ARTIFICIAL NEURAL NETWORK BASED FEATURE SELECTION TECHNIQUES

THESIS SUBMITTED

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE OF

MASTER OF ENGINEERING

IN

ELECTRONICS & TELECOMMUNICATION ENGINEERING

By

PARIKSHIT KUNDU

Examination Roll No: M4ETC19024

Registration No: 140705 of 2017-18

Under the guidance of

Prof. SHELI SINHA CHOUDHARI

Department of Electronics & Telecommunication Engineering

Jadavpur University, Kolkata – 700 032

West Bengal, India

May-2019

Faculty of Engineering & Technology
Jadavpur University

This is to certify that the thesis entitled **—ARTIFICIAL NEURAL NETWORK BASED FEATURE SELECTION TECHNIQUES** has been carried out by **PARIKSHIT KUNDU** (Class Roll No: 001710702022, Examination Roll No.: M4ETC19024 and Registration No: 140705 of 2017-18) under my guidance and supervision and be accepted in partial fulfilment of the requirement for the degree of Master of Electronics & Telecommunication Engineering.

Prof. Sheli Sinha Chaudhuri
Supervisor
Department of Electronics and
Telecommunication Engineering
Jadavpur University
Kolkata-700032

Prof. Sheli Sinha Chaudhuri
Head of the Department
Electronics and Telecommunication
Engineering
Jadavpur University
Kolkata-700032

Prof. Chiranjib Bhattacharjee
Dean
Faculty Council of
Engineering and Technology
Jadavpur University
Kolkata-700032

Faculty of Engineering & Technology

Jadavpur University

CERTIFICATE OF APPROVAL

The forgoing thesis titled **―ARTIFICIAL NEURAL BASED FEATURE SELECTION TECHNIQUES** is here by approved as a creditable study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or accept every statement made, opinion expressed or conclusion drawn there in but approve the thesis only for the purpose for which it has been submitted.

Committee on Final Examination for
Evaluation of the Thesis:

Additional Examiner

Supervisor

DECLARATION OF ORIGINALITY AND COMPLIANCE OF ACADEMIC ETHICS

I hereby declare that this thesis contains literature survey and original work by the undersigned candidate, as part of his Master of Electronics and Telecommunication studies.

All information in this document have been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name	**:**	**PARIKSHIT KUNDU**
Exam Roll No	**:**	**M4ETC19024**
Thesis Title	**:**	**ARTIFICIAL NEURAL NETWORK BASED FEATURE SELECTION TECHNIQUES**

Signature of Candidate

Acknowledgement

This thesis is the result of the work whereby I have been accompanied and supported by many people. It is a pleasant aspect that I have now the opportunity to express my gratitude to all of them.

With immense pleasure, I express my sincere gratitude, regards and thanks to my project guide Prof. Sheli Sinha Choudhari for her excellent guidance, invaluable suggestions and continuous encouragement at all the stages of my research work. Her interest and confidence in me was the reason for all the success I have achieved. I have been fortunate to have him as my guardian than guide as she has been a great influence on me, both as a person and as a professional.

Above all, I extend my deepest gratitude to my parents, for their invaluable love, affection, encouragement and support and for their struggle to make me a successful person in spite of many obstacles.

PARIKSHIT KUNDU

Abstract

Selection of variable and feature for an efficient prediction model have become the focus of much research, in areas of application for which data sets with tens or hundreds of thousands of variables are available. These areas include text processing of internet documents, gene expression array analysis, and combinatorial chemistry. The objective of variable selection is three-fold: improving the prediction performance of the predictors, providing faster and more cost-effective predictors, and providing a better understanding of the underlying process that generated the data. The contribution of this thesis cover a wide range of aspects of such problems. In this thesis some popular feature selection methods have been described and a novel feature selection methods using deep learning algorithm have been proposed. Feature selection in deep learning is a very focused research topic. Which can describe that feature selection in deep learning can improve the performance of the prediction model for significantly and also reduce the complexity of the deep learning model. Feature selection can give some good idea about which feature have more importance and which feature do not have any importance for a particular problem.

In this thesis the classification problem has been solved and generalized methods of feature selection have been presented, that is also applicable for any other supervised classification problem having more number of features and less number of samples.

Contents

List of Figures

Chapter 1

Overview on Feature Selection

Introduction

In machine learning and statistics, feature selection, also known as variable selection, attribute selection or variable subset selection,is the process of selecting a subset of relevant features (variables, predictors) for use in model construction. The central premise when using a feature selection technique is that the data contains many features that are either redundant or irrelevant, and can thus be removed without incurring much loss of information Redundant or irrelevant features are two distinct concepts. Since one relevant feature may be redundant in the presence of another relevant feature with which it is strongly correlated.

Feature selection techniques should be distinguished from feature extraction. Feature extraction creates new features from functions of the original features, whereas feature selection returns a subset of the features. Feature selection techniques are often used in domains where there are many features and comparatively few samples (or data points). The main conventional cases for the application of feature selection include the analysis of written texts and DNA micro array data, where there are many thousands of features, and a few tens to hundreds of samples. It is also applicable where individual feature for a particular problem cannot defined.

When applying data mining and machine learning algorithms on high-dimensional data, a critical issue is known as the curse of dimensionality [1]. It refers to the phenomenon that data becomes scattered in high-dimensional space.This can significantly affecting algorithms designed for low-dimensional space. In addition, with the existence of a large number of features, learning models tend to over-fit which may cause performance degradation on unseen data. Moreover, data of high dimensionality significantly increases the memory storage requirements and computational costs for data analytics.

Dimensionality reduction is one of the most powerful tools to address the previously described issues. It can be categorized into two main components: feature extraction and feature selection. Feature extraction projects the original high-dimensional feature space to a new feature space with low dimensionality. The newly constructed feature space is usually a linear or nonlinear combination of the original feature space. Examples of feature extraction methods include Principle Component Analysis (PCA)[2], Linear Discriminant Analysis (LDA)[3], Canonical Correlation Analysis (CCA)[4], Singular Value Decomposition[5], ISOMAP[6] and Locally Linear Embedding (LLE)[7], 2000). Feature selection, on the other hand, directly selects a subset of relevant features for the use of model construction. Lasso[8], Information Gain[9], Relief[10], MRMR[11], Fisher Score [12], Laplacian Score[13], and SPEC[14] are some of the well-known feature selection techniques.

Both feature extraction and feature selection have the advantages of improving learning performance, increasing computational efficiency, decreasing memory storage requirements,and building better generalization predictive models which are better accuracy and performance. However, since feature extraction builds a set of new features, further analysis is problematic and cannot getting the physical meaning of these features in the transformed space. In contrast, by keeping some original features, feature selection maintains physical meanings of the original features and gives models better readability and interpretability. Therefore, feature selection is often preferred in many real-world applications such as text mining and genetic analysis because it can give the actual subset of relevant features or variables. That's why this fasilitate better understanding of the problem and to more accurate solution.

In this section we can briefly describe what is the connotation of feature and relevant, irrelevant feature. Then we can recount this with a simple example.

1.1 Feature

In machine learning and pattern recognition, a feature is an individual measurable property or characteristic of a phenomenon being observed.In any machine learning and pattern recognition model feature plays a vital role for the prediction.

1.1.1 Relevant Feature

Feature is categorized to be of two basic type, Relevant and Irrelevant feature.Relevant feature are those which can improve the model in very great extent.In Machine learning and pattern recognition problem first we can find out the most important or relevant feature and then using those feature we can make a good predictive model.

1.1.2 Irrelevant Feature

Irrelevant feature are those feature which doesn't have any significance for the particular predictive model. Irrelevant feature can cause over-fitting and also give a very bad predictive model. So the aim is to eliminate those Irrelevant feature and make a good predictive model.

Example The problem is identification of fruit. Let there are two fruits 'Orange' and 'Water melon'. let there be three features such as 'colours' , 'Shape' and 'Weights'.Among these features colour and weight can differentiate the two fruits. So this are most 'Relevant' feature.Since both are round shape so cannot be used distinguish between them. So this feature is called 'Irrelevant' feature for this particular problem.

In the next section we first categorize the different types of feature selection algorithm. we first review traditional categorizations of feature selection algorithms from the availability of labels and from the search strategy perspective. This is the most general categorizations of feature selection algorithm.

1.2 Label perspective Feature Selection

According to the availability of label information, feature selection algorithms can be broadly classified as supervised, unsupervised and semi-supervised methods.

1.2.1 Supervised Feature Selection

Supervised learning means in the training set class label is properly defined. Supervised feature selection is generally designed for classification or regression problems. It aims to select a subset of features that are able to discriminate samples from different classes. With the existence of class labels, the feature relevance is usually assessed via its correlation with class labels. A general framework of supervised feature selection is illustrated in the Figure 1.1.

The training phase of the classification highly depends on feature selection. After splitting the data into training and testing sets, classifiers are trained based on a subset of features selected by supervised feature selection. Note that the feature selection phase can either be independent of learning algorithms (filter methods), or it may iteratively take advantage of the learning performance of a classifier to assess the quality of selected features (wrapper methods). Finally,the trained classifier predicts class labels of samples in the test set on the selected features.

