

# Investigation of Dimensionality Reduction on Classification accuracy

M.Janani<sup>1</sup>, S.Lovelyn Rose<sup>2</sup>

Department of Computer Science and Engineering, PSG College of Technology, Tamilnadu, India<sup>1,2</sup>

**Abstract:** This paper deals with the comparison of different dimensionality reduction techniques when combined with various classification techniques. The dimensionality reduction techniques considered are PCA, ICA, TSVD, LSI and RP. They are mainly used for feature extraction. Their main goal is to reduce noisy data, redundant data, memory/disk needed to store data. They prevent the problem of over-fitting and help to visualize high dimensional data. ANN, SVM, Naïve Bayes, K-NN, Random Forest are some of the commonly used supervised learning models for classification. Their main advantage is speed of training, predictive accuracy on new data and less memory usage. This paper uses PCA, ICA, and TSVD to attain dimensionality reduction and ANN, SVM, RF to attain classification accuracy on donorchoose.org dataset that tests if a project is A+ or not and the results show that ICA with RF gives the best accuracy.

**Keywords:** Principle Component Analysis, Independent Component Analysis, Truncated Singular Value Decomposition, Support vector Machine, Artificial Neural Network, Random forest.

## I. INTRODUCTION

Today's Machine learning system provides effective learning methodologies compared to traditional learning approach. Machines learn from the given input data rather than by following programming instruction. There are two major forms of machine learning techniques used namely, supervised learning and unsupervised learning. In the supervised learning model, machines learn using labelled training data and evaluate the accuracy of the model with test data. Some of the common techniques used in classification are neural networks, SVM, random forest, naive bayes. Unsupervised learning, have unlabelled training data and help to cluster data. For example consider a company having past customer records which contains customer information as well as past transactions of the customers. In such a case, clustering helps companies with natural grouping of its customers and classification helps to classify the customers. This helps to decide strategies, identify outliers etc. The data analysis requires more computational resources and consumes much time when the data is of huge volume. Hence, pre-processing and dimensionality reduction techniques are used to remove the irrelevant or noise features from the data in order to reduce the time and the resource usage. The reduction in the number of data should not reduce the accuracy of the task at hand. So, dimensionality reduction is used in combination with the classification techniques. The following sections discuss the several dimensionality reduction methods and classification processes.

Alexander *et al.*<sup>8</sup> has discussed Naïve Bayes classification for twitter dataset. In their work, they collected a corpus of 300000 text posts which is split into three sets of texts. i) text containing positive emotion such as happiness, joy ii) text containing negative emotions such as sadness, anger iii) objective text which contain only factual information. For feature extraction they use n-grams. Vivek *et al.*<sup>1</sup> has also discussed Naïve Bayes classification for text. For feature extraction they use bigram and trigram. They use this classification model, for

handling of negation texts. Daniel *et al.*<sup>10</sup> has discussed classification using support vector machines. They filtered documents based on subjective or objective text. In order to train the SVM classifier every document was represented as a vector and the goal was to determine the optimal decision boundary and finally classify the reviews as positive and negative. Abbasi *et al.*<sup>2</sup> has also discussed classification using Support vector machines for web forum postings. For feature selection they have used Entropy Weighted Genetic Algorithm that includes information-gain (IG) heuristic (filter model for ranking features) with genetic algorithm (GA) to improve the performance of feature selection. Cambria *et al.*<sup>3</sup> has discussed classification using K-Nearest Neighbour.

