Data Analysis

A Project Presented to

The Faculty of the Department of Industrial and Systems Engineering

San José State University

In Partial Fulfillment of the Requirements

For the Degree Master of Science in Industrial and Systems Engineering

Shrilekha Singh

May 2019

SAN JOSÉ STATE UIVERSITY

The Undersigned Project Committee Approves the Project Titled

Data Analysis

By Shrilekha Singh

APPROVED FOR THE DEPARTMENT OF THE INDUSTRIAL & SYSTEMS ENGINEERING

Dr. Niranjani Patel, Department of Industrial and Systems Engineering

Abstract

Machine Learning has been a growing field intersecting many disciplines such as computer sciences, statistics mathematics, and Industrial engineering. As the this area throws open a ton of problems, this project aims to pick those areas where the challenge lies in dealing with high dimensional and imbalanced data-set (p (predictors) >>n (number of observations) or imbalanced labels). The present research work provides in-depth details in laying out methods that we use to select features, visualize high dimensional data, and eventually identifying appropriate models to do classification.

To provide more details on the problems, our first problem is tied to credit default data-set where we provide details on how to efficiently classify default rates, when the proportion of it is way smaller than the non-default cases. A classifier may have a high overall accuracy but it may perform poorly on default cases. In this work we explain how to overcome this problem. Second, we discuss the case of p >>n, and show how certain algorithms struggle fail on feature selection and what alternative methods are available to overcome it. Lastly, we discuss MNIST data-set which is a well researched data-set and provides a great entry point to understand machine vision problem, how it deal with high dimensions and how what techniques we can use to visualize it.

The algorithms that we relied on is as follows: For feature selection we have PCA (Principal Component Analysis), Lasso, Elasticnet, Correlation Analysis, and Boosted Trees. After performing the necessary data processing steps we used the following machine learning algorithms to perform classification: KNN (k-nearest neighbour), logistic regression, SVM (support vector machines, Bagging, Random Forest, Boosting, and CNN (convolutional neural network). Finally we have provided all the results of our analysis in a table which helps to understand the performance on each algorithms and know which one gave the most accurate results. We concluded that Logistic regression failed to perform well on imbalance data (credit default data) whereas it gives better result on MNIST data-set. Linear SVM will polynomial kernels gives better result with all three data set but linear SVM performed poorly because data might not be linearly separable. We also observed that XGboost performed well on leukemia dataset as well as did the automatic feature selection. we performed our experiment in Python (Jupyter Notebook) and R Studio.

keywords: p >>n, SVM, KNN, Random Forest, XGBoost, Feature Selection

Contents

List of Tables

1 Chapter 1: Abbreviations, Python Libraries and Terminologies used in the report

1.1 Abbreviations used throughout the project report

- ML: Machine Learning
- MNIST: Modified National Institute of Standards and Technology- It is the large database of handwritten digits commonly used for training various image processing systems
- SVM: Support Vector Machines
- KNN: K - Nearest Neighbor
- CNN: Convolutional Neural Network
- PCA: Principal Component Analysis
- t-SNE: t Stochastic Neighbour Embedding
- NYU: New York University
- ALL: Acute lymphocytic leukemia
- AML: Acute myelocytic leukemia
- p: number of predictors in the data-set
- n: number of observations in the data-set

1.2 Libraries are used in python

- Matplotlib: used for plotting all the graphs
- Pandas: used for creating tables, and analysing the data
- Numpy: used for mathematical operations such as summation, inverse of matrix, multiplication, transpose of matrix etc.
- sklearn: used for building classifiers such as logistiC regression, CNN, KNN, SVM

- sklearn.manifold: to perform t-SNE
- sklearn.decomposition: to perform PCA
- scipy.linalg: for singular value decomposition
- csv: to import the CSV file
- sklearn.metrics: to get confusion matrix and accuracy score of the classifier

1.3 Terminologies used throughout the report

- Confusion matrix: Confusion matrix is the truth table which gives the value of false positive, false negative and correctly classified labels.
- features: It is individual measurable property or characteristic of the dataset. In literature, it is also known as predictors, variables.
- Regularization: This is a form of regression, that constrains/ regularizes or shrinks the coefficient estimates towards zero

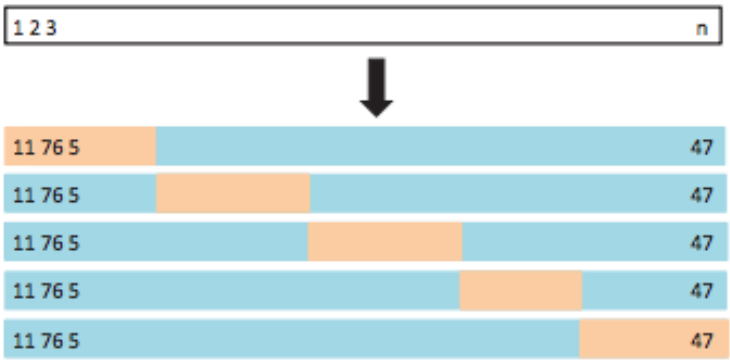

Figure 1: In this figure, training set consist of n observations. It is divided into k =5 folds and fold shown in beige is validation set and those shown in blue color is training set. This will be repeated 5 times. The test error is estimated by averaging the five resulting MSE (mean square error) estimates.

- Dimension Reduction: This involves projecting the p predictors into K-dimensional subspace, where K <p.
- Train set: We build our model using data from training set

- Validation set: We check the accuracy of our model using validation set.It allows us to improve our model if validation error is too high.
- Test set: We implement our trained model to predict or classify using test set
- K-fold cross validation: It involves randomly dividing the set of observations into k groups, or folds (shown in figure 1), of approximately equal size. The first fold is treated as a validation set, and the method is fit on the remaining k - 1 folds (training set).

2 Chapter 2: Introduction and Motivation

Machine Learning has been a growing field intersecting many disciplines such as computer sciences, statistics mathematics, and Industrial engineering. The interest in these type of problems has been increased over the years because of its potential application. It's ability to learn directly from data made it a powerful tool in almost all field. There are significant advances in machine learning, as a result system can now outperforms humans at specific task. There are several examples of machine learning that we use every day and perhaps no idea that they are driven by machine learning model. Some examples are: virtual personal assistant, traffic predictions, e-mail spam, social media services, search engine result refining, product recommendation, weather forecasting etcetera. This report discusses about various classification machine learning algorithm, feature selection method and dimensional reduction technique. The algorithms discussed in this report are: Logistic Regression, K-nearest Neighbor, Support Vector Machine, Random Forest, Convolutional Neural Network.

Our aim in this report is to build classifiers to classify test labels correctly and with high accuracy. We carefully chose three different type of data set so that we explore various techniques to improve the accuracy rate of the implemented classifier. Our first data set Credit Card default is about binary class classification problem and it has few number of predictors compare to other two data-set. This is dataset is imbalanced which means number of data points available for different the classes is different. For our second data-set- Leukemia, there 7128 features (variables) and 72 number of observations. It is almost tedious job to analyze each of the feature individually. Our aim was to explore various feature selection method such as Elasticnet, XGBoost. Our third data set is MNIST. Since the MNIST dataset is quite popular in the research community. We were motivated to know what are the proven ways which we can use to handle this data, apply algorithms and infer the results. At present, there is a wide need for solving problems that involve high dimensional data. The chosen problem is on similar lines of high-dimensional data and opens the opportunity to learn about dimensionality reduction and how to create a visualization to get a sense of the data. This problem allowed us to apply ML algorithm. As the problem is a highly researched

one, it allowed us to know if we have implemented our solution in the right way or not. The right execution solidifies our learning and motivates us to go further in the field of Computer Vision.