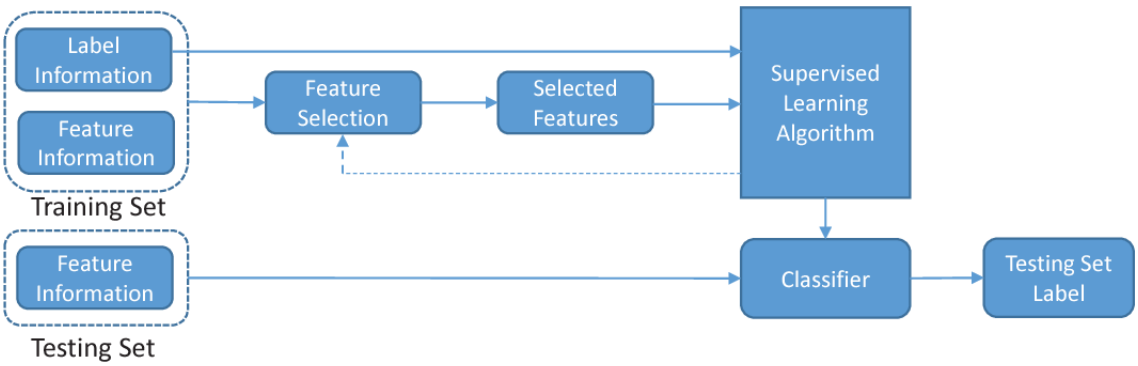

Figure 1.1: General Frame work of Supervised Feature Selection

1.2.2 Unsupervised Feature Selection

Unsupervised feature selection is generally designed for clustering problems. As acquiring labeled data is particularly expensive in both time and efforts, unsupervised feature selection on unlabeled data has gained considerable attention recently. Due to the lack of label information to evaluate feature importance, unsupervised feature selection methods seek alternative criteria such as data similarity and local discriminative information to define feature relevance. A general framework of unsupervised feature selection is illustrated in the Figure 1.2. Different from supervised feature selection, unsupervised feature selection usually uses all instances that are available in the feature selection phase. Also, the feature selection phase is either independent of the unsupervised learning algorithms (filter methods), orit relies on the learning algorithm to iteratively improve the quality of selected features (wrapper methods). After the feature selection phase, it outputs the cluster structure of all data samples on the selected features by using a typical clustering algorithm.

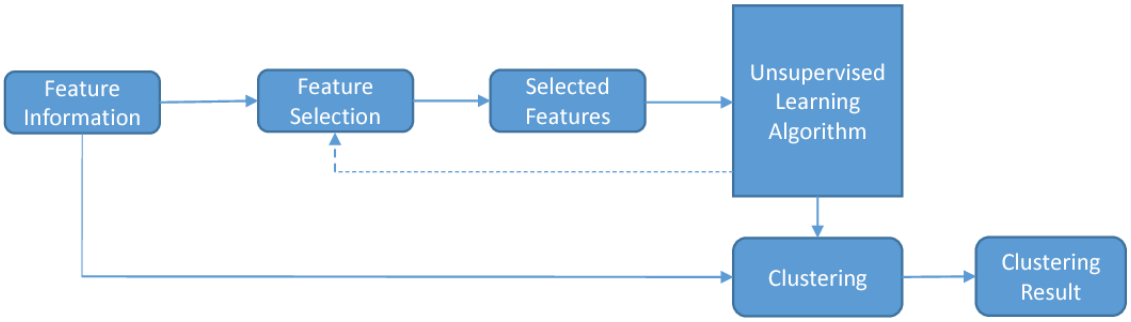

Figure 1.2: General Frame work of Unsupervised Feature Selection

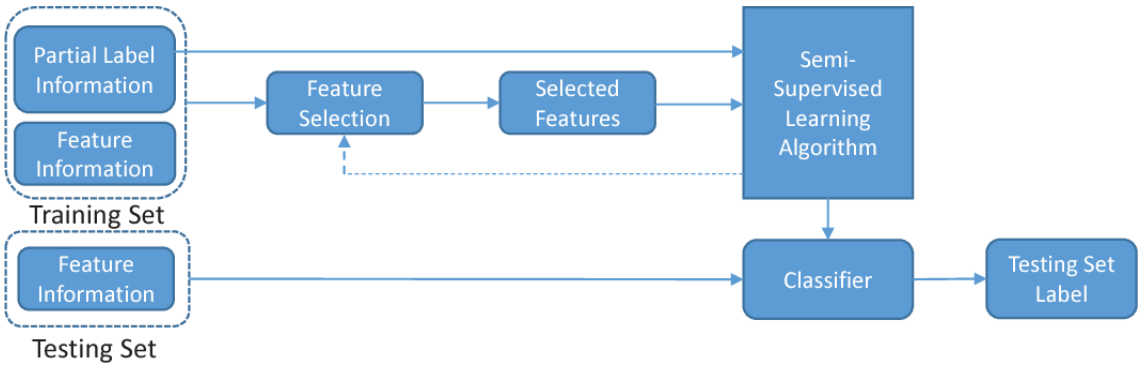

Figure 1.3: General Frame work of Semi-supervised Feature Selection

1.2.3 Semi-Supervised Feature Selection

Semi-Supervised means there are some data with perfectly label and some data are unlabeld so we cannot use supervised and unsupervised feature selection method.Supervised feature selection works when sufficient label information is available while unsupervised feature selection algorithms do not require any label information.Semi-supervised approach is mainly the combination of supervised and unsupervised learning at the same time. Semi-Supervised feature selection have good practical application. The general framework of a semi-supervised feature selection technique has been shown in the Figure 1.3.

Now I can describe feature selection in search perspective or rather feature subset selection perspective. In this section I can describe how a good feature subset can be selected from the given data set.

1.3 Search Perspective Feature selection

With respect to different selection strategies, feature selection methods can be categorized as wrapper, filter and embedded methods.

1.3.1 Filter Method

Filter methods are independent of any learning algorithms. They rely on certain characteristics of data to assess the importance of features. Filter methods are typically more efficient than wrapper methods. However, due to the lack of a specific learning algorithm guiding the feature selection phase, the selected features may not be optimal for the target learning algorithms.

Figure 1.4: General Frame work of Filter Type Feature Selection

A typical filter method consists of two steps. In the first step, feature importance is ranked by a feature score according to some feature evaluation criteria. The feature importance evaluation process can either be univariate or multivariate. In the univariate case, each feature is ranked individually regardless of other features, while the multivariate scheme ranks multiple features simultaneously. In the second step of a typical filter method, lowly ranked features are filtered out and the remaining features are kept. In the past decades, many different evaluation criteria for filter methods have been proposed. Some representative criteria include feature discriminative ability to separate samples[15,16].

1.3.2 Wrapper Method

Wrapper methods rely on the predictive performance of a predefined learning algorithm to evaluate the quality of selected features.Given a specific learning algorithm, a typical wrapper method performs two steps: (1) search for a subset of features; and (2) evaluate the selected features. It repeats (1) and (2) until some stopping criteria are satisfied or the desired learning performance is obtained. The workflow of wrapper methods is illustrated in Figure 1.5. It can be observed that the feature set search component first generates a subset of features, then the learning algorithm acts as a black box to evaluate the quality of these features based on the learning performance. The whole process works iteratively until the highest learning performance is achieved. The feature subset that gives the highest learning performance is output as the selected features. Unfortunately, a known issue of wrapper methods is that the search space for d features is 2^d, which makes the exhaustive search impractical when dis very large.[17,18]

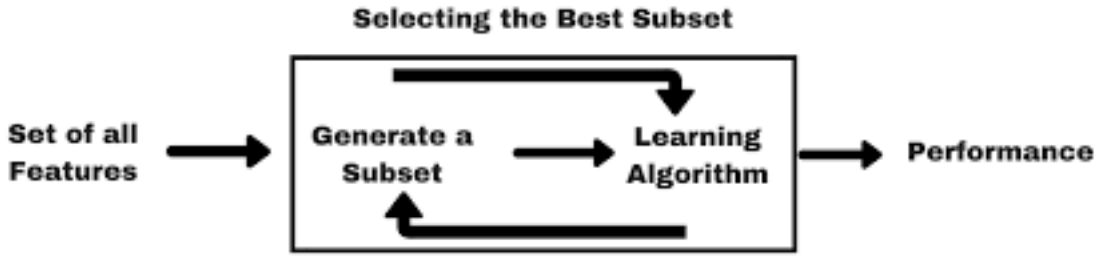

Figure 1.5: General Frame work of Wrapper Type Feature Selection

1.3.3 Embedded Method

In the Filter methods select features are independent of any learning algorithms are selected and are thus computationally efficient. But, they fail to consider the bias of the learning algorithm. This may make the selected feature not so optimal for specific learning tasks. This is which the utilize of the wrapper methods comes in. This method evaluates the importance of feature by given learning algorithm iteratively running in better accuracy in prediction. Now, the computational complexity increase in an exponential search space for high dimension feature. Embedded methods provides an optimal solution between filter and wrapper methods by embeding feature selection with the model learning. As this Embedded method is a combination of the two methods, it retains the qualities of both this wrapper and filter methods, which are – (1) they include the interactions with the learning algorithm; and (2) they do not need to even evaluate feature sets iteratively. The regularization models of the Embedded method target to fit a learning model by minimizing the fitting errors and forcing the feature coefficients to be small . Afterwards,the selected relevant feature sets are the resulting output .[16,17]

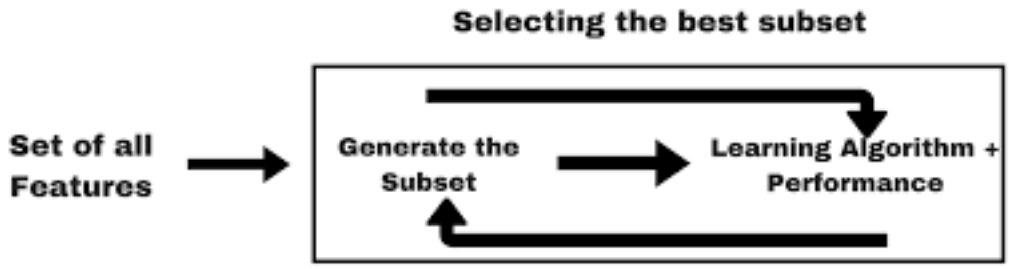

Figure 1.6: General Frame work of Wrapper Type Feature Selection

1.4 Data Perspective Feature Selection

The recent popularity of big data presents some challenges for traditional feature selection task. Meanwhile, some characteristics of big data like velocity and variety also promote the development of novel feature selection algorithms. Here we briefly present and discuss some major concerns when we apply feature selection algorithms.