They use AffectNet dataset in which each concept is represented by a vector in a space. They perform truncated singular value decomposition (TSVD) on AffectNet. They use k-NN approach to divide Affective Space into emotional clusters after analyzing texts for sentiments. Cambria *et al.*<sup>7</sup> has discussed various dimensionality reduction techniques. They use AffectNet dataset in which each concept is represented by a vector in a space i.e. Affective Space which is a multidimensional vector space obtained by applying Principal Component Analysis and Truncated Singular value decomposition on AffectNet. In order to organize and Interpret Affective Space, Hour Glass Model is used. It organizes the 6 primary emotions (anger, happiness, disgust, surprise, sadness, fear) around 4 independent dimensions Pleasantness, Attention, Sensitivity and Aptitude. Here Artificial Neural Network is used to predict which class each concept belongs to. Gulen *et al.*<sup>11</sup> has also discussed K-Nearest Neighbour classification. They use initial files generally called as training documents for classification. New document is then compared with all these training documents with respect to terms they share. If it is a match, then newly categorized document is also added to the training documents set. Nisha *et al.*<sup>4</sup> has discussed classification using Random Forest. Here Singular value decomposition

is used to find the importance of the word. RF is used for generation of 25 trees which integrates multiple unstable classifiers in order to improve the performance. Long-Sheng *et al.*<sup>5</sup> has discussed Artificial Neural Network classification to detect comments of harmful negative blogger's quickly and effectively. Nisha *et al.*<sup>6</sup> has also discussed classification using Artificial Neural Networks. They use singular value decomposition for feature extraction. Extracted features are sent to neural network for training. Two types of enhancements are implemented i) additional weights added from input to output layer ii) enhanced back propagation algorithm to get best results.

The work undertaken in this paper is to recognize the project that deserves A+ in the website donorchoose.org. Donorchoose.org is an online charity website that helps the students in need. Here they list many number of projects needed for schools which require pencils, acrylic paint, paint brushes, crayons, light table, 3D scanner, activity books, computer to microscopes. It allows eligible teachers to propose the project useful to their student with detailed description about it. Donors also surf the projects they are willing to donate and they can create account and give their donation. Once funding request is satisfied then this site directs the materials needed for the project to the vendors. The site notifies the availability of materials to the teacher's mail address and they ship the material to the school.

This helps the school students in need, helps the donors to donate to useful projects, helps the school teachers to choose the best project useful to their school students. The research work uses this dataset to find the best accuracy when combining different dimensionality reduction and classification techniques. This paper employs PCA, ICA and TSVD to determine the optimal dimensionality reduction technique with Artificial Neural Network, Support vector machine and Random forest classification with good accuracy to identify the best project.

## II. MATERIALS AND METHODS

### Experimental Design

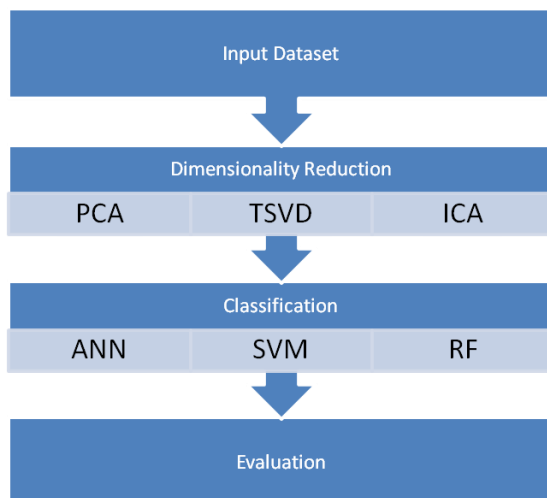


Fig. 1. Overall system design

Overall system design is given in Fig. 1, First different datasets of this project are merged using SQL inner join query. Once the dataset has been merged it is preprocessed. This preprocessed dataset is subjected to PCA, ICA, TSVD to attain dimensionality reduction and then this reduced dataset is applied to ANN, SVM and RF to find best project and also to identify best dimensionality reduction and classification techniques with the following metrics: F-measure, precision and recall.

### A. Principle Component Analysis

PCA is a Dimensionality Reduction technique. It reduces the dimensionality of a dataset with respect to the variations present in it. It takes as input the noisy, corrupted dataset and filters out the noise and redundant data in it. PCA selects the normalized direction in m-dimensional space such that the variance is maximized and notes it as p1. It finds the second highest variance in another direction perpendicular to the first and marks as the second principal component. This could continue until m directions are selected. The resulting ordered set of principal components are the required dimensions.

