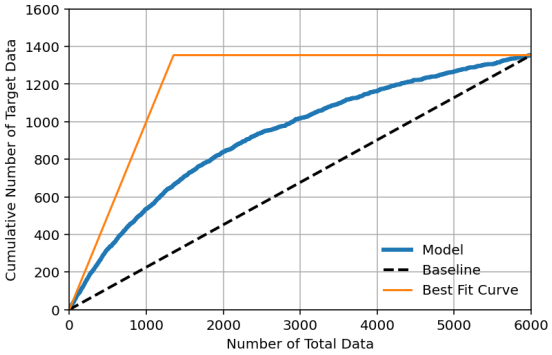
Also based on area ratio of testing data the ranking of the models are:Random Forest,Support Vector Machine,Classification Tree,Naive Bayes Classifier,Multi-Layer Perceptron,Logistic Regression,K-Nearest Neighbours,Linear Discriminant Analysis.The lift charts for the models are given below:



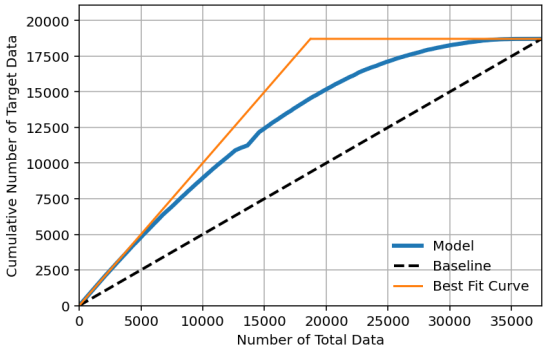
(a) Testing Data

(b) Training Data

Figure 10: Lift Chart for Logistic Regression with SMOTE

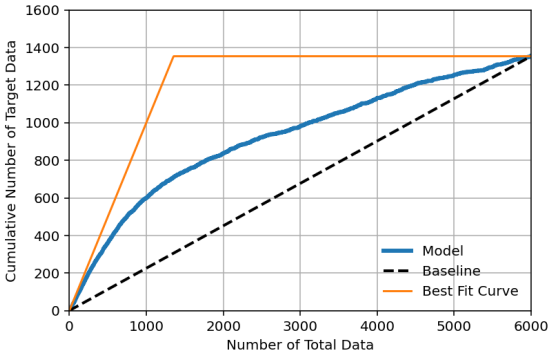


(a) Testing Data

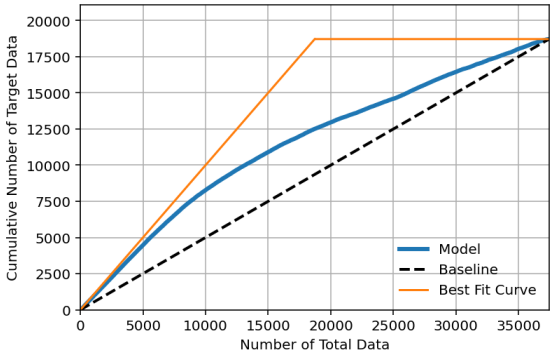


(b) Training Data

Figure 11: Lift Chart for k Nearest Neighbours with SMOTE

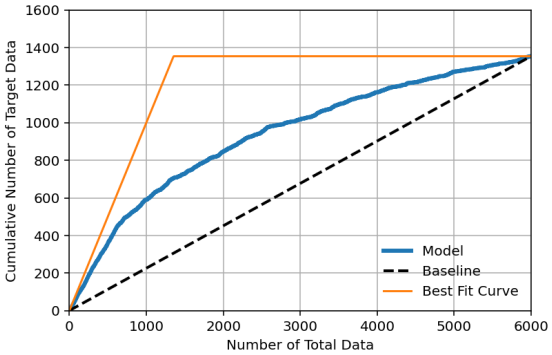


(a) Testing Data

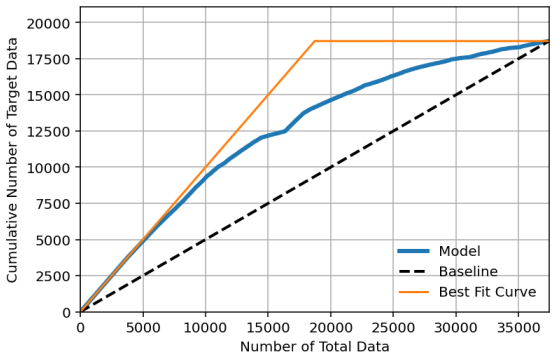


(b) Training Data

Figure 12: Lift Chart for Linear Discriminant Analysis with SMOTE

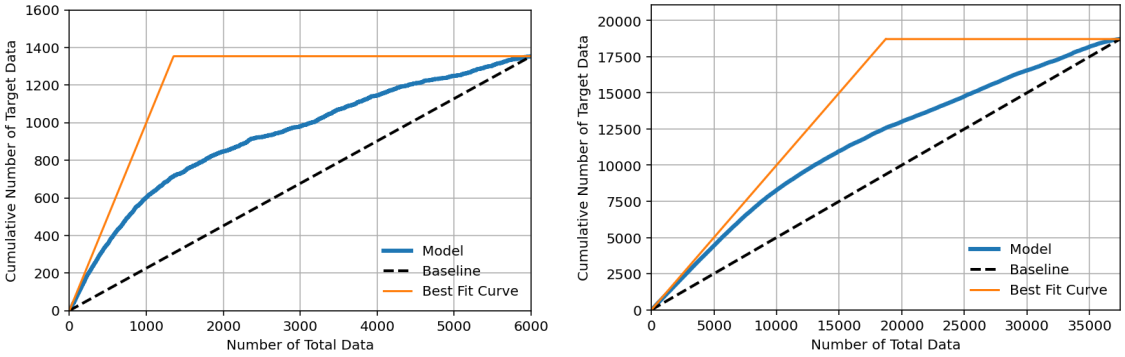


(a) Testing Data



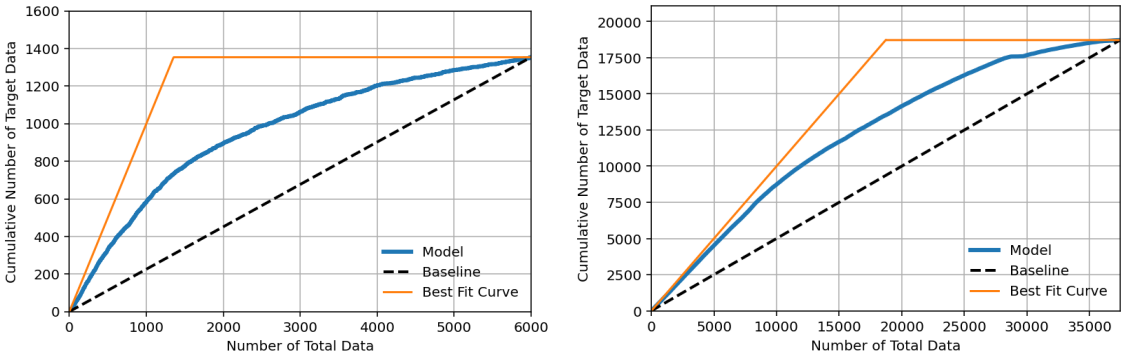
(b) Training Data

Figure 13: Lift Chart for Classification Tree with SMOTE



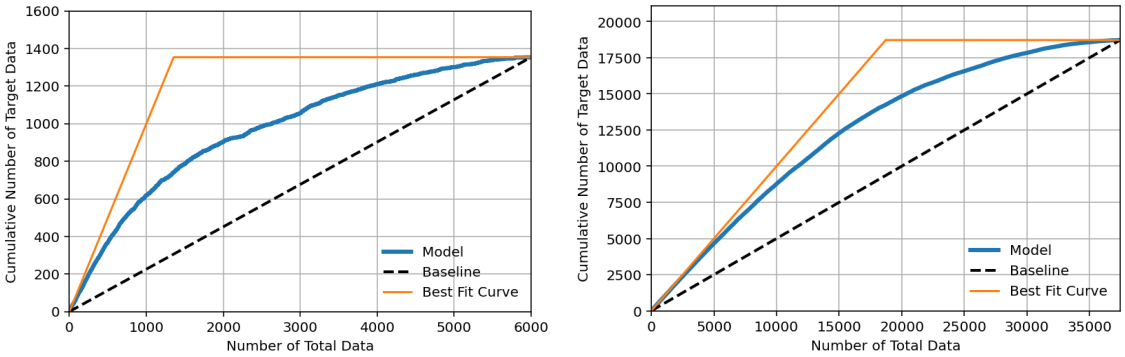
(a) Testing Data (b) Training Data