3 Chapter 3: Data set Introduction

Following are the detailed description of three data-sets covered in this report:

3.1 Default of Credit Card Clients Data set

Default of Credit Card Clients Data set: This dataset is taken from the UCI repository. This dataset contains information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005. It has 25 variables:

1. ID: ID of each client
2. LIMIT_BAL: Amount of given credit
3. SEX: Gender (1=male, 2=female)
4. EDUCATION: (1=graduate school, 2=university, 3=high school, 4=others)
5. MARRIAGE: Marital status (1=married, 2=single, 3=others)
6. AGE: Age in years
7. PAY_0: Repayment status in September, 2005 (-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). Similarly for PAY_2,PAY_3, PAY_4, PAY_5, PAY_6
8. BILL_AMT1: Amount of bill statement in September, 2005 (NT dollar)
9. BILL_AMT1: Amount of bill statement in September, 2005 (NT dollar)
10. BILL_AMT2: Amount of bill statement in August, 2005 (NT dollar)
11. BILL_AMT3: Amount of bill statement in July, 2005 (NT dollar)
12. BILL_AMT4: Amount of bill statement in June, 2005 (NT dollar)
13. BILL_AMT5: Amount of bill statement in May, 2005 (NT dollar)
14. BILL_AMT6: Amount of bill statement in April, 2005 (NT dollar)

15. PAY_AMT1: Amount of previous payment in September, 2005 (NT dollar)
16. PAY_AMT2: Amount of previous payment in August, 2005 (NT dollar)
17. PAY_AMT3: Amount of previous payment in July, 2005 (NT dollar)
18. PAY_AMT4: Amount of previous payment in June, 2005 (NT dollar)
19. PAY_AMT5: Amount of previous payment in May, 2005 (NT dollar)
20. PAY_AMT6: Amount of previous payment in April, 2005 (NT dollar)
21. default.payment.next.month: Default payment (1=yes, 0=no)

    Variable 1- 20 are predictors and 21 is the response variable.

3.2 MNIST data set

MNIST: The MNIST database of handwritten digits, formed by Yann LeCun of NYU, has a total of 70,000 examples from approximately 250 writers. The images are 28*28 in size. In this report, we flattened the 28*28 matrix into a vector of 784 features. There are 60,000 images in the training set while 10,000 images in the test set. The digits in both the dataset are labeled as any number between 0 to 9. All pixels is considered as features and digits 0-9 as response (labels) variables.

3.3 leukemia

These data arise from the landmark Golub et al. (1999) science paper and taken from the Stanford website. It is about gene expression measurements on 72 leukemia patients, 47 “ALL (Acute lymphocytic leukemia) and 25 AML (Acute myelocytic leukemia). It consists of 7128 * 72 matrix. In this dataset, all the gene expressions are features and ”ALL” and ”AML” as response variables. Our aim is classify gene expressions as AML or ALL.

4 Chapter 4: Data Preprocessing and Data Viszualization

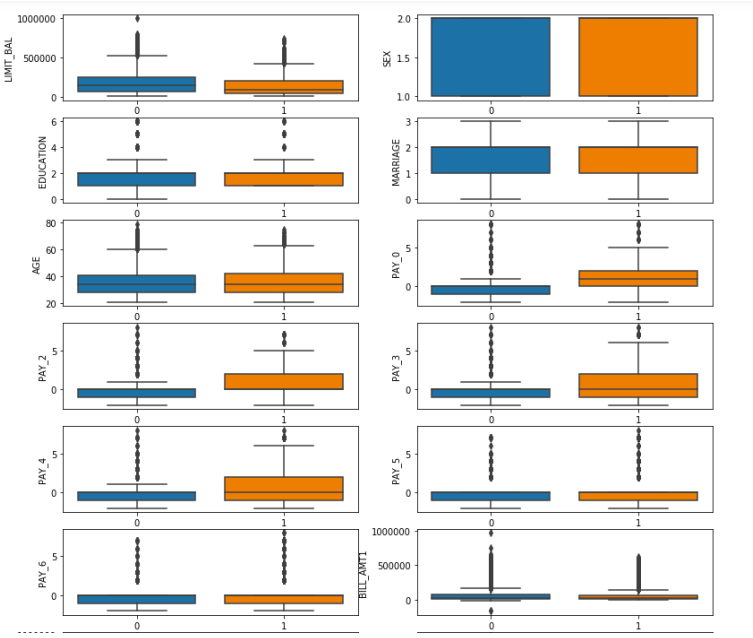

Figure 2: Box plot to showing various predictors of default credit data set against response variable

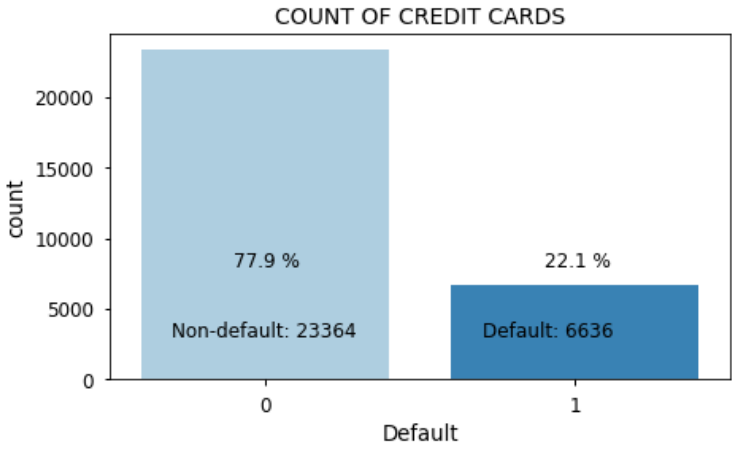

Figure 3: Histogram comparing count of defaulted and non-defaulted in the dataset

To visualize the Default credit data set, Box plot (figure 2) has been used. The plot compares each predictor with the response variable. The plot shows that predictor variables such as age, sex, education, marriage does not depend on the response variable as they are of same height. The histogram (figure 3) shows that 77.9 % are non-defaulted and 22.1 % are defaulted. Also, in figure 3 color-bar on the right hand side of the

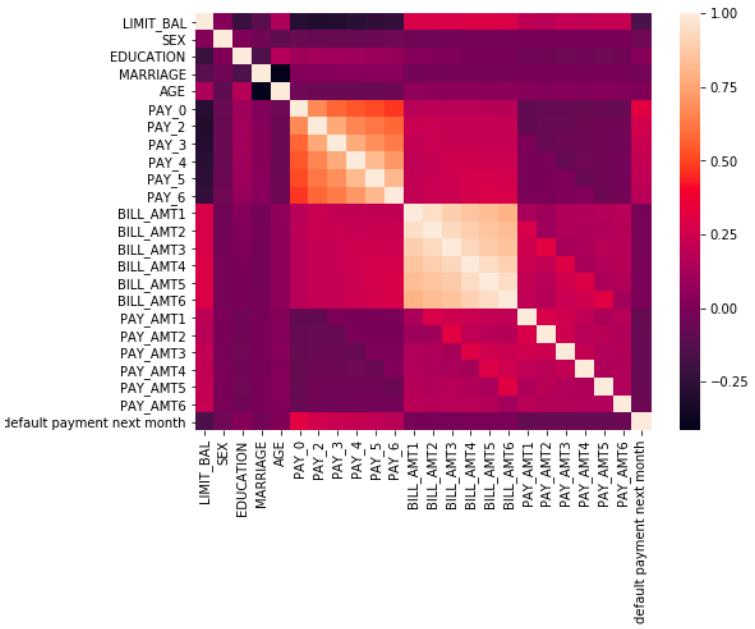

Figure 4: Correlation matrix for Credit default data set

figure shows the correlation among the variables. The lightest color shows that there is the highest correlation among the variables. It shows that response variable default payment is majorly co-related to PAY_0, PAY_1, PAY_2, PAY_3, PAY_4,PAY_5 and the remaining predictors does not have strong correlation with the response variable. We have used seaborn in python to get the correlation map also known as heat map.

4.1 Feature Selection

Selecting important features in the data-set is important. It makes the interpretation much easier and reduces the complexity of the model. It also saves computational time while training the model. There are several techniques available for feature selection. Some of the techniques are looking for multicollinearity among the predictors and dropping anyone variable which are highly collinear or looking at the correlation between response variable and predictors. As discussed above in credit card data-set, we dropped those predictors which were not strongly co-related with the response variables.