PCA consists of the following steps:

- 1) Convert all correlated variables in dataset into linearly uncorrelated variables called principal components. The number of Principal components should be less than or equal to the original variable numbers.
- 2) Calculate the covariance matrix from the dataset and mark it as A. It is the degree to which the variables are linearly correlated and is represented by their covariance.
- 3) Calculate the eigen values by solving the determinant.

$$|A - \lambda I| = 0 \quad \dots(1)$$

- 4) Calculate the eigen vector by solving the matrix with eigen values. Thus eigen vector X is given by

$$(A - \lambda I) * [X] = [0] \quad \dots(2)$$

- 5) Finally obtain the coordinates of data points in the direction of the eigen vectors.

### B. Independent Component Analysis

ICA finds the independent components from multivariate data. It calculates the covariance matrix from the dataset. ICA uses centering pre-processing where it subtracts the mean to create a zero mean variable and then uses whitening pre-processing with the help of eigenvalue decomposition of covariance matrix.

Following are the steps used in ICA

- 1) First it finds the Independent components which are latent variables. A and s are estimated using only the observable random vector x. So after estimating A, we can compute  $W = A^{-1}$  and s is given by

$$s = Wx = A^{-1}x \quad \dots(3)$$

- 2) Then it performs centering a preprocessing strategy where it subtracts its mean vector to make x a zero-mean variable. It estimates mean vector of s by  $A^{-1}m$ , where m is the subtracted mean and is given by

$$x = x' - E\{x'} \quad \dots(4)$$

3) Then it does whitening, the second preprocessing strategy where it transforms  $x$  so that its components are uncorrelated. If  $\tilde{x}$  is the whitened data, then the covariance matrix of the whitened data is the identity matrix and are given by

$$E \{ \tilde{x} \tilde{x}^T \} = I \quad \dots(5)$$

Using eigenvalue decomposition (EVD), covariance matrix of the data is given by

$$E \{ x x^T \} = E D E^T \quad \dots(6)$$

where  $E$  is the orthogonal matrix of eigenvectors and  $D$  is the diagonal matrix of its eigenvalues. Once eigenvalue decomposition is done, the whitened data is given by

$$x \leftarrow E D^{-1/2} E^T x \quad \dots(7)$$

4) Finally it performs Fast ICA using the following algorithm.

- i. Choose an initial random weight vector  $w$
- ii. Let  $w^+ = E \{ x g(w^T x) \} - E \{ g'(w^T x) \} w$

Derivatives of contrast functions  $G$  are given by

$$\begin{aligned} g_1(u) &= \tanh(a_1 u), \\ g_2(u) &= u \exp(-u^2/2) \end{aligned} \quad \dots(8)$$

- iii. Let  $w = w^+ / \|w^+\|$ . (Normalization step)
- iv. If  $w$  does not converge go to step ii

### C. Truncated Singular Value Decomposition

TSVD removes unnecessary data that are linearly dependent in the point of view of Linear Algebra. This is a well-known matrix factorization technique that factors an  $m \times n$  matrix  $R$  into three matrices as in equation (9) where  $U$  and  $V$  are two orthogonal matrices

$$R \approx U S V^T \quad \dots(9)$$

$S$  is a diagonal matrix of size  $r \times r$  having all singular values of matrix  $R$  as its diagonal entries. All the entries of matrix  $S$  are positive and stored in decreasing order. The columns of  $U$  are the left singular vectors of the matrix  $R$ , and the columns of  $V$  are the right singular vectors. To compute the TSVD is to find the eigenvalues and the eigenvectors of  $RR^T$  and  $R^T R$ . The eigenvectors of  $RR^T$  are the columns of  $U$  and the eigenvectors of  $R^T R$  are the columns of  $V$ .