Figure 14: Lift Chart for Multi-Layer Perceptron with SMOTE



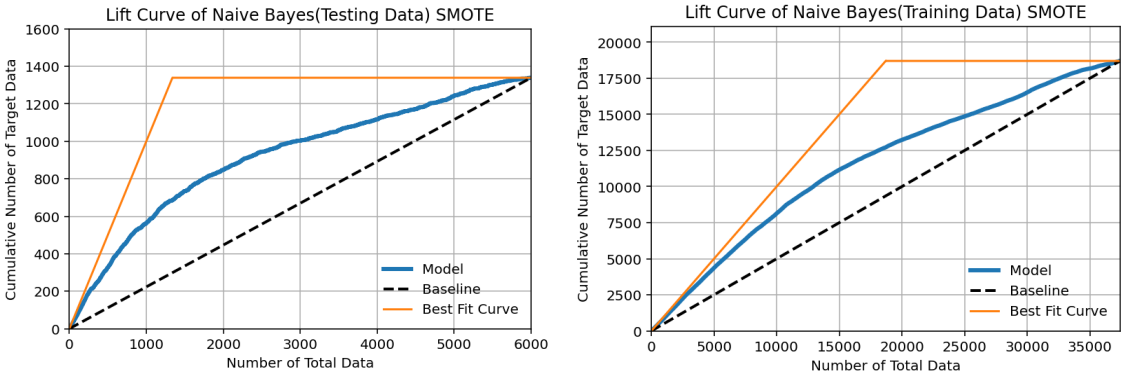
(a) Testing Data (b) Training Data

Figure 15: Lift Chart for Support Vector Machine with SMOTE



(a) Testing Data (b) Training Data

Figure 16: Lift Chart for Random Forest with SMOTE



(a) Testing Data

(b) Training Data

Figure 17: Lift Chart for Naive Bayes Classifier with SMOTE

The table of prediction accuracy and area ratio of the models with SMOTE is given below:

Model	Accuracy		Area Ratio	
	Training	Testing	Training	Testing
Logistic Regression	0.6706	0.6803	0.4569	0.4776
k-Nearest Neighbours	0.7759	0.6925	0.7286	0.4765
Linear Discriminant Analysis	0.6715	0.6880	0.4513	0.4721
Classification Tree	0.7576	0.7365	0.6615	0.4977
Multi-Layer Perceptron	0.6732	0.6870	0.4639	0.4808
Support Vector Machine	0.7243	0.7811	0.6201	0.5384
Random Forest	0.7602	0.7791	0.6719	0.5659
Naive Bayes Classifier	0.6945	0.7440	0.4735	0.4808

Table 2: Classification accuracy of the models with SMOTE

Based on this study we can conclude that Random Forest is the most reliable of all the eight models implemented for default prediction