Regularization mainly Lasso and ElasticNet are another important technique which can be used for variable selection. We can use Lasso regression but it does not do well with p >>n dataset. It can atmost select n (number of observations) variables before

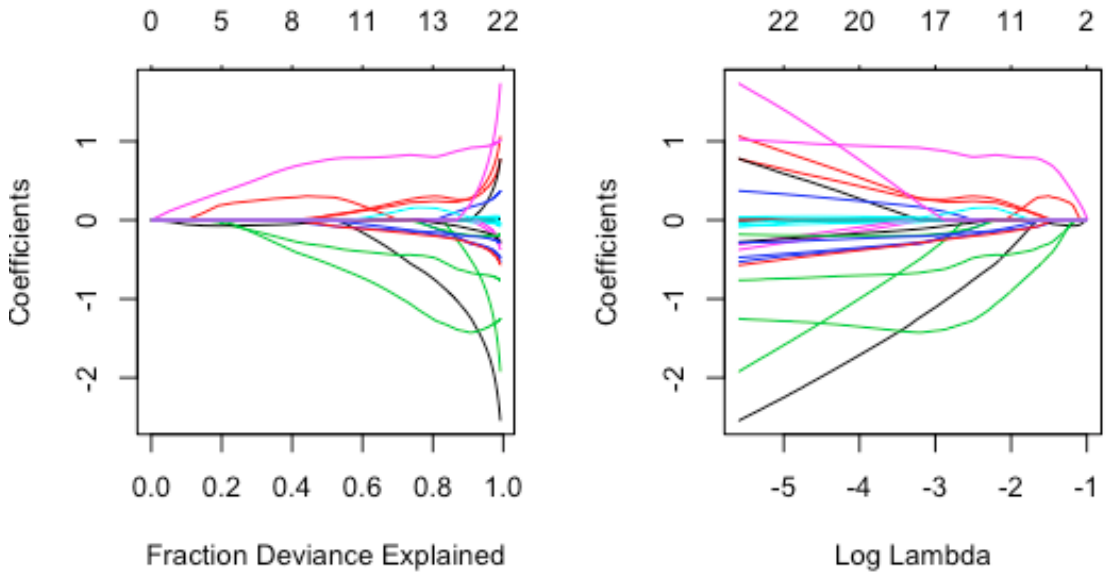

Figure 5: The above charts shows how Lasso allocates weights to the predictors. The left chart shows the spread of weights w.r.t. the total deviation explained which at the most can be 1. Notice that as we go closer to 1.0 the weights increase rapidly, this signals to stop at a point where marginal gain in deviation explained cannot justify the rapid increase in the weights. The right chart shows how feature selection changes as we increase the weight penalty. Notice that with an increasing lambda (weight penalty) the feature weights converge to zero.

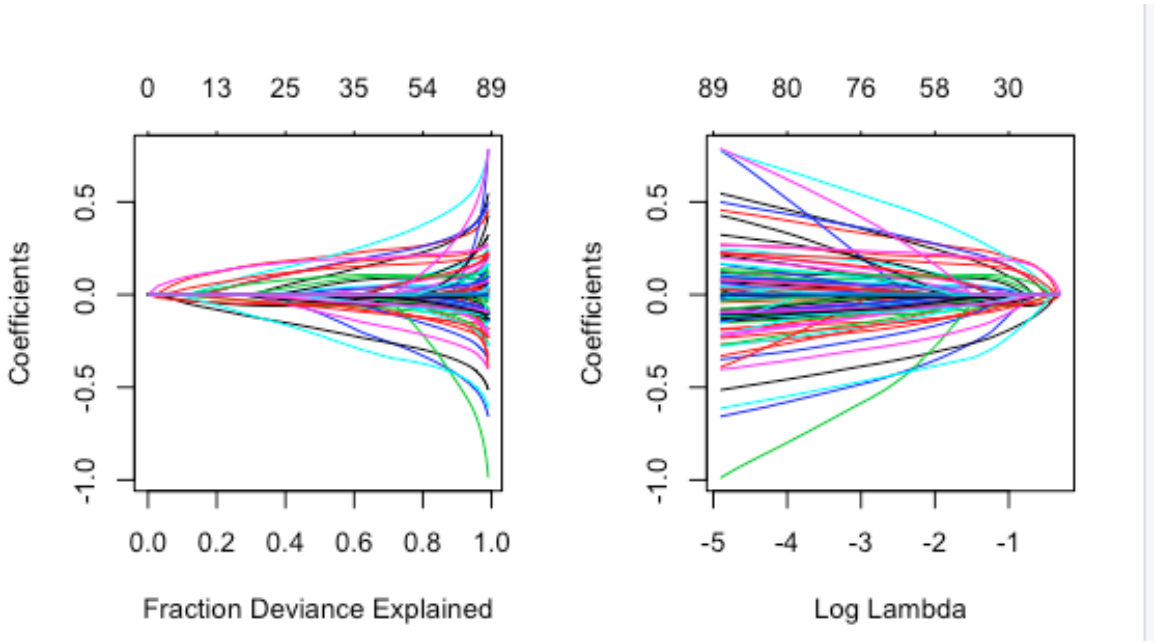

Figure 6: With ElasticNet we observe that with varying degrees of deviance explained and weight penalty lambda, compared to Lasso, more features are selected. The overall trend of number of features selected with fraction-deviation-explained or lambda remains the same as with Lasso.

it saturates. We have used "glmnet" package in R Studio to perform regularization. Figure 5 shows that lasso can select atmost 22 variables. To overcome this problem, we have used ElasticNet [1] in our report with alpha equal to 0.5. Figure 6 shows the output for Elastic net, which can select upto 86 variables.

Also, we can use XGBoost for variable selection (tree based method) as discussed in later section.

4.2 Data Visualization using t-SNE

To better visualize the high dimensional digit data set we used the algorithm, namely, t-SNE [2]. It helps to visualize the high dimensional data in 2-D or 3-D. It can be used for dimension reduction as well but we are using it for data visualization in our report.

t-SNE algorithm is mainly divided into two parts :

1. The first part of the algorithm consists of constructing a probability distribution between the high-dimensional data points in such a way that similar points have a high probability of being chosen for the embedding, while dissimilar points have a

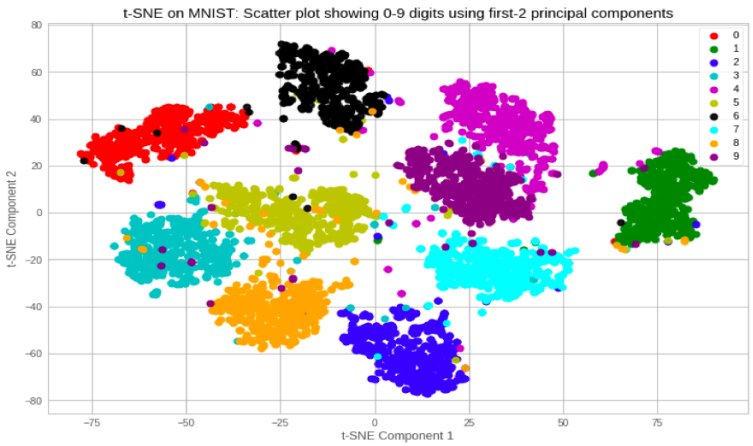

Figure 7: Visualization of digits using t-SNE

small probability of being picked. Let x_i and x_j are two points in high dimensional space. Then the conditional probability is given by using Gaussian distribution:

$$p_{i|j} = \frac{exp(-||x_i - x_j||^2}{2\sigma_i^2} \over \sum_k (1 + ||y_k - y_l||^2)^{-1} \tag{1}$$

$p_ij = \frac{p_{i|j} + p_{j|i}}{2}$. The similarities between two points y_i and y_j. Here $p_j|_i$ is a conditional probability that x_i would pick x_j as its neighbor if neighbors were picked in proportion to their probability density under the Gaussian distribution centered at x_i. Low-dimensional models of x_i and x_j are measured using a normalized heavy-tailed kernel points conditional property is given by using t-distribution. It is given by:

$$q_{i|j} = \frac{((1 + ||y_i - y_j||^2)^{-1}}{\sum_k (1 + ||y_i - y_j||^2)^{-1})} \tag{2}$$

2. Similar steps are performed in the low dimensional data and then we minimize the Kullback-Leibler divergence between the two joint probability distributions P (in high dimension) and Q (in low dimension) is given by:

$$C = KL(P||Q) = \sum_i \sum_j (p_{ij} \log \frac{p_{ij}}{q_{ij}}) \tag{3}$$

In our project, we applied t-SNE in python using Sklearn libraray. Figure 7 shows the output from python in 2-D where all 10 labels can be visualized explicitly.

4.3 Dimension Reduction using PCA

Dimension reduction maps the high dimensional data into a lower dimension. Data in low dimension retains the most of information and discard the features with very less variation within it. It helps to mitigate the computational cost as well as help to visualize the data. The MNIST data in our case has 784 dimensions, which is difficult to deal with.