Following are the steps used in TSVD

1. Compute  $RR^T$  normally.
2. Compute the eigenvalues and the eigenvectors of  $RR^T$  normally.
3. Compute  $R^T R$ .
4. Compute the eigenvalues and the eigenvectors of  $R^T R$ .
5. Compute the square root of the common positive eigenvalues of  $RR^T$  and  $R^T R$ .
6. Finally, assign the computed values to  $U, V$  and  $A$ .

TSVD helps to produce a low-dimensional representation of the original donorschoose.org dataset.

### D. Artificial Neural Network

An artificial neural network is one of the most used predictive modeling techniques. This is a mathematical model or computational model that is stimulated by the structure or functional aspects of biological neural networks. The algorithm used is the feed forward neural network that allows signals to travel from input to output neurons in one way only. There is no feedback (loops) i.e. the output of any layer does not affect that same layer. The network is trained by providing it with input and matching output patterns. For activation, bias node is included with hidden and output neurons.

The network is trained by providing it with input and matching output patterns. These input-output pairs can be provided by the system which contains the self-supervised neural network. The Back propagation neural network is a multilayered feed forward neural network and can readjust the weight values internally in the network.

Back Propagation neural network with hidden layers is shown in Fig. 2.

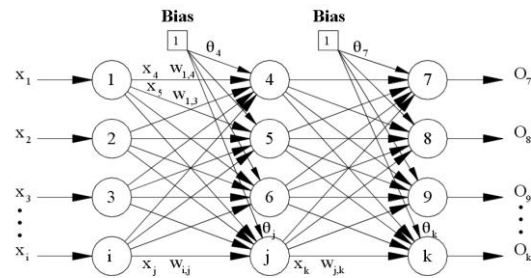


Fig. 2. Back Propagation neural network

This Back propagation network like other classification techniques has two stages namely, training and testing. During the training phase the given input and output learning functions are used to automatically adjust the network's weights and biases and learn accordingly. In the testing phase, for the given input it should classify to the correct class label with the help of the training data.

#### Training Phase

- Step 1: Initialize the weights and learning rate to some random values
- Step 2: Perform steps 2-9 until the stopping condition is false.
- Step 3: Each input unit receives input signal  $x_i$  and sends it to the hidden unit ( $i=1$  to  $n$ )
- Step 4: Each hidden unit  $Z_j(j=1$  to  $p$ ) sums its weighted input signals to calculate the net inputs and is given by

$$z_{inj} = v_{oj} + \sum_i x_i v_{ij} \quad \dots(12)$$

Calculate output of the hidden unit by applying its activation functions over  $Z_{inj}$  (binary or bipolar sigmoid activation function) which is given by

$$z_j = f(z_{inj}) \quad \dots(13)$$

Then send the output signal from the hidden unit to the input of output layer units.

Step 5: For each output unit  $y_k$  ( $k=1$  to  $m$ ), calculate the net input given by

$$y_{ink} = w_{ok} + \sum_{j=1}^p z_j w_{jk} \quad \dots(14)$$

Apply the activation function to compute output signal given by

$$y_k = f(y_{ink}) \quad \dots(15)$$

Step 6: Each output unit  $y_k$  ( $k=1$  to  $m$ ) receives a target pattern corresponding to the input training pattern and computes the error correction term and is given by

$$\delta_k = (t_k - y_k) f'(y_{ink}) \quad \dots(16)$$

On the basis of the calculated error correction term, update the change in weights and bias using

$$\begin{aligned} \Delta w_{ik} &= \alpha \delta_k z_j \\ \Delta w_{ok} &= \alpha \delta_k \end{aligned} \quad \dots(17)$$

Send  $\delta_k$  to the hidden layer backwards.