6 Sorting Smoothing Method

The novel technique called Sorting Smooting Method(SSM) was proposed in [1] to estimate the real probability of default.The real probability was estimated using the following formula

$$P_i = \frac{\sum_{j=-n}^{n} Y_{i+j}}{2n+1} \text{ ,where}$$

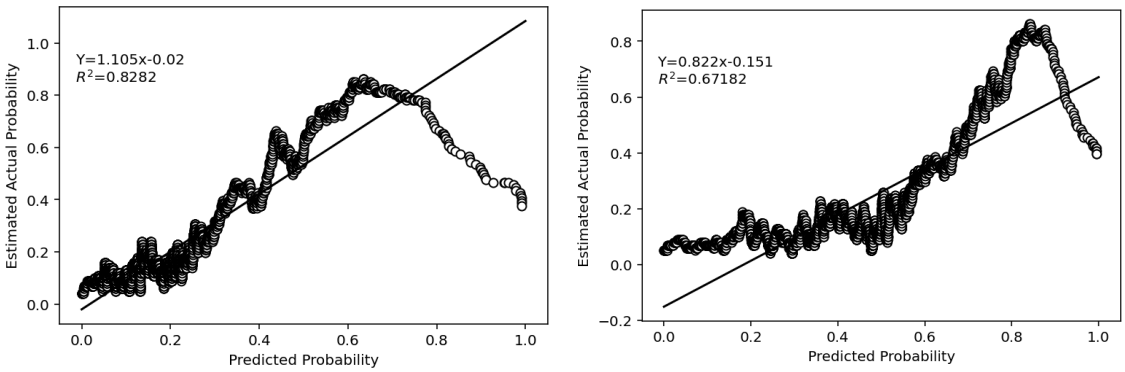
P_i = estimated real probability of default in the i-th order of validation data
Y_i = binary variable with real default risk in the i-th order of validation data
$Y_i = 1$ stands for "happened"
$Y_i = 0$ stands for "not happened"

n = numbers of data for smoothing.

n=50 was used in our case

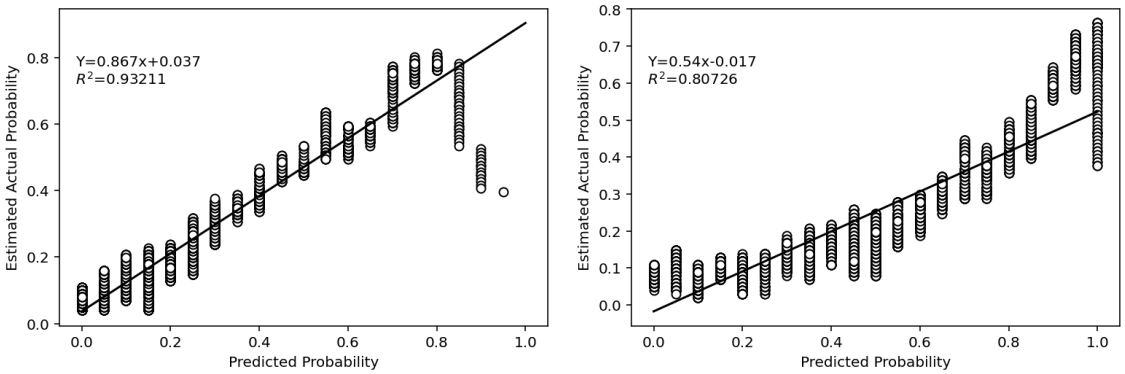
Treating these P_i's as the real probability of default we drew a sctterplot with the x-axis representing the preditive default probability and the x-axis representing the real probability of default estimated using SSM.Then we fitted a linear regression model on the data.If the coefficient of determination(R^2) is close to 1,the intercept(β_0) close to 0 and the slope (β_1) close to 1 then we can conclude that the model represents the actual probability.