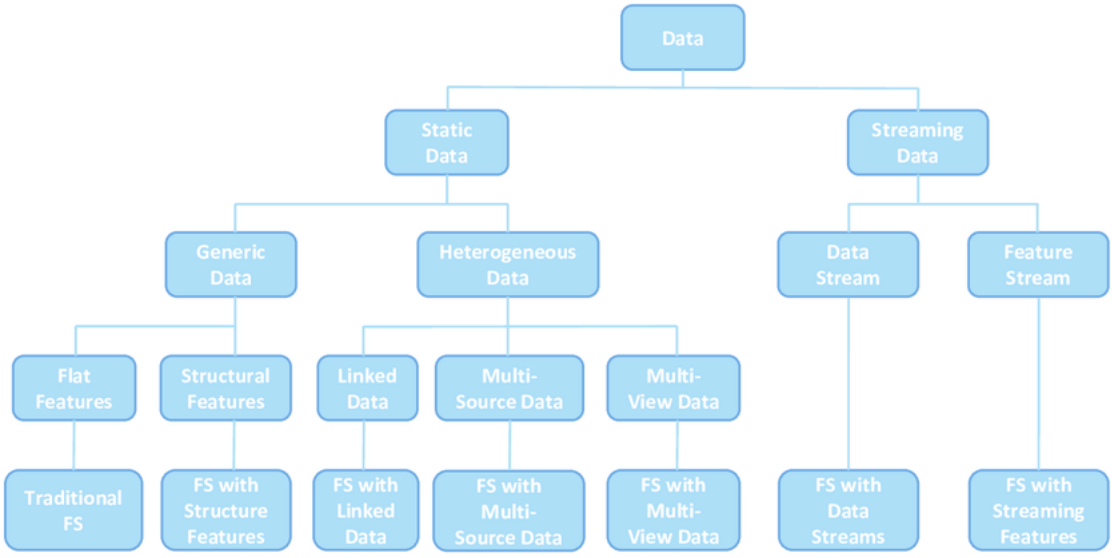

Figure 1.7: Data Perspective Feature Selection

1.4.1 Streaming Data and Features

Streaming data and features have become more and more prevalent in real-world applications. It poses challenges to traditional feature selection algorithms, which are designed for static datasets with fixed number of features. For example in Twitter, new data like posts and new features like slang words are continuously being user-generated. It is impractical to apply traditional batch-mode feature selection algorithms to find relevant features at each round when new data or new feature arrives. In addition, the volume of data could be too large to be loaded into memory. And in many cases, a single scan of the data is desired.

1.4.2 Heterogeneous Data

Most existing feature selection algorithms are designed to handle tasks with single data source and they always assume that data is independent and identically distributed (i.i.d.). However, multi-source data is quite prevalent in many domains. For example, in social media, data comes from heterogeneous sources such as text, images, tags. In addition, linked data is ubiquitous and presents itself in various forms such as user-post relations and user-user relations.

Chapter 2

Feature Selection Technique

Introduction

In this chapter define a pathway of feature selection and can tell about some existing machine learning algorithm which can find the feature importance. In previous chapter can give brief overview of different feature selection methods and here I can discuss how to solve a complex problem easily using feature selection approach. I can also through some light on feature extraction approach and how its differ from feature selection or variable selection.Here mostly discuss search perspective of feature selection methods in detail.

2.1 Development of Feature Selection

The process of selecting a subset of relevant and informative features from the original set of features can be divided into five main stages as shown in the below Figure 2.1.The decision made at each stage influences the feature selection performance.

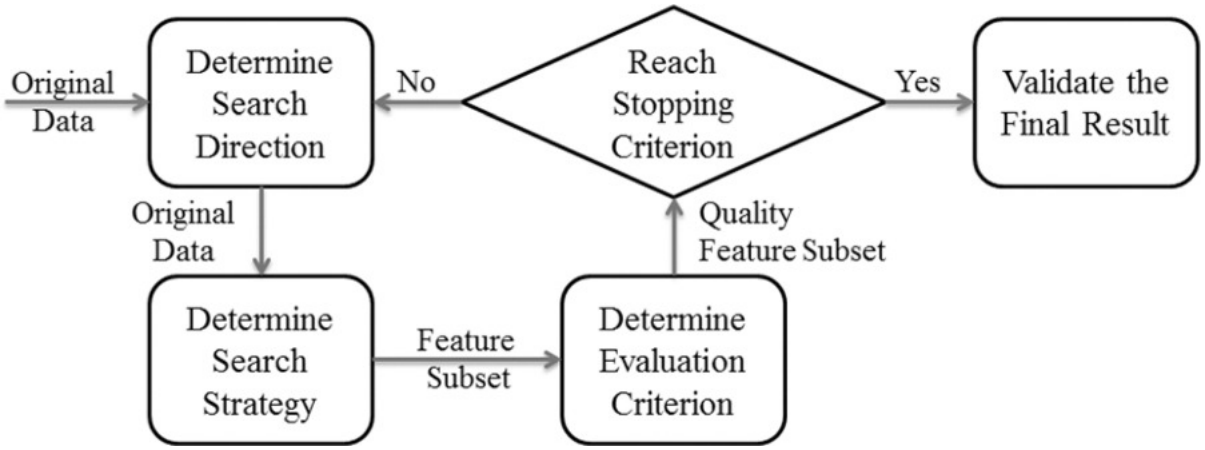

Figure 2.1: Stages of Feature Selection

Stage 1: Determine search direction. The first stage is to determine the starting point and the search direction. Search may start with an empty set and successively adds new features in each iteration, called forward search. In contrast,the search can be started with a full set and then the features are eliminated consecutively in each iteration, called backward elimination search. Another alternative is to begin with both ends by simultaneously adding and removing the features in each iteration, called bi-directional search.Search may also begin somewhere in the middle by randomly selecting the features to form the subset.

Stage 2: Determine search strategy. A good search strategy should provide good global search capability, rapid convergence to near optimal solution, good local search ability, and high computational efficiency[19]. Search strategies can be categorized into three groups: exponential, sequential, and randomized.

Exponential search, also called complete search, is the most exhaustive global search strategy. It starts from the original feature set and guarantees to find the optimal result.

However, this strategy is generally impractical and computationally intensive especially for high dimensional data sets, and prohibitive and intractable for all but a small initial number of features. An example of this strategy is exhaustive search, a search that evaluates all possible subsets to find the optimal subset[20].

Sequential search, also called greedy hill-climbing search[21],adds or removes one feature at a time. The most common sequential strategies are sequential forward selection (SFS) and sequential backward selection (SBS). It is relatively simple to implement, its complexity is polynomial with respect to the number of features, and it is robust to multi collinearity problems.

Randomized search strategy starts by randomly selecting the features and then proceeds with two different search strategies. The first uses the classical sequential or bidirectional search.

Stage 3: Determine evaluation criterion. Originally evaluation methods of feature selection are classified into three types: filter, wrapper and embedded. In recent years, another kind of evaluation method is developed, called ensemble feature selection. In the previous chapter we can briefly discuss this type of methods[16].

Stage 4: Define stopping criteria. A stopping criterion determines when the feature selection process should halt. A suitable stopping criterion can avoid over-fitting and thus leads to a more efficient process in producing an optimal feature subset with lower computational complexity. The decisions made in the previous stages will influence the choice of stopping criterion. The common stopping criteria are:

- Predefined number of features
- Predefined number of iterations
- Percentage of improvement over two consecutive iteration steps
- Obtaining an optimal feature subset according to some evaluation function.

Stage 5: Validate the result. To evaluate the effectiveness of potential feature sets for classification and prediction, various error estimation or validation techniques have been proposed. The most common error estimation methods are cross validation (CV) and performance measurements based on confusion matrix.

Cross validation is the most common and popular validation method. In this method, the original data sets are split into two parts: training and testing sets. The training set is used to train the classifier, and then the test set is used for the final evaluation. CV has the advantage of producing an effectively unbiased error estimate.

This is the stages of feature selection process. In the next section some popular feature selection technique can describe.

2.2 Feature Selection Technique

2.2.1 Pearson Correlation

Pearson correlation or bivariate correlation is a measure of linear correlation between two variables X and Y.According to the Cauchy-Schwarz inequality it has a value between +1 and -1, where 1 is total positive linear correlation, 0 is no linear correlation, and -1 is total negative linear correlation[21].

$$r_{XY} = \frac{cov(X,Y)}{\sigma_X * \sigma_Y}$$

where

- cov(X,Y) is the covariance of X and Y

$$cov(X,Y) = \sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})$$

- σ_X is the standard deviation of X

$$\sigma_X = \sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}$$

- σ_Y is the standard deviation of Y

$$\sigma_Y = \sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}$$

- $\bar{X}$ and $\bar{Y}$ is the mean of X and Y respectively

$$\bar{X} = \sum_{i=1}^{n} X_i$$

$$\bar{Y} = \sum_{i=1}^{n} Y_i$$

According to correlation coefficient I can select feature.If correlation coefficient between two variable is 1, then this variables or features are same contribution on the prediction. So any one of the feature can be selected.

2.2.2 Mutual Information

Mutual information is a measure between two (possibly multi-dimensional) random variables X and Y, that quantifies the amount of information obtained about one random variable, through the other random variable. The mutual information is given by

$$I(X;Y) = \iint p(x,y) log \frac{p(x,y)}{p(x)p(y)}\, dx\, dy$$

where p(x,y) is the joint probability density function of X and Y, and where p(x) and p(y) are the marginal density functions. The mutual information determines how similar the joint distribution p(x,y) is to the products of the factored marginal distributions. If X and Y are completely unrelated (and therefore independent), then p(x,y) would equal p(x)p(y), and this integral would be zero. Feature can be selected by maximize the mutual information[22].

2.2.3 Fisher Score

Fisher Score is a supervised feature selection algorithm. Suppose the class labels of n samples$y = y1, y2, ..., yn$ come from c classes, Fisher Score selects the features such that the feature values of samples within the same class are small while the feature values of samples from different classes are large. The Fisher score of each feature f_i is evaluated as follows:

$$fisher_score(f_i) = \frac{\sum_{j=1}^{c} n_j(\mu_{i,j} - \mu_i)^2}{\sum_{j=1}^{c} n_j \sigma_{i,j}^2}$$

where n_j ,μ_i, $\mu_{i,j}$ and σ^2 i,j indicate the number of samples in class j, mean value of feature f_i, mean value of feature fi for samples in class j, variance value of feature fi for samples in class j, respectively.[12] Using fisher score feature ranking can computed.