Step 7: Each hidden unit ( $Z_j$ ,  $j=1$  to  $p$ ) sums its delta inputs from the output units and is given by

$$\delta_{inj} = \sum_{k=1}^m \delta_k w_{jk} \quad \dots(18)$$

The term  $\delta_{inj}$  gets multiplied with the derivative of  $f(z_{inj})$  to calculate the error term and is given by

$$\delta_j = \delta_{inj} f'(z_{inj}) \quad \dots(19)$$

On the basis of the calculated  $\delta_j$ , update the change in weights and bias using

$$\begin{aligned} \Delta v_{ij} &= \alpha \delta_j x_i \\ \Delta v_{oj} &= \alpha \delta_j \end{aligned} \quad \dots(20)$$

Step 8: Each output unit ( $y_k$ ,  $k=1$  to  $m$ ) updates the bias and weights and is given by

$$\begin{aligned} w_{jk}(\text{new}) &= w_{jk}(\text{old}) + \Delta w_{jk} \\ w_{ok}(\text{new}) &= w_{ok}(\text{old}) + \Delta w_{ok} \end{aligned} \quad \dots(21)$$

Each hidden unit ( $Z_j$ ,  $j=1$  to  $p$ ) updates its bias and weights and is given by

$$\begin{aligned} v_{ij}(\text{new}) &= v_{ij}(\text{old}) + \Delta v_{ij} \\ v_{oj}(\text{new}) &= v_{oj}(\text{old}) + \Delta v_{oj} \end{aligned} \quad \dots(22)$$

Step 9: Check for the stopping condition. This may be based on certain number of epochs reached or when the actual output equals the target output.

### E. Support Vector Machine

SVM is a supervised learning algorithm in which learning machine is given a set of input with the associated labels. SVMs construct a hyperplane that separates two classes to achieve maximum separation between the classes. Separating the classes with large margin minimizes generalization error. The two planes parallel to the classifier and which pass through one or more points in

the dataset are called bounding planes. The points falling on the bounding planes are called support vectors. This support vectors are important training points which define the hyperplane. Separating the classes with a large margin minimizes the generalization error i.e. chance of making error during classification when new examples arrive for classification.

Fig. 3 illustrates the concept of maximum margin with bounding planes and support vectors. The distance between bounding planes is called the 'margin' and SVM learning is to finding a central hyperplane which maximises this margin.

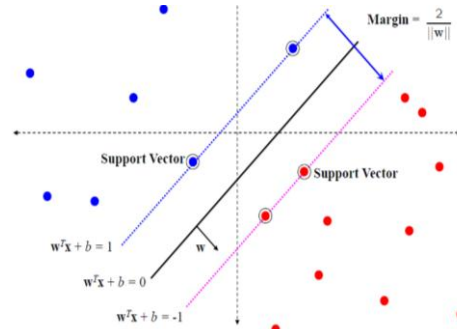


Fig. 3. Maximal Margin Classifier

There are many linear classifiers that can separate the data but there is only one that maximizes the margin i.e. the distance between the nearest data point of each class in the given dataset. This linear classifier is termed by optimal separating hyperplane. The support vectors are the data points that are closest to the separating hyperplane. These points are on the boundary of the slab.

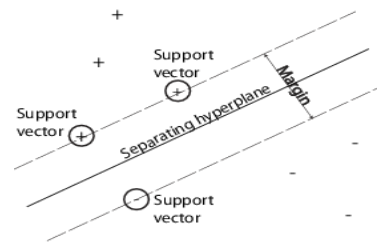


Fig. 4. SVM Classification

The Fig. 4 illustrates these definitions, with + indicating data points of type 1, and - indicating data points of type -1

### B Training Phase

First SVM finds separating hyperplane of the form

$$w_1 x_1 + w_2 x_2 + \dots + w_n x_n - \gamma = 0 \quad \dots(23)$$

The two bounding plane are of the form given below

$$\begin{aligned} w_1 x_1 + w_2 x_2 + \dots + w_n x_n - \gamma &\geq +1 \\ w_1 x_1 + w_2 x_2 + \dots + w_n x_n - \gamma &\leq -1 \end{aligned} \quad \dots(24)$$