We repeated the same thing after appling SMOTE to the training dataset.The graphs are given below:



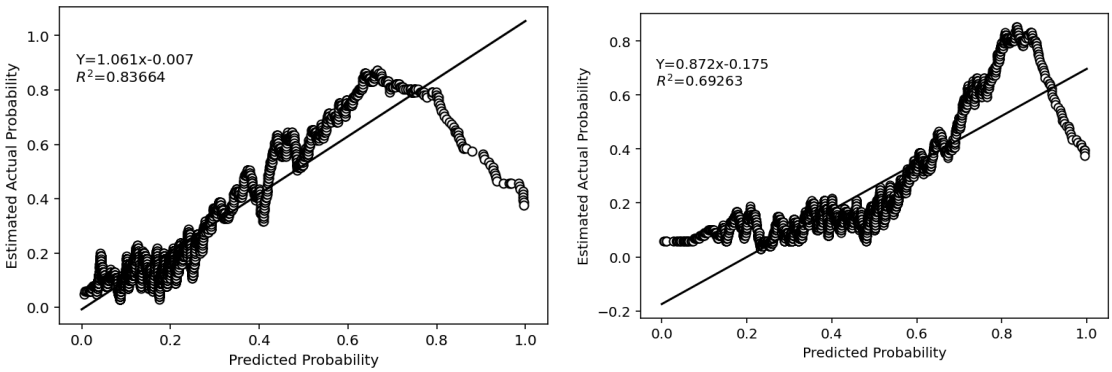
(a) without SMOTE (b) with SMOTE

Figure 18: Scatterplot diagram for Logistic Regression



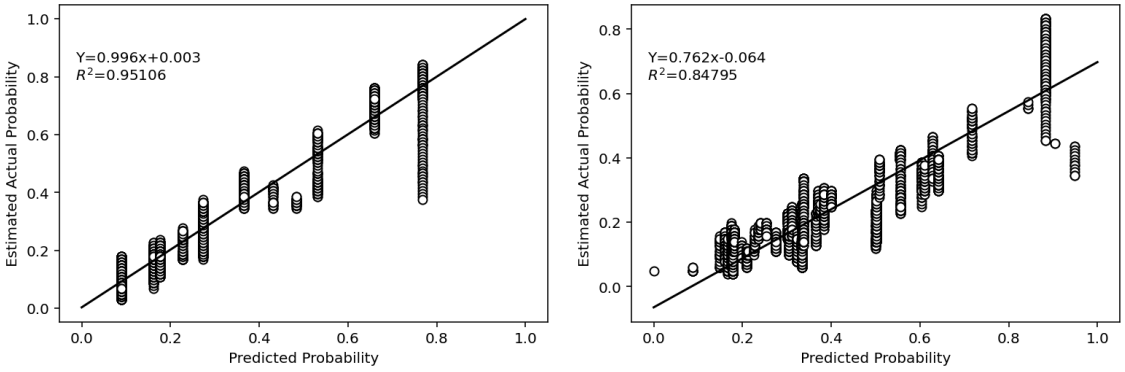
(a) without SMOTE (b) with SMOTE

Figure 19: Scatterplot diagram for k nearest neighbours



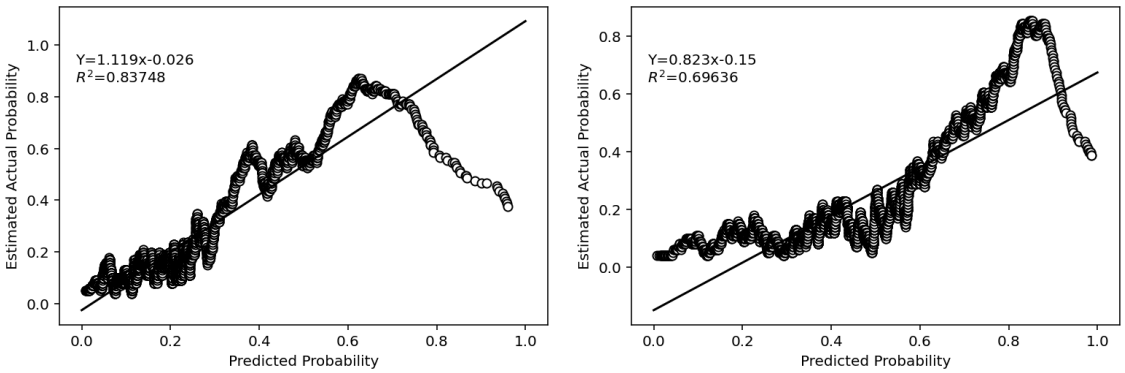
(a) without SMOTE (b) with SMOTE

Figure 20: Scatterplot diagram for Linear Discriminat Analysis



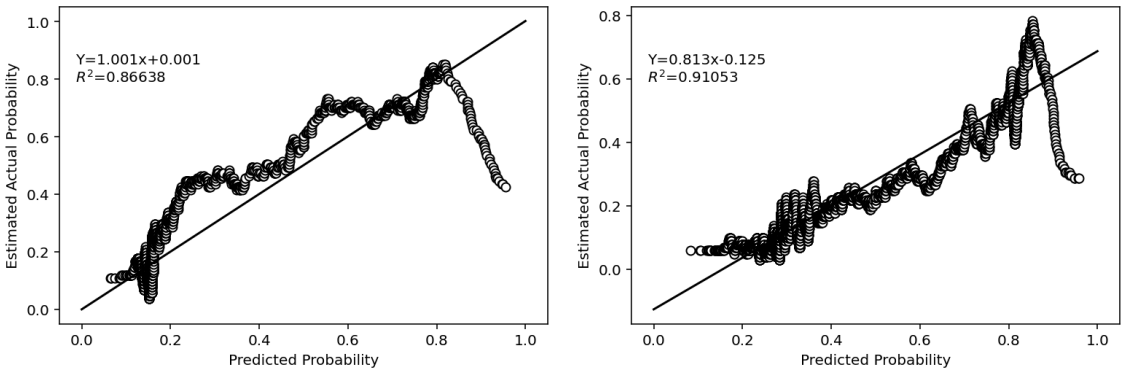