2.2.4 Low variance

Low Variance is a simple feature selection algorithm which eliminates features whose variance are below some threshold. For example, for the features that have the same values on all instances, the variance is 0 and should be removed since they cannot help to discriminate instances from different classes. Suppose that the data set consists of only boolean features,i.e., the feature values are either 0 and 1. Since the boolean features are Bernoulli random variables, their variance values can be computed as:

$$variance - score(f_i) = p(1-p)$$

where p denotes the percentage of instances that take the feature value of 1. After obtaining the variance of features, the features with a variance score below a predefined threshold can be directly eliminated.[23]

2.2.5 T-Score

T-score is used for binary classification problems. For each feature f_i, suppose that μ_1 and μ_2 are the mean feature values for the instances from the first class and the second class respectively. σ_1 and σ_2 are the corresponding standard deviation values. n_1 and n_2 denote the number of instances from these two classes. Then the t-score for the feature f_i can be computed as:

$$t - score(f_i) = \frac{|\mu_1 - \mu_2|}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

The basic idea of t-score is to assess whether the feature can make the means of two classes to be different statistically by computing the ratio between the mean difference and the variance of two classes. Usually, the higher the t-score, the more important the feature is.[24]

2.2.6 Chi Square Score

Chi-square score utilizes the test of independence to assess whether the feature is independent of the class label. Given a particular feature with r different feature values, the Chi-square score of that feature can be computed as:

$$Chi - Squere - score(f_i) = \sum_{j=1}^{r}\sum_{s=1}^{c}\frac{(n_{js} - \mu_{js})^2}{\mu_{js}}$$

where n_{js} is the number of instances with the j-th feature value. In addition, $\mu_{js} = \frac{n_{*s}n_{j*}}{n}$,where n_{j*} indicates the number of data instances with the j-th feature value, n_{*s} denotes the number of data instances in class r. Normally, a higher Chi-square score indicates that the feature is relatively more important.[25]

2.2.7 Gini Index

Gini index is a statistical measure to quantify if the feature is able to separate instances from different classes. Given a feature f_i with r different feature values, for the j-th feature value, let W denote the set of instances with the feature value smaller than or equal to the j-th feature value, let $\hat{W}$ denote the set of instances with the feature value larger than the j-th feature value. In other words, the j-th feature value can separate the dataset into W and $\hat{W}$, then the Gini index score for the feature f_i is given as follows:

$$Gini{-}Index{-}Score(f_i) = min(p(W)(1{-}\sum_{s=1}^{c}p(C_s|W)^2){+}p(\hat{W})(1{-}\sum_{s=1}^{c}p(C_s|\hat{W})^2))$$

where Cs indicates that the class label is s. p(.) denotes the probability, for instance, $p(C_s|W)$ indicates the conditional probability of class s given the set of W. In previous equation the gini index score is obtained by going through all the possible split W. Usually for binary classification problem, it can take a maximum value of 0.5, but it can also be used in multi-classification problems. Unlike previous mentioned statistical measures, the lower the gini index value, the more relevant the feature is.[26]

Chapter 3

Feature Selection with ANN

In this section I can first discuss about artificial neural network then I can proposed my methods for feature selection with artificial neural network.Two methods have been proposed for feature selection with ANN.

3.1 Artificial Neural Network

This is the most simplest architecture in deep neural network. Artificial neural network(ANN) is mostly used for solving classification and regression problem. We can used ANN as a classifier and we can also find out the feature importance of this classifier or selecting relevant feature or variable subset. Here we can discuss about the ANN.

3.1.1 Introduction

Neural networks were first created in an attempt to model the human brain. However, throughout the years, they have become much different of what they used to be and derived from their initial purpose. They are now used as powerful machine learning tools in a multitude of fields: computer vision, stock market analysis and robotics being some of them. This subsection aims at giving a brief introduction on their internal working such that the rest of the document can be understood.

In the Figure 3.1 given the basic and most simplest architecture of Artificial Neural Network. There are n number of input and m number of hidden layer.In this neural network number of input node is n and m number of

hidden layers node such that $n > m$.

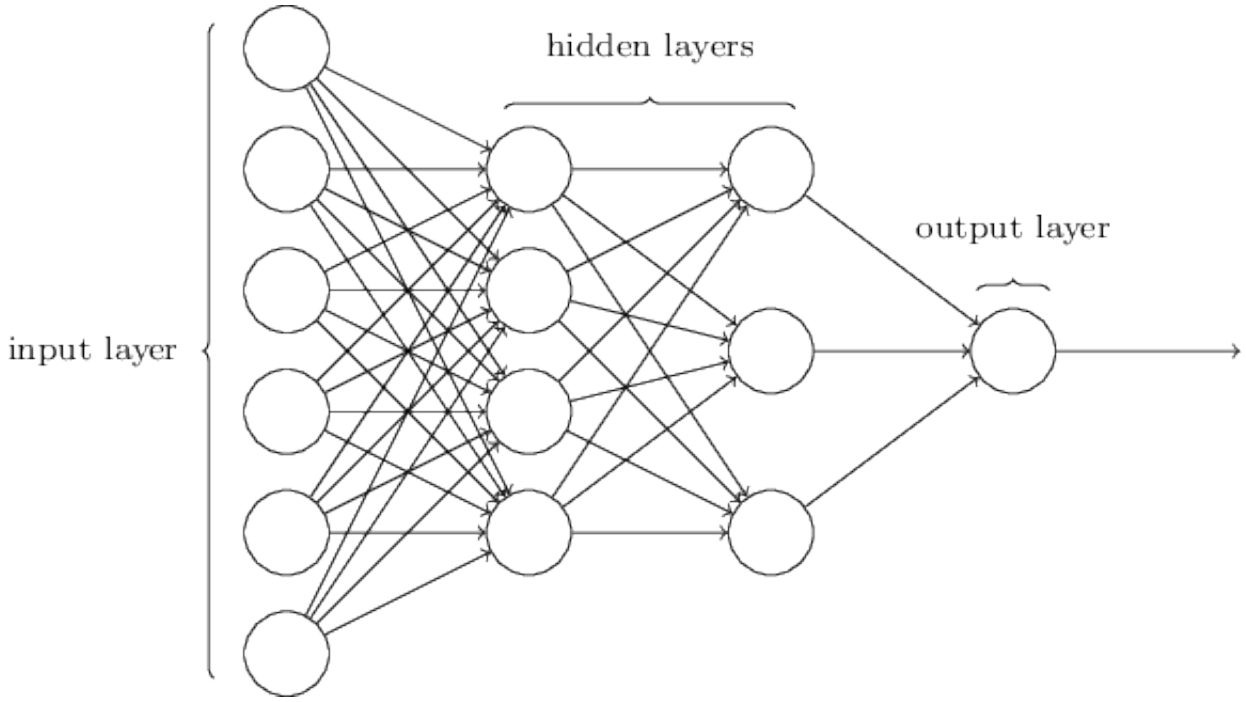

Figure 3.1: Neural Network

3.1.2 Basic Principle of Neural Network

Each neuron consists of an activation function which takes the weighted sum of the previous layer's outputs as input as well as a bias. Let us define the following:

- n_1,...., n_k the neurons of the layer i.
- m_1,....,m_l the neurons of layer j where $i > j$
- f the activation function of the neurons.
- w_{ij} the weight of the link between output of neuron n_i and neuron m_j .
- Bias(n_i) the bias of neuron i.
- Out(n_i) the output of neuron i.

Thus we have:

$$out(m_i) = f(\sum_{x=1}^{k} Out(n_i) * w_{zi} + Bias(m_i))$$

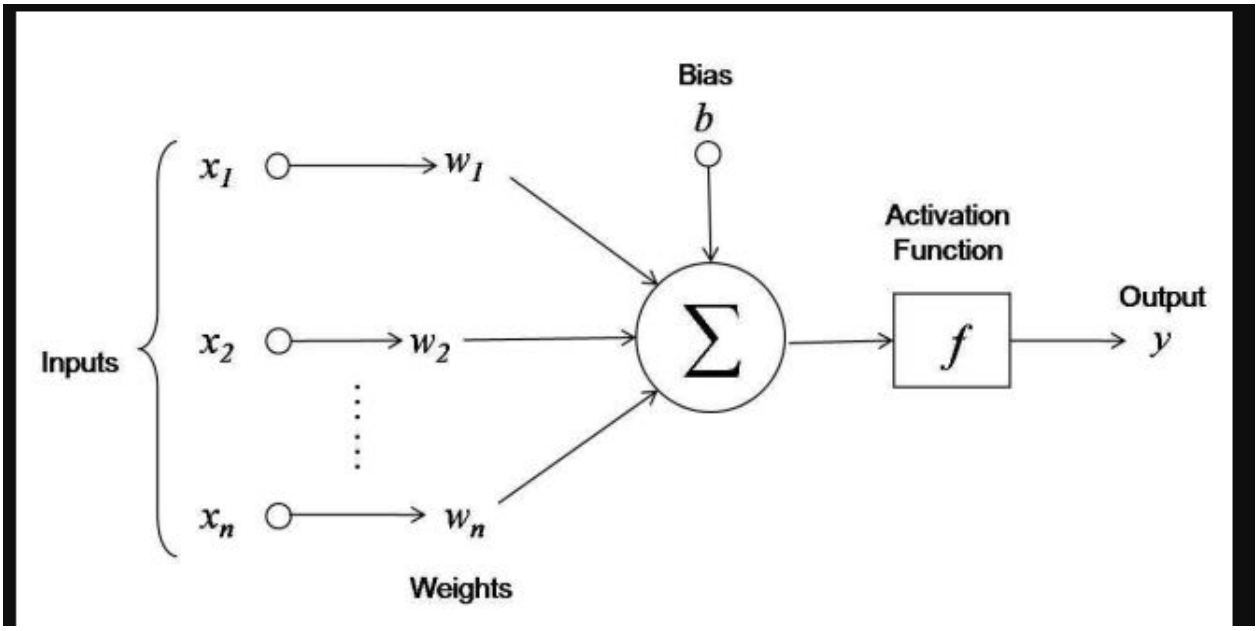

Figure 3.2: Neural Network Definition

It is now quite easy to see how to get a prediction. One simply needs to feed an input to the network and propagate it thanks to previous Equation in order to get the final output.This is called a feed forward propagation, since an input is passed from shallower layers to deeper ones. Now one needs to understand how to train the network. Let us denote by y_f the output of the neural network computed with a feed forward pass and by output the ground truth.Now denote the loss function $l(y_t; y_f)$. The mean square error and cross-entropy are often used as loss function respectively in regression and classification settings. The weights and biases of the networks are updated in order to minimise the loss function thanks to stochastic gradient descent.