In our project, we have used Principal Component Analysis for dimension reduction on the MNIST data-set. It attempts to get rid of less informative dimensions to save the computational cost. PCA constructs a small number of linear features to summarize the input data. The idea is to rotate the axes so that the important dimensions in this new coordinate system become self evident and can be retained while the less important ones get discarded. But, it should be done carefully. Reducing large number of feature may result in loosing important information. So, the idea is to minimize the reconstruction error.

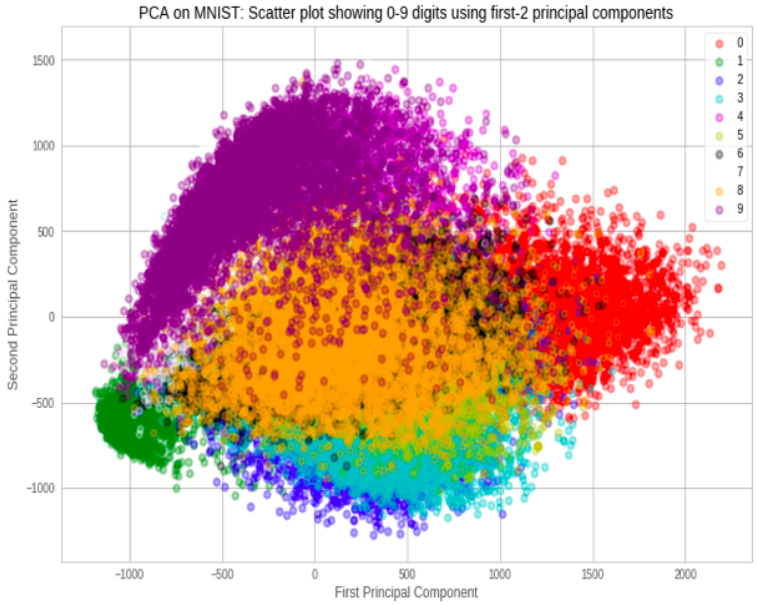

Figure 8: digit visualization using PCA

In PCA, dimensions with largest variation corresponds to one with eigen vector with largest largest eigen value. Consider a vector x (in the standard coordinate system) and some other coordinate system v_1,...,v_d, z_1, . . . , z_d can be defined as new coordinate

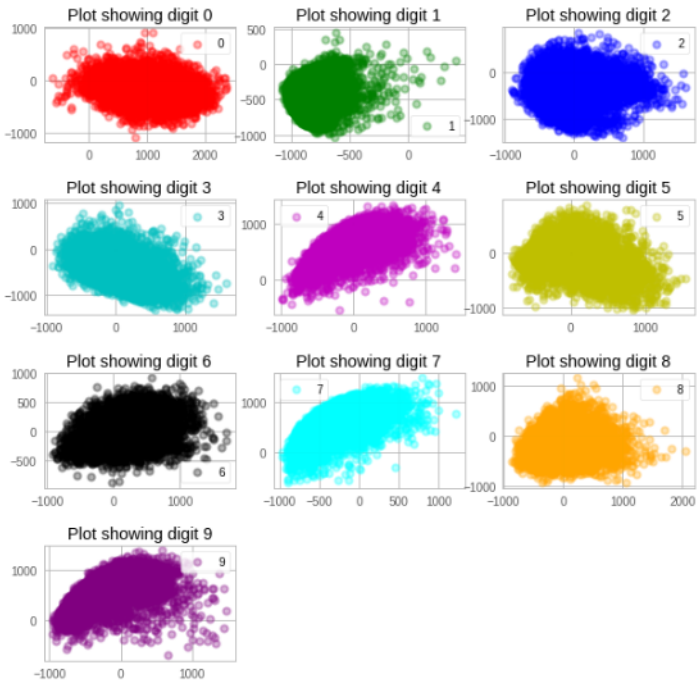

Figure 9: Representation of each digit using classwsie PCA

system. Considering all components of new coordinate system:

$$x = \sum_{i=1}^{d} z_i v_i \tag{4}$$

Now consider the top k components of new coordinate system:

$$\hat{x} = \sum_{i=1}^{k} z_i v_i \tag{5}$$

Reconstruction error is defined as the difference between equation (4) and equation (5). It is captured as:

$$||x - \hat{x}||^2 = \sum_{i=k+1}^{d} (z_i)^2 \tag{6}$$

In order to have good coordinate system, this reconstruction error has to be as small as possible. In our project, after the centering the deskewd data we use the number of features corresponding to 95 percent of variance explained by the new coordinate system. It gives us 115 number of components. Figure 5 shows the relation between

variance explained by the PCA and corresponding number of components. Figure 4 shows the comparison between image with 784 features and image with 115 features. We can see that our PCA does not loose important information and at the same time reduce the dimension to save the computation time. Figure 9 shows the class-wise global PCA in 2 dimensions.

Figure 10: The first row shows the numbers without any transformation. Second row shows deskewed digit

It was very natural that every person has different handwriting which poses several challenge to machine. One of the challenge is change in orientation of each digit. This can lead to mis-classification of a digit. Therefore, we deskewd each digit to improve the classification. Figure 10 shows some examples of deskewed image compared to raw image from our code.

5 Chapter 5: Classifiers

5.1 Logistic Regression

In statistics, logistic regression is a regression model where the dependent variable is categorical. In our case for two data-set, the dependent variable only has two categories, which are 0 and 1. We call this a binary case. If the dependent variable has more than two categories, it is referred to as multinomial logistic regression Logistic regression has similarities to regression and classification model, as the output is real (like regression) but bounded (like classification). The logistic regression model is given by

$$h(x) = \theta(W^T X) \tag{7}$$

where θ is the sigmoid function whose output is given between 0 and 1.

$$\theta(z) = \frac{1}{1 + \exp^{-z}} \tag{8}$$

The output can be interpreted as a probability for a binary event. Linear classification also deals with a binary event, but the difference is that the "classification" in logistic regression is allowed to be uncertain, with intermediate values between 0 and 1 reflecting this uncertainty the logistic function is refereed to as a soft threshold.

By default, people use 0.5 as the cutoff for classification in logistic regression, which means in this case, the points with probabilities in the interval (0,0.5) belong to category 0 and the points with probabilities in interval (0.5,1) belong to category 1

We trained our data using logistic regression classifiers using Sklearn in python.With credit card default data set, we first split the data-set into ratio of 3:7 (test:train).Logistic regression classifier gave accuracy rate of 0.784. But, as per the confusion matrix (figure 11), this classifier is not able to predict defaulted person as defaulted at all. Although, it's giving good accuracy but its inability to work with defaulted data points makes it a weak model. After dropping variables such as 'SEX', 'AGE','MARRIAGE', 'EDUCATION','BILL_AMT1','BILL_AMT2','BILL_AMT3', 'BILL_AMT4','BILL_AMT5' ,'BILL_AMT6','PAY_AMT1','PAY_AMT2','PAY_AMT3', 'PAY_AMT4','PAY_AMT5', 'PAY_AMT6', accuracy is 0.812. Also, as per confusion matrix our classifier is able to predict defaulted person as defaulted better than the previous model.

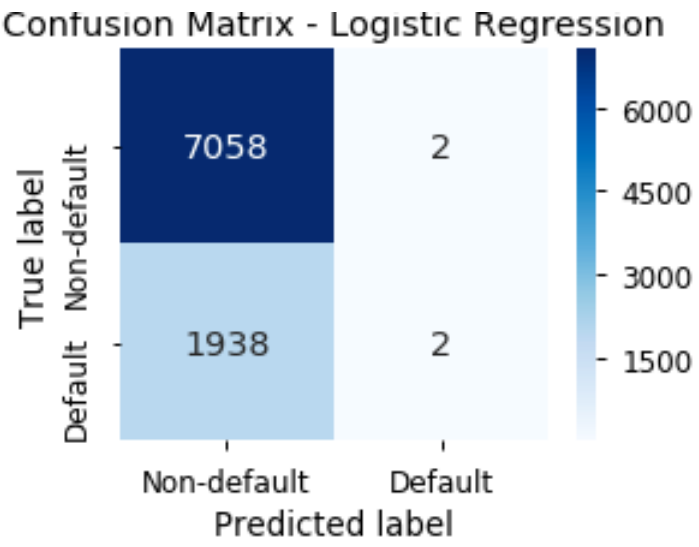

Figure 11: Confusion Matrix for Logistic Regression

From the correlation plot (figure 4), we dropped the variables which has small correlation with response variables. Therefore, after dropping these variables test accuracy is 0.787.