(a) without SMOTE (b) with SMOTE

Figure 21: Scatterplot diagram for Classification Tree



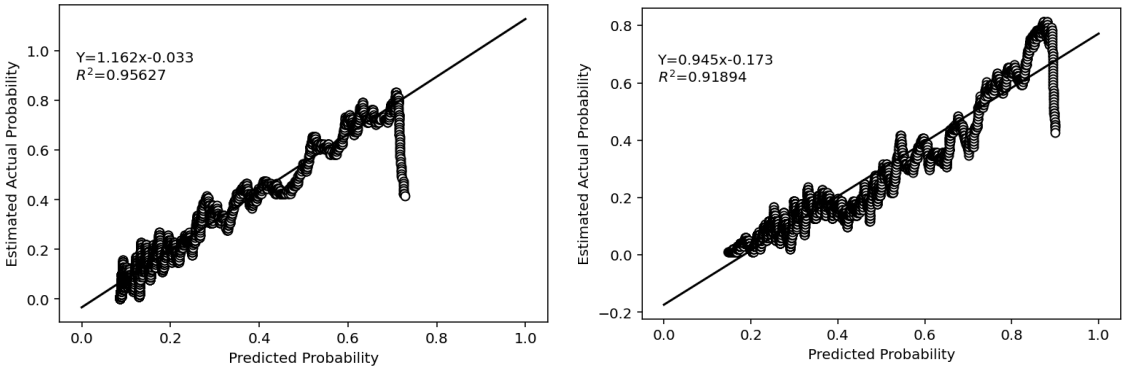
(a) without SMOTE (b) with SMOTE

Figure 22: Scatterplot diagram for Multi-Layer Perceptron



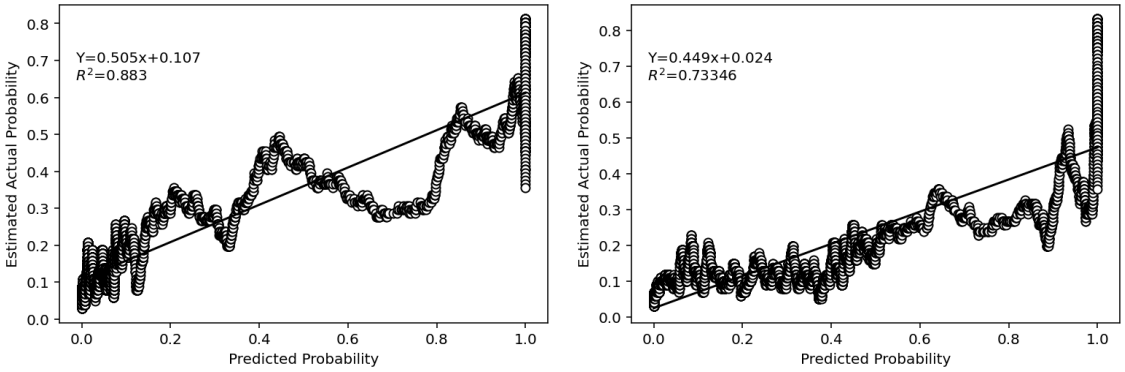
(a) without SMOTE (b) with SMOTE

Figure 23: Scatterplot diagram for Support Vector Machine



(a) without SMOTE (b) with SMOTE

Figure 24: Scatterplot diagram for Random Forest



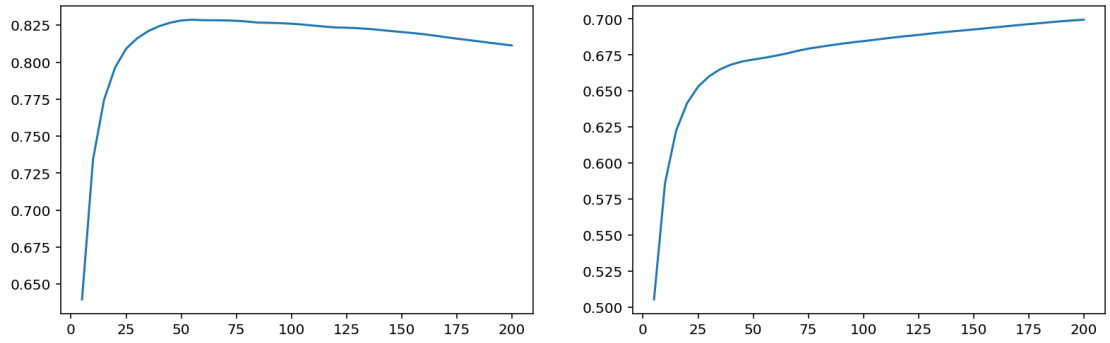
(a) without SMOTE (b) with SMOTE