For example, to update the weight w_{ijk} one computes$\frac{d_l}{d_{w_{ijk}}}$ and makes the following update:

$$w_{ijk} \leftarrow w_{ijk} - \delta * \frac{d_l}{d_{w_{ijk}}}$$

with $\delta \in [0, 1]$.Note that those derivatives can be computed efficiently using back propagation. Indeed, due to neural networks architecture, once the$\frac{d_l}{d_{w_{ijk}}}$ have been computed they can be used to get the $\frac{d_l}{d_{w_{i-1jk}}}$ without carrying too much additional computation. Thus in order to make a training iteration, one has to make a feed forward pass through the network in order to compute the output of each neuron and get y_f. Once done, each$\frac{d_l}{d_{w_{ijk}}}$ are computed layer by layer (from deeper ones to shallower ones) while the network's weights are updated accordingly. In practice, the inputs are fed by batches of samples to the network. Therefore, a training iteration consists of computing all the weights update (one per sample) and adding them all together before updating the network's parameters. Batches are usually taken at random from the data set (while making sure that each data sample appears equally) such that the network sees different batches most of the time. Finally, note that an epoch represents the number of training iterations to be done such that

nbr training iterations * batch size = nbr training samples

In other words, an epoch represents the number of training iterations required for the network to have seen each training sample at least once. Usually,artificial neural networks have to be trained on many epochs in order to get good results.[22]

3.1.3 Feature Selection with ANN

Neural network can be built in a plentitude of ways and are subject to many parameters, neural architecture being the first one. Indeed, neural networks can take many forms, ranging from very shallow to very deep and very narrow to very wide. Many constraints can also be added in the architecture itself, convolutional and encoder layers are some of them. All of these parameters can be changed regarding the problem we are facing. In our case we decided to limit ourselves to test our algorithms on networks with fully connected hidden layers. We did this choice since our data didn't give us a priori reasons to introduce structure into our network. Furthermore, this is the more generic and simplest architecture that can be found. However, all the algorithms that will be proposed in next section using the artificial neural network structure.

Another important parameter is the activation function, it was considered for a long time that the best activation functions were either sigmoids or hyperbolic tangent. However, it was recently shown empirically that Rectified Linear Units(ReLU)[27] functions were very effective [Nair and Hinton, 2010] (note that ReLU(x) = max(0; x)). Since then, pretty much all neural networks have been built with ReLU neurons, which has proven to be extremely effective.Thus I decided to carry out the tests with such neural networks. In the final hidden layer to output layer I can use sigmoid activation function for great result.

I have to choose which training algorithm to use. All of them are based on the gradient descent principle and use back-propagation [28] in order to be computationally efficient. ADAM is a recent algorithm that has been proposed in [29] and which revolves around modifying the gradient descent in order to include momentum. The momentum introduced implies that the gradient descent has a sort of short memory, i.e. while optimising the weights on a given data batch it also uses the derivatives computed on some of the previous batches. Due to its good performances, I decided to train all of my networks with the ADAM optimiser.

3.1.4 Motivation and Introduction to our Methods

The main motivation of proposed methods was the neural network working principal. Neural network is a very good classifier among all of the classifier like Random Forest, Desision Tree etc. When I can train the neural network firstly I can randomize the weights of each input node to hidden layers node. After this forward propagation through the hidden layers it can compute some output. This computed output can be compaired with the actual output and then network can adjust the weights of the network. This weight adjustment occur in backward direction so this is called the Back propagation.

This step of forward propagation and back propagation can compute for each input vector of the training set. After training of the neural network, weights of each connected layers are computed accurately. If we can validate our test set then this weights plays the main role for the prediction. So this weights matrix give the main significance of our prediction.

Now I can think about how to manipulate or perform some statistical operation to find out which node of the input node have the most contribution about the prediction. In proposed methods, the prime focus on the first weight matrix or rather say weight matrix between input node and first hidden layer.In the proposed method I can perform some relevant statistical operation to find out important feature of the input vector. Eliminating some irrelevant feature of the input vector building ANN model for this relevant feature subset. After creating the prediction model checking accuracy of the prediction using confusion matrix and ROC curve.Confusion matrix and ROC curve are the best way to validate the prediction model.

In this section the proposed methods and experiment result of each methods. Methods have been limited to "Supervised Classification Problem".In the experiment we use "Keras" API backend with tensorflow has been used in the experiment.In every method we can first build an Artificial neural network has to build first and train the network using all the features in the data set.After training weight matrix between input layer and first hidden layer has to be extracted.

3.2 First Method

In the previous section the main motivation of the weight matrix of the neural network.Let there are n number of inputs, so we can say that the ANN has n input node and there are m number of nodes in the first hidden layer $m <= n/2$.Each input node is connected to each output node in such a manner that every input have m number of weights. The weights matrix between input layer and first hidden layer be:

$$\begin{bmatrix} w_{11} & w_{12} & w_{13} & \dots & w_{1n} \\ w_{21} & w_{22} & w_{23} & \dots & w_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ w_{m1} & w_{m2} & w_{m3} & \dots & w_{mn} \end{bmatrix}$$

Here in the weights matrix row represent the number of input node and column represent the number of node of the first hidden layer. For each feature there are m number of weights and there are n number of feature. For each feature there are m number value, now this can be define as:

$$w_{1i}, w_{2i}, w_{3i}, ..., w_{(m-1)i}, w_{mi}$$

For each feature with the sample value the variance or variability of each feature can be computed as: variance of i^{th} feature can be define as:

$$\sigma_j^2 = \frac{1}{m}\sum_{i=1}^{m}(w_i - \mu)^2$$

where

$$\mu = \frac{1}{m}\sum_{i=1}^{m} w_i$$

The variance of each feature is calculated as $\sigma_1^2, \sigma_2^2, ...\sigma_j^2, ..., \sigma_n^2$. The average variance is found out as:

$$V = \frac{1}{n}\sum_{i=1}^{n}\sigma_j^2$$

This average variance can be calculated because I do not know which variable or feature may be over fit the prediction.Then is found out the difference between each feature variance and the average variance is calculated as:

$$D_j = \sigma_j^2 - V$$

All the difference value be $D_1, D_2, D_3, ...D_n$.This difference set of value can be calculated as the maximum and minimum difference value. The feature having maximum difference value is called the most irrelevant then other feature and which feature have less difference value is the more relevant than the other features.

80 percent of the relevant feature can be selected according to the difference criteria mention above.Also I can validate our prediction model and prove that the rule of finding the relevant and irrelevant feature is correct.

3.3 Second Method

Let there are n number of inputs, so the ANN has n input node and let there are m number of nodes in the first hidden layer $m <= n/2$.Each input node is connected to each output node in such a manner that every input have m number of weights.

The weights matrix between input layer and first hidden layer be:

$$\begin{bmatrix} w_{11} & w_{12} & w_{13} & \dots & w_{1n} \\ w_{21} & w_{22} & w_{23} & \dots & w_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ w_{m1} & w_{m2} & w_{m3} & \dots & w_{mn} \end{bmatrix}$$

In this method we introduce a concept of the signal.Let us suppose neural network as a signal generator and there are n number of discrete signal with each have m sample value. So each column of the weight matrix represent a discrete signal. The i_{th} signal sample value can be represent as: $w_{1i}, w_{2i}, w_{3i}, ..., w_{mi}$.But I do not know that which signals are relevant and which are not. There may a chance of the presences of a noise signal which can be very crutial to the prediction.

Now we calculate power of each signal as:

$$P_i = \frac{1}{2m+1} \sum_{j=1}^{m} w_{ji}^2$$

We do not predict the power of a noise signal it may be high or less value. So for this case we can calculate the average power and we can also calculate difference between average power and the actual power of each signal. Average power can be calculated as:

$$P_{av} = \frac{1}{n} \sum_{i=1}^{n} P_i$$

Difference between average power and signal power as:

$$D_i = |P_i - P_{av}|$$

The maximum and minimum difference value can be found.Selecting relevant signal whose difference value i.e D_i is minimum. 70 to 80 percent of the signal can be selected for creating a accurate prediction model. I can also validate the prediction model with some accuracy measurement technique.

3.4 Experiment and Result

This section about how the prediction model can be validated. Using this technique proposed model can be validated.

3.4.1 Method validation Technique

Let us start by introducing confusion matrices which are often built in order to assess the performances of classification models. Consider a binary classification problem with a class corresponding to a positive outcome (for example an alarm activation) and the other to a negative outcome (the alarm doesn't activate). Also consider a binary classification model which is used to classify an input (for example a motion detector) to one of the two classes. For each data sample, the classification made by the model belongs to one of the following categories:

True Positive(TP): This occur if the model activates the alarm when it should have been.No error is made.

True Negative(TN): This occurs if the model doesn't activate the alarm rightfully so (i.e. the alarm should not have been activated). No error is made.

False Positive(FP): This occurs when the model activates the alarm although it should not have been. An error is made and leads to Type 1 error.

False Negative(FN): This occurs when the model doesn't activate the alarm although it should have been. An error is made and leads to Type 2 error.