We also applied logistic regression on leukemia dataset. To apply the classifiers, we first coded the type of leukaemia as a 0–1 response y where 0 corresponds to ALL (Acute myeloid leukemia) and 1 corresponds to AML (Acute myelogenous leukemia). Then split the data-set with training size of 50 and test size of 22. Predicted accuracy is 0.95 and according to the confusion matrix only 1 sample in the test set was misclassified.

In MNIST data-set, it is the extended case of classical logistic regression. Since we have more than two outcomes. The probability that the responses on observation i take on one of the m+1 possible outcomes can be modeled as

$$P(y_i) \leq \frac{\exp[x_i'\beta^m]}{1 + \sum_{j=1}^{m} \exp[x_i'\beta^m]} \tag{9}$$

Therefore, we applied logistic regression for one-vs-all and one-vs-one model for data set with 115 and 46 dimensions. It is observed that one vs rest classification has maximum error rate for 85% variance compared to other.

5.2 Support Vector Machine

5.2.1 Linearly Separable case

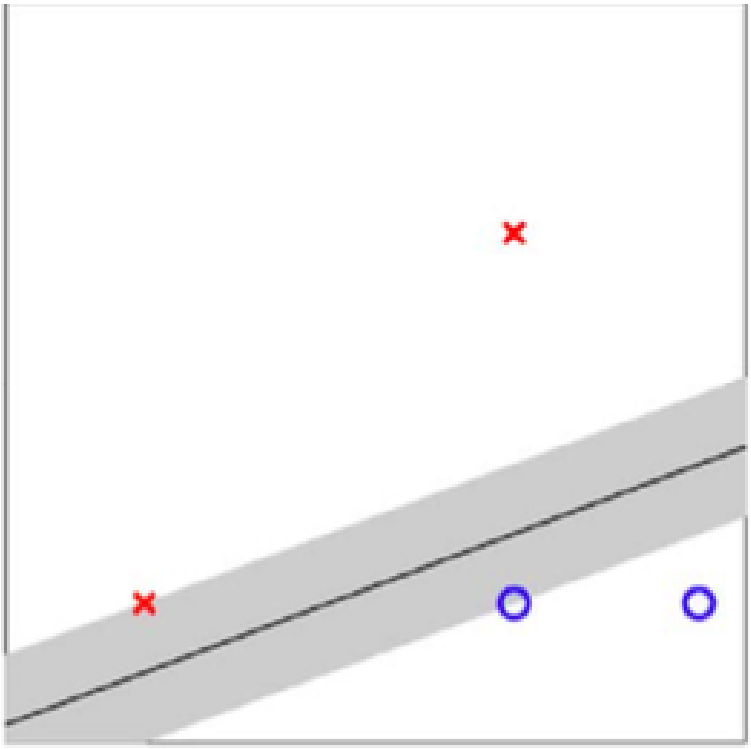

Figure 12: Support Vector Machine for linearly separable data-set

A data set in R^2 is said to be linearly separable if there exist at least one line in the plane that can separate one class of the data set on one side of the line and other class on the other side of the line. If the data set is multidimensional, then a hyper-plane is used. Perceptron Learning Algorithm finds the line/hyper plane which can linearly separate the data and there are infinitely many lines/hyperplane which can separate the data. The aim is to obtain the optimized hyper plane which can seperate the data. This is when Support Vector Machine comes into picture. The SVM uses a safety cushion' when separating the data. As a result, SVM is more robust to noise; this helps combat over-fitting.This classification method was introduced by Vapnik et al [3].

The steps follows the next:

Let's consider a d dimensional data which can be represented as (x_i, y_i) for i = 1,2,...n and labels as $\{-1, 1\}$. Suppose the data is lineally separable, equation of the

hyper will be given as:

$$W^T x + b = 0 \tag{10}$$

where w is normal to the hyperplane. Eqaution of hyper-plane when datapoint belongs to the -1 will be given by

$$W^T x + b = -1 \tag{11}$$

and the one which belongs to +1 will given by

$$W^T x + b = 1 \tag{12}$$

Both the hyperplanes are parallel to the $W^T x + b = 0$ and we want to maximize the separation of both the classes. Let m be the width between the hyperplanes $w^T x + b = 1$ and $w^T x + b = +1$. Our goal is to maximize the width between the hyperplane such that no point is misclassified (hard threshold). It can be written as maximization problem as follows:

$$\begin{aligned} &\min_{x \in Rd} && ||w||^2 \\ &\text{subject to} && y_i(w^T x_i + b) \leq 1,\ i = 1, \ldots, m. \end{aligned} \tag{13}$$

This hard threshold SVM works well if data is linearly separable. In our case, it performed very poor on Leukemia data-set and gave the 0.5 accuracy. Also, it performed fairly on MNIST and credit card dataset.

5.2.2 Soft Margin Extension

If the data is not perfectly linearly separable or "almost" linearly separable we may allow some data points in one class to appear on the other side of the boundary. To accommodate this misclassification, a slack variable was introduced ξ_n

$$\begin{aligned} &\underset{b,W,\xi}{\text{minimize}} && \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}\xi_n \\ &\text{subject to:} && y_n(\mathbf{w}^T\mathbf{x}_n + b) \leq 1 - \xi_n, \\ & && \xi_n \geq 0, \qquad\qquad i = 1, \ldots, N. \end{aligned} \tag{14}$$

The constant C is the regularization parameter, a trade-off between large margin and noise tolerance. Small C corresponds to large margin whereas large C corresponds

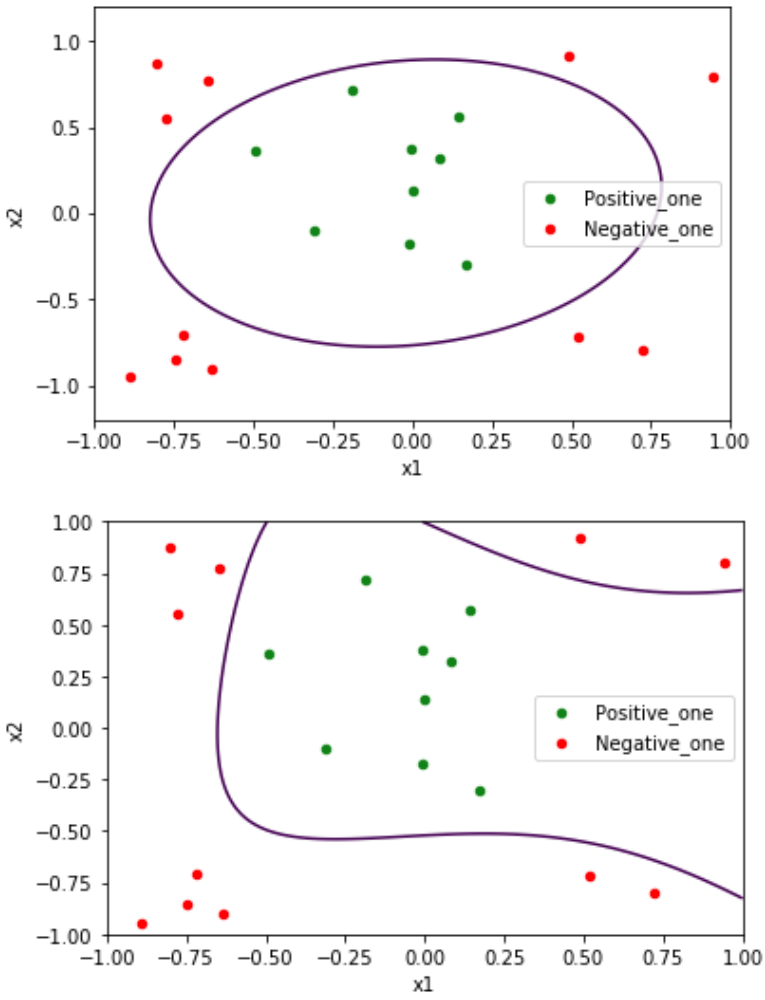

Figure 13: SVM with second and third degree polynomial kernel

to small margin (less violations).Figure 13 shows the SVM with second and third degree polynomial kernel.