Figure 25: Scatterplot diagram for Naive Bayes Classifier

From here also we can conclude that Support Vector Machine and Random Forest are two of the best classifiers for this purpose

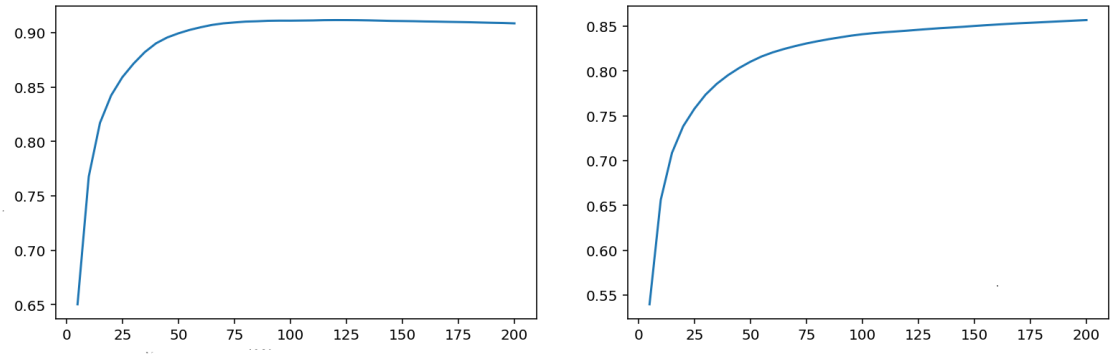
7 Choice of n in SSM

Both in [1] and our study the value of n=50 was taken while implementing SSM.So we wanted to study how robust this choice is. For that purpose we looked at the values of $n = 5, 10, 15, \ldots, 200$ and looked at the plot of n vs R^2, the coefficient of determination and obtained the following result:



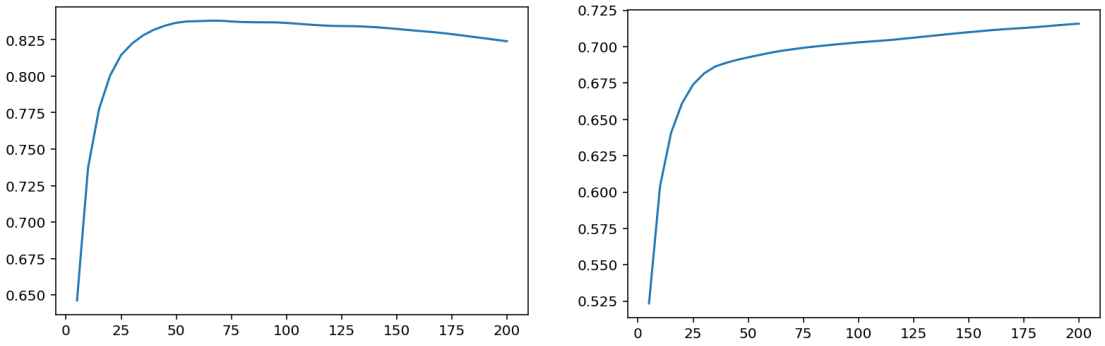
(a) without SMOTE (b) with SMOTE

Figure 26: n vs R^2 for Logistic Regression



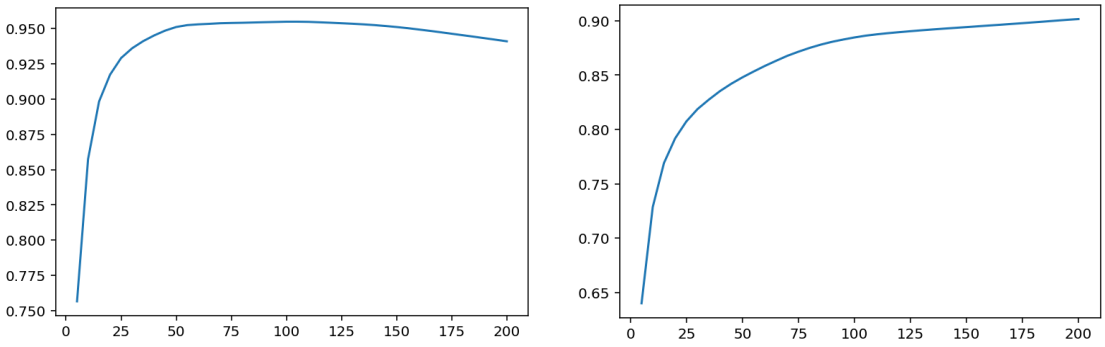
(a) without SMOTE (b) with SMOTE

Figure 27: n vs R^2 for k-th nearest neighbours



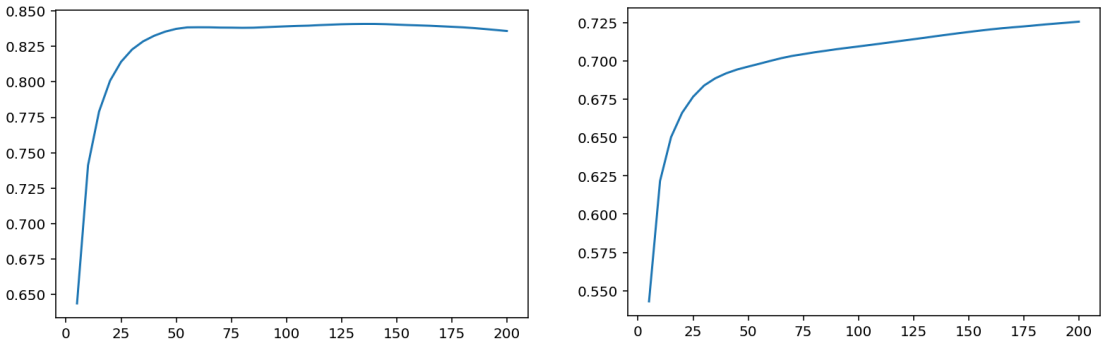
(a) without SMOTE (b) with SMOTE

Figure 28: n vs R^2 for Linear Discriminant Analysis



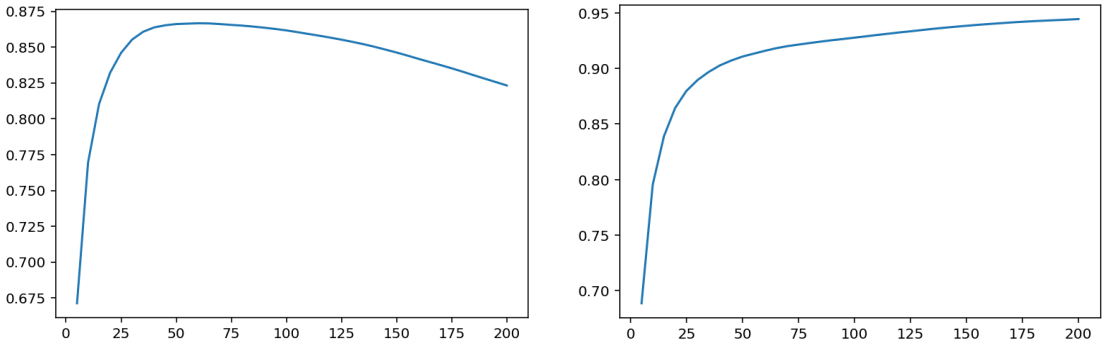
(a) without SMOTE (b) with SMOTE

Figure 29: n vs R^2 for Classification Tree



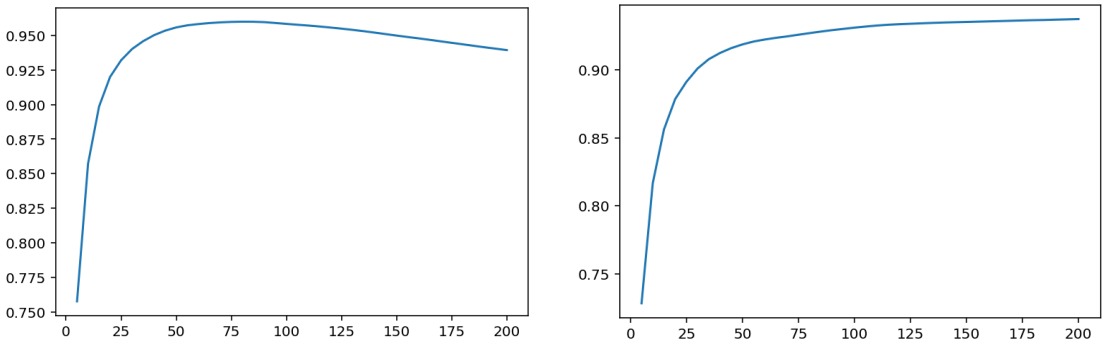
(a) without SMOTE (b) with SMOTE

Figure 30: n vs R^2 for Multi-Layer Perceptron



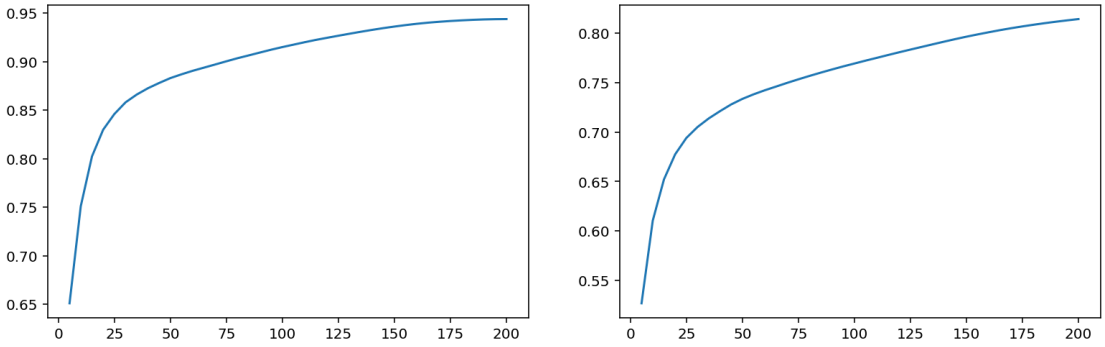
(a) without SMOTE (b) with SMOTE

Figure 31: n vs R^2 for Support Vector Machine



(a) without SMOTE (b) with SMOTE

Figure 32: n vs R^2 for Random Forest



(a) without SMOTE (b) with SMOTE

Figure 33: n vs R^2 for Naive Bayes Classifier

From the figures one can observe that at first the value of R^2 increases with the increase in value of n but after a certain point(near n=50) the value of R^2 becomes stable. This shows

50 is a good choice for n.

References

[1] I-Cheng Yeh and Che-hui Lien,*The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients*,Expert Systems with Applications(2009),Vol. 36,No. 2,2473-2480.

[2] F. Pedregosa,G. Varoquaux,A. Gramfort,V. Michel,B. Thirion,O. Grisel,M. Blondel,P. Prettenhofer,R. Weiss,V. Dubourg,J. Vanderplas,A. Passos, Cournapeau,M. Brucher,M. Perrot and E. Duchesnay,*Scikit-learn: Machine Learning in Python*,Journal of Machine Learning Research(2011),Vol. 12,2825-2830.