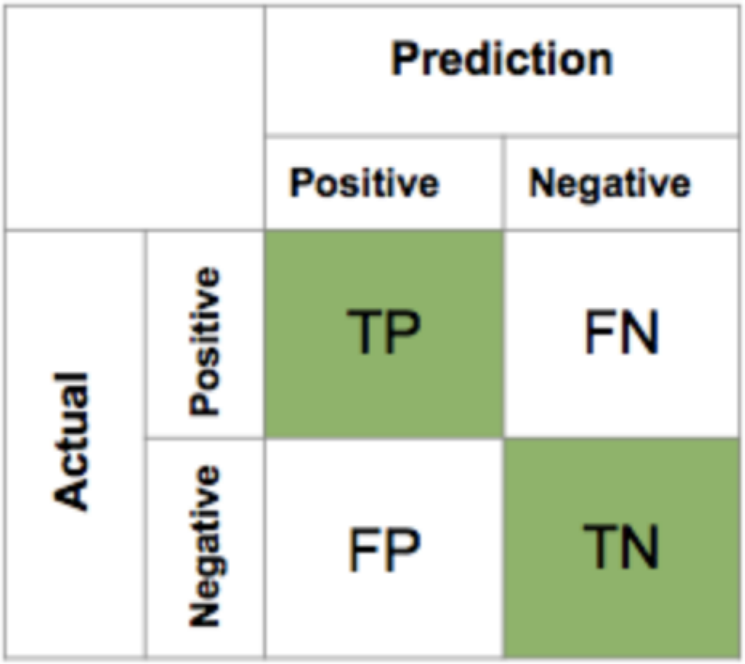

Figure 3.3: Confusion Matrix

From the confusion matrix two measure can be introduce:

Sensitivity : Which is defined as $\frac{TP}{TP+FN}$ and represents the true positive rate among all classified positives.

Specitivity : Which is defined as $\frac{FP}{FP+TN}$ and represents the false positive rate.

These measure can also use in order to assess feature selection. Using true positive rate and false positive rate we can plot the ROC curve.An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds.An ROC curve plots TPR vs. FPR at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives. The following figure shows a typical ROC curve.

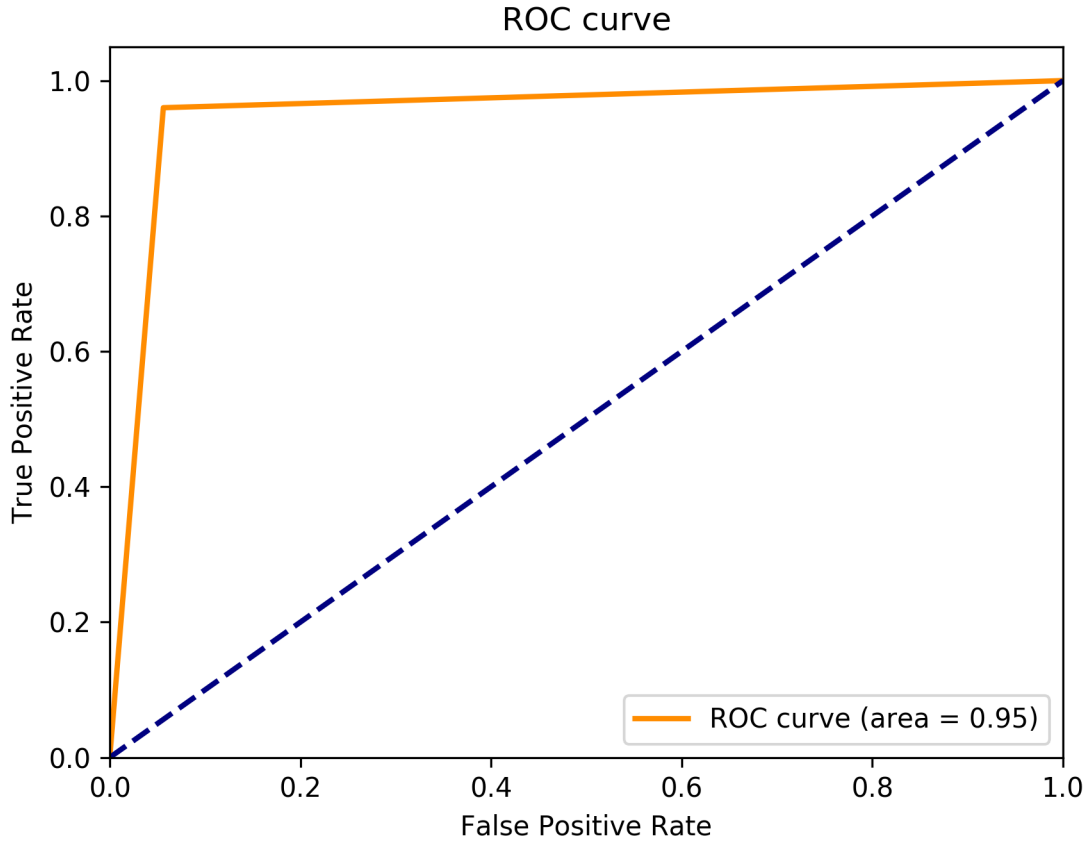

Figure 3.4: Receiver Operating Characteristic Curve

3.4.2 Results

Sevaral data sets used in validation for proposed methods. Experiment using one data set is given below section.

3.4.3 Description of Dataset

This data set was collected from Kaggle a machine learning repository. The goal data set refers to the presence of heart disease in the patient.This data set contains 13 attributes and one target value. The target value is 0 consider

as the patient do not have the heart disease and 1 means patient have the heart disease. This a binary classification problem which can be solved using proposed methods. There are total 330 patient data available and for the training set considering 80 percent of samples, remaining for the test set.Test set contain only 61 sample.Using this test the predictive model was validated using confusion matrix and ROC curve.

3.4.4 Correlation

Correlation measurement is important aspect to find out the relationship between the variables or features. If correlation coefficient between the two variable is 1 then they are strongly correlated and for this case any one variable can be selected for creating predictive model. The correlation matrix is given the Figure 3.5.

According to this correlation matrix there are no two variables are perfectly correlated so all of the variable can be used for creating a predictive model.

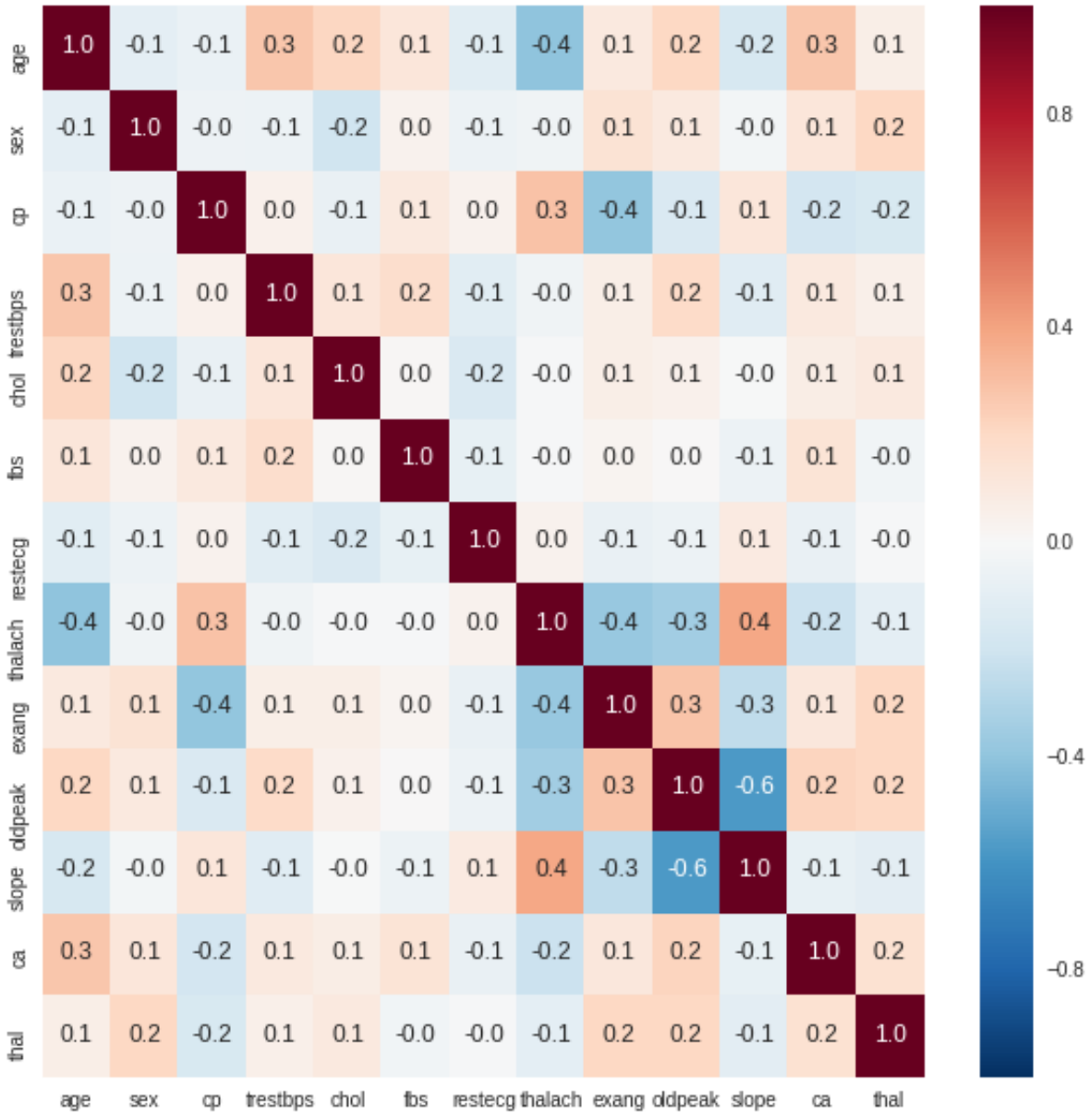

Figure 3.5: Correlation coefficient Matrix

3.4.5 ANN Predictive Model

I can create very basic Artificial Neural Network for the prediction.In this Artificial Neural Network model there are 13 feature so 13 input node. In this ANN contains two hidden layer each having 7 node or neuron and only one output node for the binary value. After training of ANN the validation results are given below:

Figure 3.6: Confusion Matrix for ANN

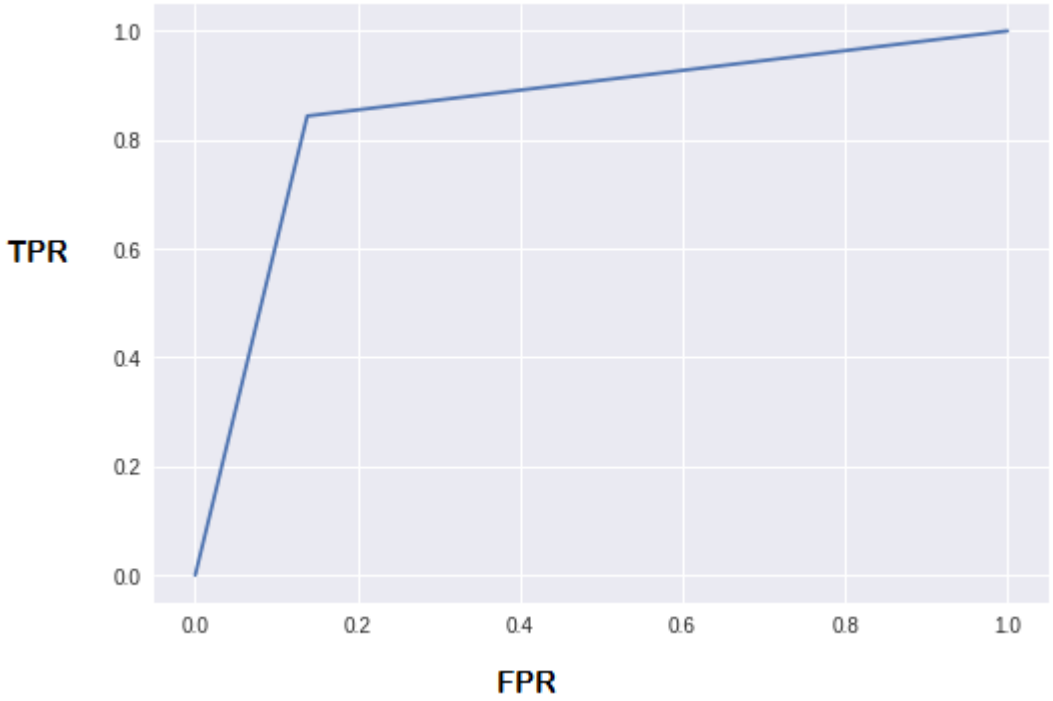

Figure 3.7: ROC curve for ANN Model

In the ANN model prediction accuracy was 85 percent.After this classification the weight matrix between input layer and first hidden can extracted. In the Figure 3.8 weight matrix is given.

Weights between input and First Hidden layer

FEATURE	0	1	2	3	4	5	6
0	-0.412744	-0.230654	-0.133827	-0.232592	-0.062326	-0.096632	-0.018187
1	-0.287821	0.287242	-0.054948	-0.356109	-0.226468	0.046953	-0.299066
2	0.100102	-0.662339	-0.395202	0.000481	0.292894	-0.144885	0.214301
3	0.186916	0.001841	0.382688	0.221907	0.010622	0.247990	-0.081417
4	-0.374905	-0.115232	-0.164330	0.139116	-0.349946	0.026578	0.020544
5	0.157711	-0.084290	-0.168186	0.014477	-0.007257	-0.012069	-0.080298
6	0.298616	-0.479295	-0.132443	0.004819	0.133658	0.136968	-0.114123
7	0.315318	0.166221	-0.141871	0.239020	0.116120	-0.298401	0.052109
8	0.000458	0.470245	0.324466	-0.340370	0.186809	0.309757	0.229391
9	0.107907	0.434853	0.272773	-0.000192	-0.049244	0.540827	-0.032553
10	0.212619	-0.071235	-0.283980	-0.074712	0.214076	-0.411884	0.082679
11	-0.398046	0.122723	0.206226	-0.629236	-0.424998	0.177335	-0.710310
12	-0.166564	-0.006589	0.451380	-0.231749	-0.057518	0.177866	-0.233472

Figure 3.8: Weight Matrix of ANN

3.4.6 First Proposed Method

Here the validation results of the first method is given. Using this weight matrix variance of each feature and difference from the average variance calculated. The table below shows the results of the calculation.

Feature	Variance	Difference from average Variance
0	0.01547	0.035998
1	0.046415	0.004991
2	0.101012	0.049607
3	0.023473	0.027993
4	0.032593	0.018812
5	0.008928	0.042477
6	0.054688	0.003283
7	0.040117	0.011288
8	0.060937	0.009532
9	0.048446	0.002959
10	0.049005	0.0024
11	0.133810	0.082405
12	0.053437	0.002032

According to "Difference from Average Variance" Score 1,2,5 and 11 number features or variables had the maximum value. So without using this feature the next ANN model can created and the prediction result of the ANN given below.

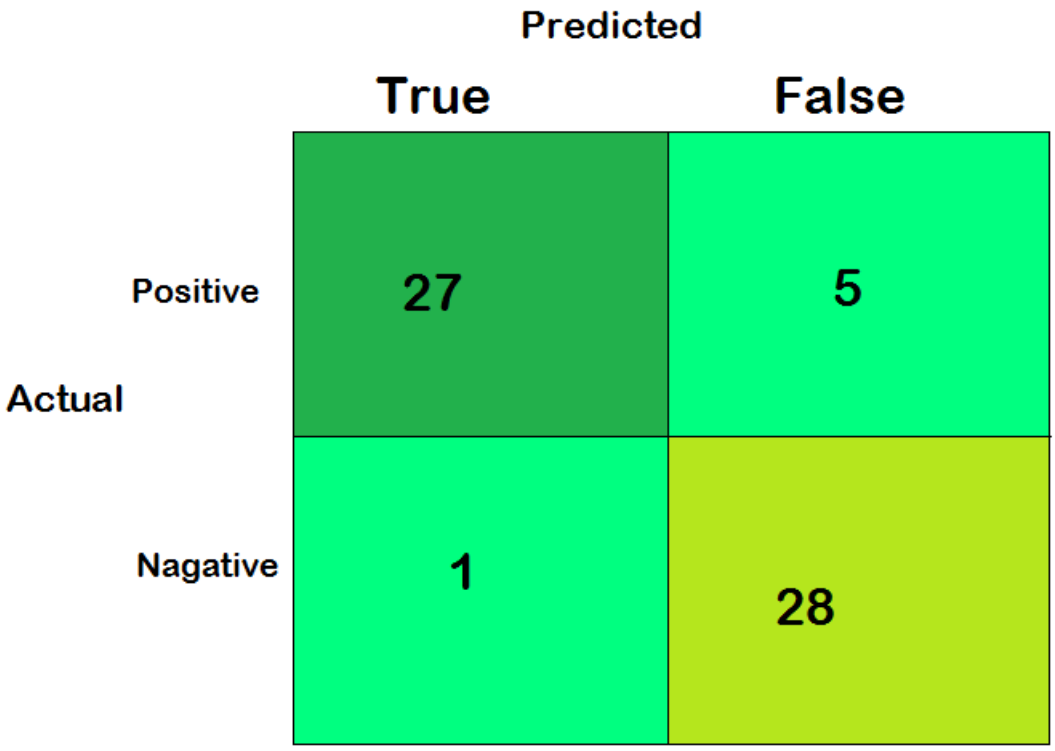

Figure 3.9: Confusion Matrix of ANN for First method

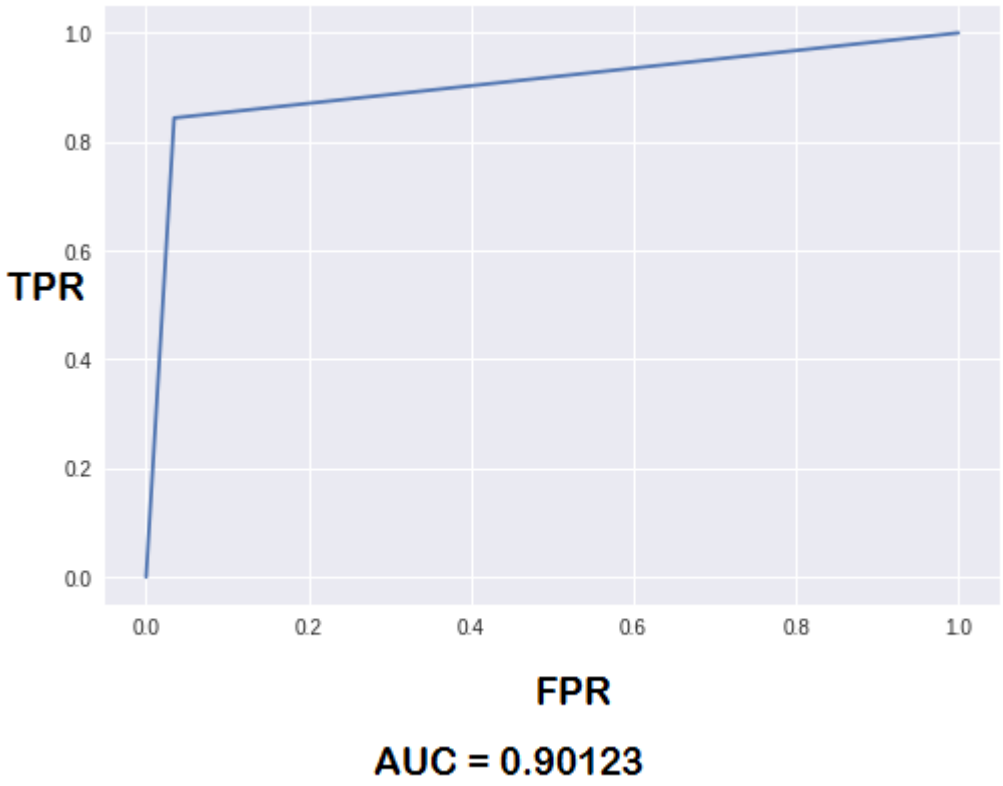

Figure 3.10: ROC Curve of ANN for First method

3.4.7 Second Method

Here the validation results of the second method is given. Using this weight matrix the signal power of each feature and difference from the average power calculated. The table below shows the results of the calculation.

Feature	Signal Power	Difference from average power
0	0.020608	0.026251
1	0.029207	0.017652
2	0.050506	0.003647
3	0.024910	0.021949
4	0.021585	0.025274
5	0.004474	0.042385
6	0.025740	0.021119
7	0.020637	0.026222
8	0.041715	0.005144
9	0.038036	0.008823
10	0.023921	0.022938
11	0.088584	0.041725
12	0.024979	0.021880

According to "Difference from Average power" Score 0,5,7 and 11 number features or variables had the maximum value. So without using this feature the next ANN model can created and the prediction result of the ANN given below.