5.2.3 Kernel Trick

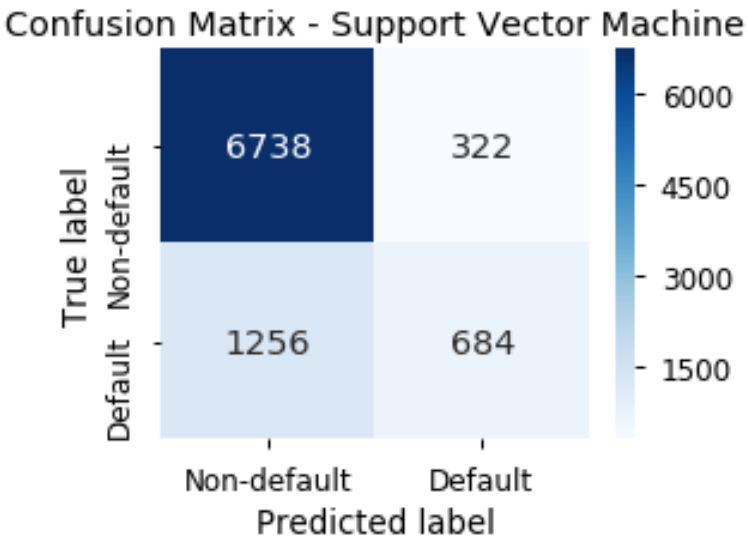

Figure 14: Confusion matrix for support vector machine

How to use non linear transformation without physically transforming data to Z-space (as shown in figure 13). Since the kernel function only depends explicitly on the original data , we can apply the usual linear SVM in the original data space instead of the high dimensional feature space which may have infinite dimensions. We do not need to calculate the actual transformation Φ, everything is embedded in the kernel function. In practice, there are three common used kernel functions: Polynomial Kernel, Gaussian Kernel, Sigmoid Kernel.

For credit default data-set, SVM with polynomial kernel were able to classify defaulted correctly, whereas Logistic regression cannot. Figure 14 shows the confusion matrix for SVM applied in Python. Since, SVM has the feature of automatic regularization, it gave the best result for Leukemia data-set with accuracy rate of 0.94. We also implemented SVM with Gaussian kernel on Leukemia data-set with different value of parameter (gamma and C). Gamma in Gaussian kernel can seen as the inverse of the radius of influence of samples selected by the model as support vectors. It can be

defined as how far the influence of a single training example reaches, with low values meaning ” far” and high values meaning ”close”. It is given by:

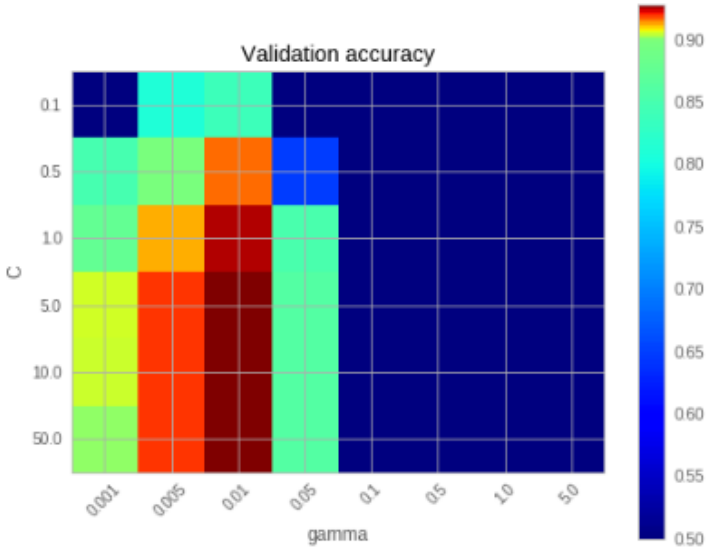

Figure 15: Chart showing the score against each of the model parameters inside the grid

$$k(x_i, x_j) = \exp \frac{-||x_i - x_j||^2}{2\sigma^2} \tag{15}$$

As me increase the value of gamma for Leukemia data-set, accuracy rate decreases.

For MNIST dataset, the challenge in Gaussian kernel is to get the optimal sigma. since, its the multi class problem [5], optimizing the parameters is time taking. Due to time constrains We tuned the model parameter using grid search. Since, running Gaussian kernel is really computationally costly. We used this classifier with 20 percent of the training data set. Figure 14 shows the grid search output. Best parameter found are ”C” = 5, gamma = 0.01. Gaussian kernel gives the the error rate of 2 % when trained when applied on 20 percent training data points.

5.3 k-nearest neighbor

K-nearest neighbour is one of the easiest classification algorithms. It can also be used for prediction, but it is widely used for classification problems. it is simple and easy to interpret. Consider the figure 16, there are two types of circle: blue circle and orange

circle. To predict the class of new data point as shown in figure, we consider the label of nearest K neighbours around the new data point. In our case K=3, out of which two are blue and one is orange, therefore label of new data point will be as orange class. To get the optimum value of K, we have used k-fold cross validation.

For credit card default-data-set, figure 17 shows that as we increase the value of k cross-validation error is decreased, but it can led to over-fitting. Therefore, we decided to choose k =3. We performed cross validation with other data-set as well to get the optimum K.

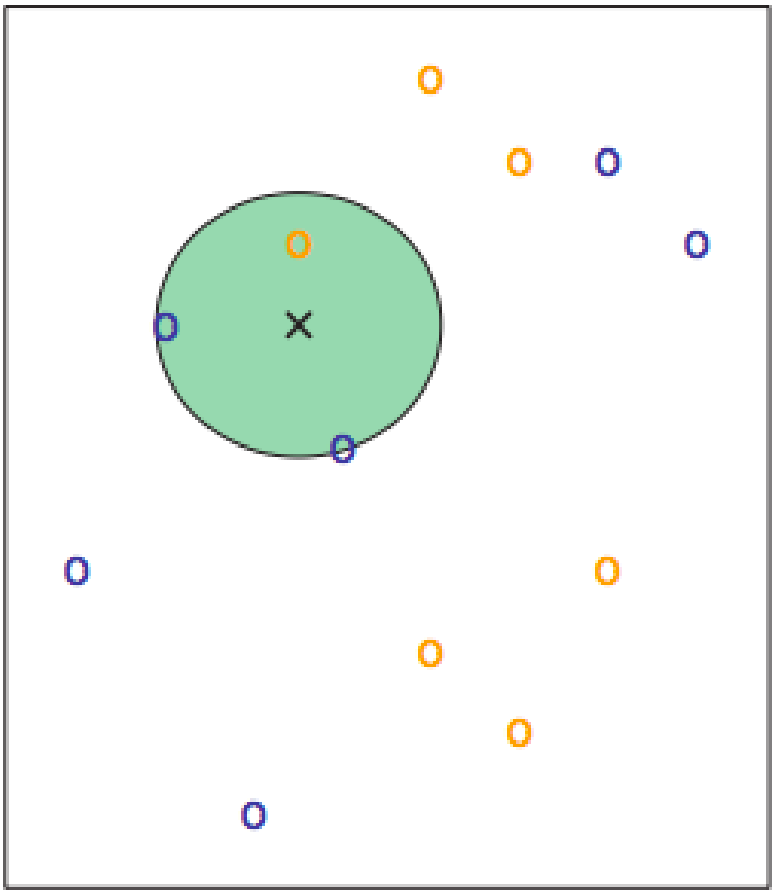

Figure 16: KNN

5.4 Tree Based Methods

Decision trees can be applied to both classification and regression problems. Tree based algorithms involve segmenting the predictor into a number of simple regions. We use

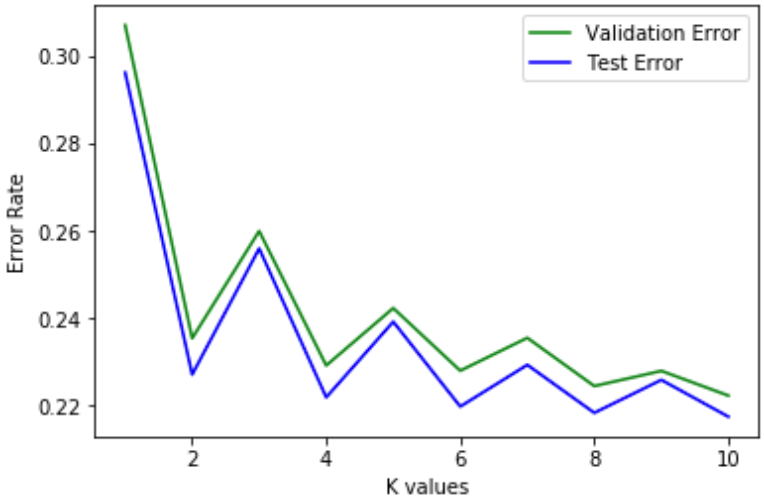

Figure 17: K-fold cross validation for credit card default data-set to get the optimum value of K

mean or mode of training observations for each predictor space to predict or classify new observation as shown in figure 17.