[3] Guillaume Lemaître,Fernando Nogueira and Christos K. Aridas,*Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning*,Journal of Machine Learning Research(2017),Vol. 18,No. 17,1-5.

[4] Michael J.A. Berry and Gordon S. Linoff,*Data Mining Techniques for Marketing, Sales, and Customer Relationship Management.*Wiley Publishing Inc.(2003).

[5] N. V. Chawla,K. W. Bowyer,L. O. Hall and W. P. Kegelmeyer, *SMOTE: Synthetic Minority Over-sampling Technique*.Journal of Artificial Intelligence Research(2002),Vol. 16, 321–357.

[6] Trevor Hastie,Robert Tibshirani and Jerome Friedman,*The Elements of Statistical Learning: Data Mining,Inference and Prediction*,Springer(2008).

[7] Chris J. Lloyd,*A Note on the Area under the Gains Chart*,International Journal of Statistics in Medical Research(2018),Vol. 7,66-69.

[8] Lior Rokach and Oded Maimon,*Data Mining with Decision Trees:Theory and Applications*,World Scientific(2014).

[9] Lei Brieman,Jerome H. Friedman,Richard A. Olsen and Charles J. Stone,*Classification and Regression Trees*,Chapman and Hall(1998).

[10] Richard O Duda,Peter E Hart and David G Stork,*Pattern Classification*,Wiley India(2006).

[11] Usama M. Fayyad and Keki B Irani,*On the handling of continuous-valued attributes in decision tree generation*,Machine Learning(1992),Vol. 8,No. 1,87-102.

[12] Peter Hall,Byeong U. Park,Richard J. Samworth,*Choice of Neighbour Order in Nearest-Neighbour Classification*,The Annals of Statistics(2008),36(5),2135-2152.

[13] Kevin P. Murphy,*Machine Learning: A Probabilistic Perspective*,The MIT Press(2012).