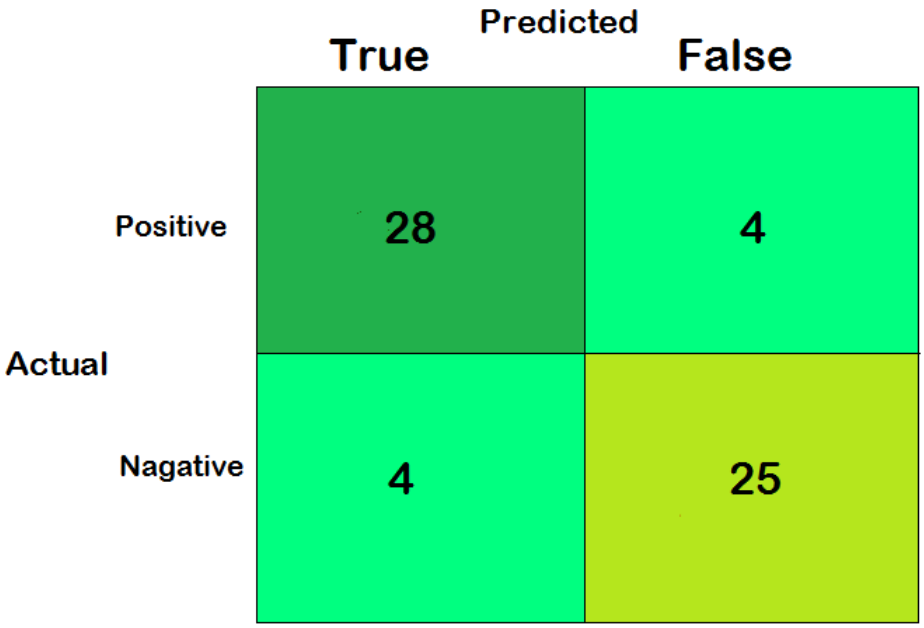

Figure 3.11: Confusion Matrix of ANN for Second method

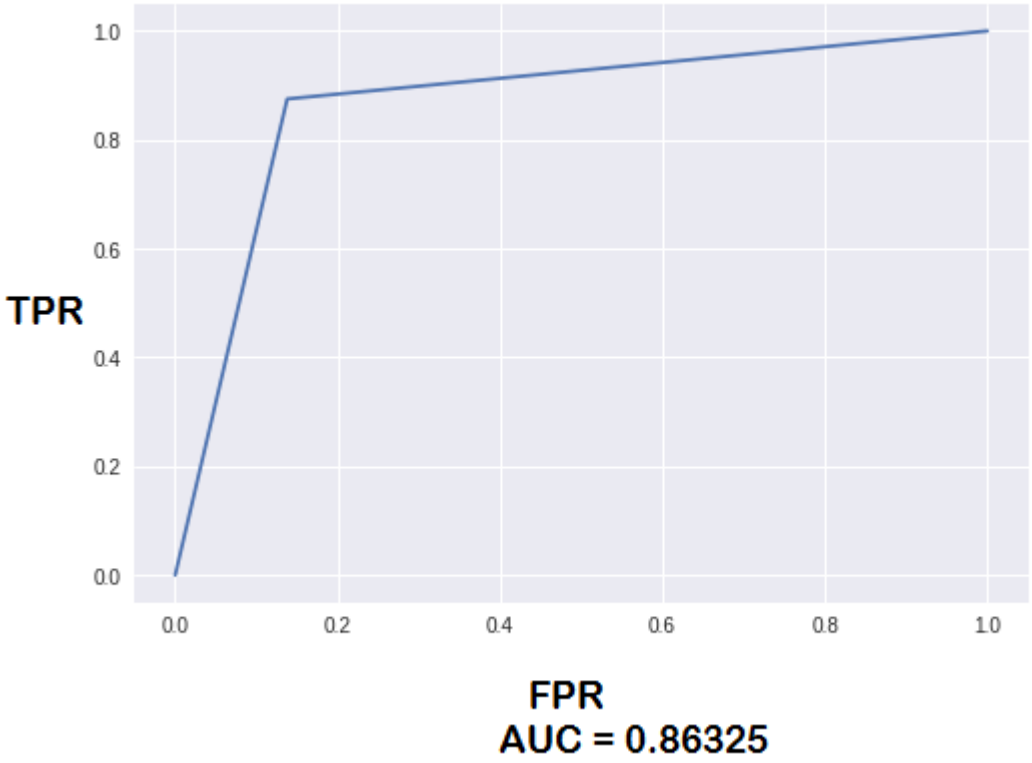

Figure 3.12: ROC Curve of ANN for Second method

Chapter 4

Conclusion and Future Work

4.1 Conclusion

Feature selection is effective in preprocessing data and reducing data dimensionality which is essential to successful data mining and machine learning applications. Meanwhile, it has been a hot research topic with practical significances in many areas such as statistics, pattern recognition,machine learning, and data mining (including web, text, image, and microarrays). The objectives of feature selection include: building simpler and more comprehensible models, improving data mining performance and helping prepare clean and understand data.

The main objective was to determine relevant feature by emploing the weight matrix of the Artificial Neural Network. After classification using ANN the weight matrices gave main idea about the variables or features which were used to create the prediction model. In the proposed methods focus was to find out the exact pattern for weights matrix. Using this pattern relevant and irrelevant feature can be easily found and using this relevant features thus found an efficient prediction model can be build which is giving batter accuracy than an normal ANN model. The first proposed method can achieve at least 7 percent more accurate prediction than ANN. The second method can achieve only 2 percent more accuracy than normal ANN model.

Feature selection with ANN is mainly focused because ANN is very efficient classifier then any other tree classifier. Using the proposed methods understanding real world problem become easy or easily understand the relevant variables for any particular problem without having great knowledge on that problem.This is the main outcome of proposed methods.

4.2 Future Work

Feature selection with ANN can be more focus on how to find the exact pattern from the weights matrix.For the large data set where number of features more than 200,here first dimensionality reduction technique can be applied before feature selection technique.Some relevant statistical can also be applied in weight matrix to find out the relevant feature or feature importance.

Feature selection technique can also applicable in different types of neural network such as Recurrent neural network(RNN), Convolution neural network(CNN) etc.

Reference

- [1] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. Statistical Learning with Spar- sity: The Lasso and Generalizations. CRC Press, 2015.
- [2] Ian Jolliffe. Principal component analysis. Wiley Online Library, 2002.
- [3] Bernhard Scholkopft and Klaus-Robert Mullert. Fisher discriminant analysis with kernels. Neural networks for signal processing IX, 1:1, 1999.
- [4] David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. Neural computation, 16(12):2639– 2664, 2004.
- [5] Gene H Golub and Charles F Van Loan. Matrix computations, volume 3. JHU Press, 2012.
- [6] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. Science, 290(5500):2319–2323, 2000.
- [7] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. Science, 290(5500):2323–2326, 2000.
- [8] Robert Tibshirani. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological), pages 267–288, 1996.
- [9] Thomas M Cover and Joy A Thomas. Elements of information theory. John Wiley and Sons, 2012.
- [10] Kenji Kira and Larry A Rendell. The feature selection problem: Traditional methods and a new algorithm. In AAAI, volume 2, pages 129–134, 1992a.

- [11] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 27(8):1226–1238, 2005.
- [12] Richard O Duda, Peter E Hart, and David G Stork. Pattern classification. John Wiley and Sons, 2012.
- [13] Xiaofei He, Deng Cai, and Partha Niyogi. Laplacian score for feature selection. In Advances in neural information processing systems, pages 507–514, 2005.
- [14] Zheng Zhao and Huan Liu. Spectral feature selection for supervised and unsupervised learning. In Proceedings of the 24th international conference on Machine learning, pages 1151–1157. ACM, 2007.
- [15] Kenji Kira and Larry A Rendell. A practical approach to feature selection. In Proceedings of the ninth international workshop on Machine learning, pages 249–256, 1992b.
- [16] M. Dash and H. Liu, "Feature selection for classification," Intell. Data Anal., vol. 1, no. 3, pp. 131–156, Jan. 1997.
- [17]Isabelle Guyon and Andr'e Elisseeff. An introduction to variable and feature selection. The Journal of Machine Learning Research, 3:1157–1182, 2003.
- [18]Y. Saeys, "Feature selection for classification of nucleic acid sequences," Ph.D. dissertation, Ghent Univ., Ghent, Belgium, 2004.
- [19] I. A. Gheyas and L. S. Smith, "Feature subset selection in large dimensionality domains," Pattern Recognit., vol. 43, no. 1, pp. 5–13, Jan. 2010.
- [20] P. A. Devijver and J. Kittler, Pattern Recognition: A Statistical Approach. Englewood Cliffs, NJ, USA: Prentice-Hall, 1982.
- [21]Ron Kohavi and George H John. Wrappers for feature subset selection. Artificial intelli- gence, 97(1):273–324, 1997.

- [21] Karl Pearson (20 June 1895) "Notes on regression and inheritance in the case of two parents," Proceedings of the Royal Society of London, 58 : 240–242.

- [22] Roberto Battiti. Using mutual information for selecting features in supervised neural net learning. Neural Networks, IEEE Transactions on, 5(4):537–550, 1994.

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

- [24] Sewall Wright. The interpretation of population structure by f-statistics with special regard to systems of mating. Evolution, pages 395–420, 1965.

- [25] Huan Liu and Rudy Setiono. Chi2: Feature selection and discretization of numeric attributes. In tai, page 388. IEEE, 1995.

- [26] Mark A Hall and Lloyd A Smith. Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper. In FLAIRS conference, volume 1999, pages 235–239, 1999.

- [27] Nair, V. and Hinton, G. E. (2010). Recti

  ed linear units improve restricted boltzmann machines. In Proceedings of the 27th international conference on machine learning (ICML-10), pages 807814.

- [28] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1988). Learning representations by back-propagating errors. Cognitive modeling, 5(3):1.

- [29] Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.