Tree based algorithms follows top-down, greedy approach which is known as recursive binary splitting. It is top-down because it begins at the top of tree and split the into parts based on condition defined. It is greedy approach because it looks for best split at each rather than looking ahead and pick the best split which will gives the minimum residual error. In our project we mainly focused on classification decision tree.

For classification, error rate is given below:

$$E = 1 - max_k(p_{mk}) \tag{16}$$

where p__m_k is proportion of observations from the k^{th} class in the m^{th} region.

There are several advantages of tree based methods such as, easy to interpret, simple concept, can be displayed graphically, and can easily handle qualitative predictors without the need to create any dummy variables. But, there are few disadvantages for example trees can be very non-robust. Therefore, to increase the predictive performance, we can use aggregate several trees using methods like bagging, random forests, and boosting.

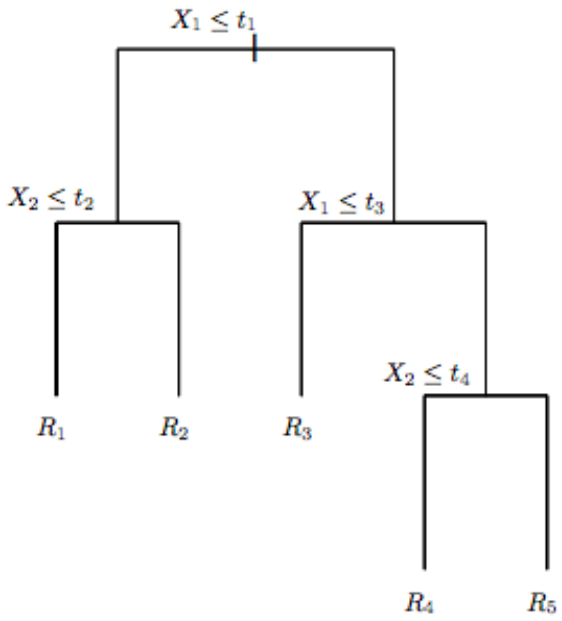

Figure 18: Tree based algorithm for classification or prediction

5.4.1 Bagging

Bagging uses the concept of Bootstrapping (figure 19) which is defined as randomly picking few observations with replacement and repeating this n times. Ordinary tree methods suffers from high variance which means if we split the tree into two halves at random and fit a tree, then results could be different. Our aim is to reduce the variance. Bootstrapping, aggregation, or bagging is the general approach to reduce the variance. The average of all the predictions is given by:

$$f_{bag}(x) = \frac{1}{B}(\sum_{b=1}^{B} f*^{b(x)}) \tag{17}$$

where B represents number of times training samples taken with replacement. This is called bagging. In case of classification, for each test observation we can record the class predicted by each of the B trees and take one with maximum vote.

5.4.2 Random Forests

Random Forest is the special case of bagging. It provides an important improvement over bagging by the way of small tweak that decorrelates the trees. In random forest,

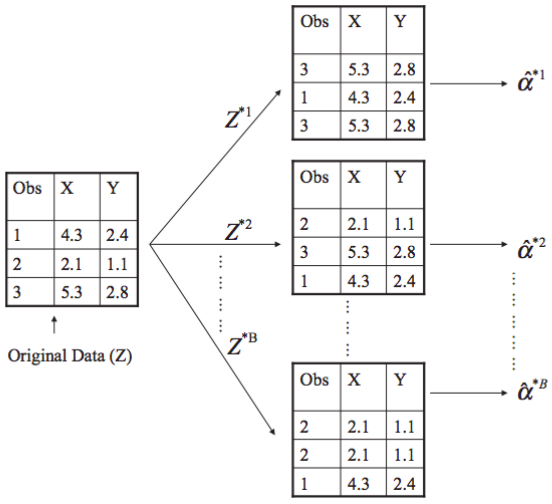

Figure 19: Bootstrapping

square root of of the total number of predictors is considered at each split to improve the accuracy.

5.4.3 Boosting

Boosting does not involve bootstrap sampling: instead trees are grown sequentially which means each tree is grown using information from the previous tree.

5.4.4 XGBoost

XGBoost is a package that add things to gradient boosted trees to improve their performance. XGBoost is implementation of gradient boosted trees with some added features to improve on over-fitting.and it's implemented in such a way that it leverages all the available computing resources in an efficient way making the computation faster. Figure shows the XGBoost applied on Leukemia data-set. As discussed in feature selection that it can be used to select important features. It selected only six features out of approximately 7000 (figure 20).

5.5 Convolutional Neural Network

Neural networks are a set of algorithms that are designed to recognize patterns. They are structured like human brain. They receive an input and transform it through series

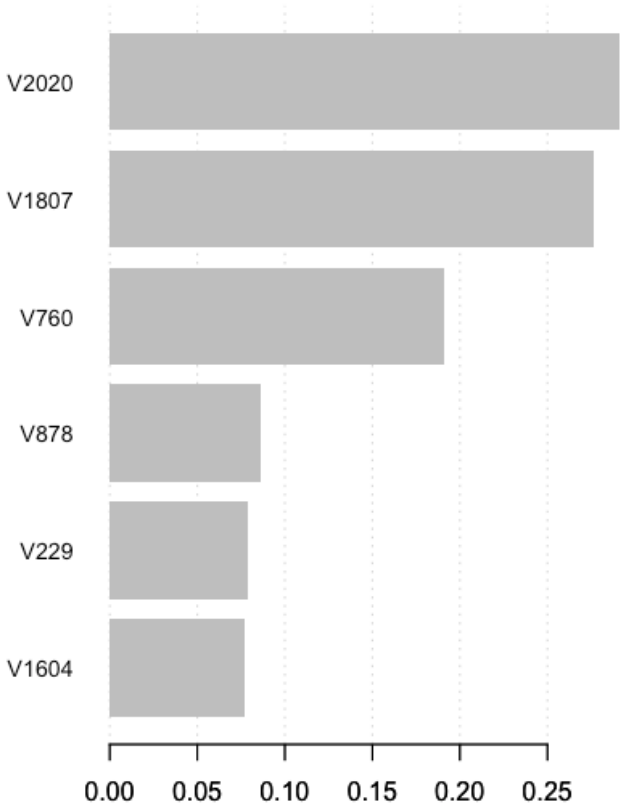

Figure 20: XGBoost to select important features with Leukemia data-set

of hidden layers. Each hidden layer is made up of neurons and which are independent of each other and is fully connected to neuron in previous layer. Last output layer is the output layer which classify the as per the problem statement. But, it can led to complex neural network when we have large size input. For example, if there is an input image of size 200*200*3 then it would led to learn 120000 weights which seems not manageable as well and learning huge amount of weight can lead to over-fitting.

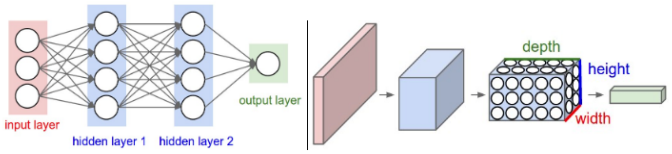

Figure 21: left: Neural Network, Right: Architecture of Convolutional Neural Network

Unlike neural network, Convolutional Neural Network (ConvNet) are 3-dimensional. This means that the neurons are arranged in 3D: length, breadth, height as shown in the figure 21. ConvNet strucrue is divided into three parts:

- Convolutional Layer: It is the main block which do all computation. It consists of filters (or kernels)with small receptive field but more depth. During the forward pass, each filter convoluted through the input volume and compute the dot products between the entries of filter and entries in input space. As shown in the figure 22

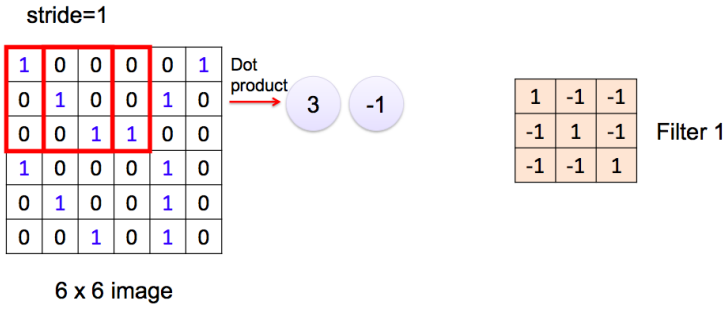

Figure 22: Shows the 6*6 input image and 3*3 size filter

- Pooling Layer: It is common to periodically insert a Pooling layer in-between successive Conv layers in a ConvNet architecture. Its function is to progressively

reduce the spatial size of the representation to reduce the amount of parameters and computation in the network, and hence to also control over-fitting.

- Fully Connected Layer: This is the traditional neural network with less weight to learn

Figure shows the three component of ConvNet.

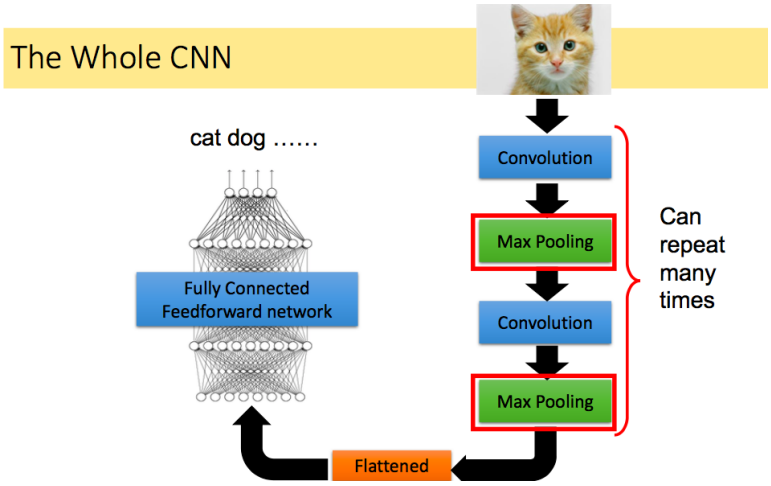

Figure 23: The whole CNN consist of: ConvNet, Pooling layer, Fully connected layer

We have applied CNN on MNIST to improve the accuracy with filter size of 3* and used softmax as activation function.

6 Chapter 6: Results and Analysis

Table 1 summarizes the accuracy rate of all classifiers applied on all the three data-set. As discussed all these experiments are done in Jupyter notebook (XGBoost in R Studio). Column 1 gives the name of the data-set, Column 2 gives information about the classifier and its parameter. Column 3 gives the accuracy rates. We can see there is no winner among any classifier. For credit card data-set, boosting and SVM with 2 degree polynomial kernel gave the better result. For Leukemia data-set, SVM with Gaussian Kernel is giving the highest accuracy and for MNIST CNN improved the accuracy.

Data Set	***Classifier and its parameters***	***accuracy rates***
default of credit card clients	Linear SVM	0.7873
default of credit card clients	Logistic Regression	0.784
default of credit card clients	KNN with k = 1	0.702
default of credit card clients	KNN with k = 3	0.7451
default of credit card clients	KNN with k = 6	0.7798
default of credit card clients	2 degree polynomial kernel	0.82
default of credit card clients	Random forest	0.8074
default of credit card clients	Boosting	0.82
leukemia (p >>n)	KNN with k = 3	0.863
leukemia (p >>n)	linear SVM	0.636
leukemia (p >>n)	SVM (with 3 degree polynomial Kernal, C = 0.1	0.5
leukemia (p >>n)	SVM (with 3 degree polynomial Kernal, C = 1	0.7727
leukemia (p >>n)	SVM (with 3 degree polynomial Kernal, C = 100	0.954
leukemia (p >>n)	SVM (with gaussian Kernal, gamma = 0.0001, C = 1000	0.954
leukemia (p >>n)	SVM (with gaussian Kernal, gamma = 0.0001, C = 1000	0.954
leukemia (p >>n)	SVM (with gaussian Kernal, gamma = 0.001, C = 1000	0.636
leukemia (p >>n)	SVM (with gaussian Kernal, gamma = 100, C = 1000	0.5
leukemia (p >>n)	XGBoost (selected 5 features)	0.834
MNIST digit (Multiclass)	Logistic regression (one vs one), deskewing and PCA (115 components)	0.9626
MNIST digit (Multiclass)	Logistic regression (one vs one), deskewing and PCA (46 components)	0.9639
MNIST digit (Multiclass)	Logistic regression (one vs all) deskewing and PCA (115 components)	0.9514
MNIST digit (Multiclass)	Logistic regression (one vs all) deskewing and PCA (46 components)	0.946

MNIST digit (Multiclass)	Logistic regression (multinomial) deskewing and PCA (115 components)	0.9597
MNIST digit (Multiclass)	Logistic regression (multinomial) deskewing and PCA (46 components)	0.9507
MNIST digit (Multiclass)	kNN deskewing and PCA (115 components)	0.9855
MNIST digit (Multiclass)	kNN, deskewing and PCA (46 components)	0.9849
MNIST digit (Multiclass)	Linear SVM, deskewing and PCA (115 components)	0.9637
MNIST digit (Multiclass)	3 degree polynomial SVM (one vs one) deskewing and PCA (115 components)	0.988
MNIST digit (Multiclass)	3 degree polynomial SVM (one vs rest) deskewing and PCA (46 components)	0.9844
MNIST digit (Multiclass)	3 degree polynomial SVM (one vs all) deskewing and PCA (115 components)	0.98
MNIST digit (Multiclass)	3 degree polynomial SVM (one vs rest) deskewing and PCA (46 components)	0.9864
MNIST digit (Multiclass)	Bagging, deskewing and PCA (115 components)	0.94
MNIST digit (Multiclass)	Bagging deskewing and PCA (46 components)	0.9538
MNIST digit (Multiclass)	Random Forest deskewing and PCA (115 components)	0.9672
MNIST digit (Multiclass)	Random Forest deskewing and PCA (46 components)	0.974
MNIST digit (Multiclass)	CNN, activation function = softmax, filter size 3*3	0.9932

Table 1: Model Comparison

7 Chapter 7: Conclusion

We have worked on three different data-set. It is clear that we can not always follow the same procedure to build a model. For the credit data sat, we did feature selection by dropping the variables which has the low co-relation with the response variable. Whereas, with Leukemia data set there were large number of predictors compare to the number of observations. We used ElasticNet and XGBoost to select the important features. But, it was not the case with MNIST data-set, we were more interested in reducing the dimension of the data-set because building classifier on original data-set was computationally expensive.We applied PCA and reduced our dimension to 115 components (explaining 95% variance in data) and 46 components (explaining 85% variance in data) from 784 components. However, PCA is not very useful if the predictors are almost uncorrelated. Also, there is no best algorithm to apply. it depends on type of problem we are solving. For example for credit default data-set, Logistic Regression was not able to classify defaulter person. But we can improve this by changing the cut off value (we have taken 0.5 in our report). On the other hand it preformed well with other two data-sets. KNN also performed good on all three, but we need to be careful while choosing the best K using k-fold cross validation, else we can over-fit our data if we aim to get the lowest validation error. Other classifiers such as Random Forest and SVM have inbuilt regularization which reduces our work. Other classifiers such as XGBoost can also be used for classification/prediction as well as important feature selection.

References

[1] H. Zou,and T. Hastie, "Regularization and variable selection via the elastic net" Stanford University, USA, September 2004.

[2] L. Maaten, G. Hinton,"Visualizing Data using t-SNE",9(Nov):2579–2605, 2008.

[3] Y.LeCun, L. Bottou,C. Cortes,H.Drucker,I.Guyon,V. Vapnik," Comparison of Classifier Methods: A casestudy in handwritten digit recognition",

[4] C.Hsu, C. Lin,"A comparison of methods for multiclass support vector machines",IEEE Transactions on Neural Networks (Volume: 13 , Issue: 2 , Mar 2002)

[5] C.Burges, B. Scholkopf, "Improving the Accuracy and Speed of Support Vector Machines",Denver, Colorado — December 03 - 05, 1996

[6] In J. H. Oh, C. Kwon, S. Cho (Eds.), Neural networks: The statistical mechanics perspective (pp. 261-276). World Scientific.

[7] H. Zou, T. Hastie, and R. Tibshirani, "Sparse Principal Component Analysis", 01 Jan 2012

[8] Y.LeCunn, L. Bottou, Y. Bengio,"Gradient Based Learning Applied to Document Recognition",1998

[9] CS231n Convolutional Neural Networks for Visual Recognition, Stanford.

[10] Learning from Data authored by S. Yaser, M. Malik, L. Hsuan-Tien