The aim is to find maximally separating bounding planes such that data points with class label  $d=-1$  satisfy the constraint given by

$$w_1 x_1 + w_2 x_2 + \dots + w_n x_n - \gamma \leq -1 \quad \dots(25)$$

and data points with class label  $d=+1$  satisfy the constraints given by

$$w_1x_1 + w_2x_2 + \dots + w_nx_n - \gamma \geq +1 \quad \dots(26)$$

The resulting equation in the matrix form is given by

$$D[Aw - \gamma] \geq e \quad \dots(27)$$

Where  $D$  is vector representing a target value of  $m$  data points which takes the value  $+1$  or  $-1$ ,  $A$  is the function that maps the input vector into a high dimensional feature vector,  $w$  is a coefficient vector corresponding to the hyper plane and  $\gamma$  is a scalar which is generally known as bias term.

Distance between the bounding hyperplanes is  $\frac{2}{\sqrt{w_1^2 + w_2^2}}$ . The aim here is to find such a  $w$  and  $\gamma$  that maximizes this distance and at the same time it should satisfy the matrix constraint equation because it minimizes the number of misclassification. Once  $w$  and  $\gamma$  is obtained, then decision boundary is  $w^T x - \gamma = 0$ .

To minimize the classification error and bound on the dimension of the classifier during the training phase

$$y^i [(w \cdot x^i) + b] \geq 1 - \xi_i, \quad \dots(28)$$

$\xi_i$  is a positive quantity that is added or subtracted to the left of the inequality to satisfy the above constraints.

Thus the resulting equation in matrix form is given by

$$D[Aw - \gamma] + \xi \geq e \quad \dots(29)$$

The objective function is given by

$$C e^T \xi + \frac{1}{2} w^T w \quad \dots(30)$$

$C$  is a scalar value that controls the weightage and  $e^T \xi$ . Minimization of the above quantity with respect to  $w$  and  $\xi$  causes maximum separation between bounding planes.

Since the objective function is quadratic, quadratic optimization algorithms can identify which training points  $x_i$  are support vectors with non-zero Lagrangian multipliers  $\alpha_i$ . Quadratic programming is required for the solution and is given by

$$\min_{w, \gamma, \xi} \frac{1}{2} w^T w + C \sum_{i=1}^m \xi_i \quad \dots(31)$$

### F. Random Forest

This is an ensemble classifier which uses many decision tree models. This classifier consists of a collection of tree-structured classifiers  $\{h(x, \Theta_k), k = 1, \dots\}$  where the  $\Theta_k$  are independently, identically distributed random trees and each tree casts a unit vote for the final classification of input  $x$ . Random Forest uses the Gini index for determining the final class in each tree. The final class of each tree is aggregated and voted by weighted values to construct the final classifier. It runs well on large data bases. Without variable deletion, it can handle thousands of input variables and can also locate outliers. Its learning is fast and produces a high accuracy.

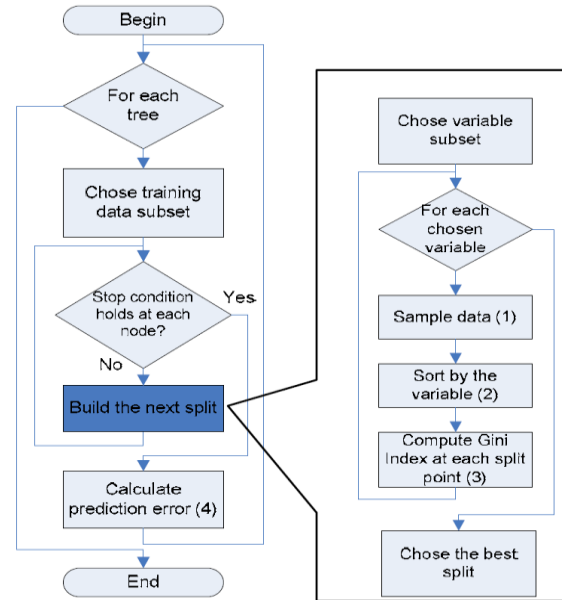


Fig. 5. Flowchart of Random Forest Algorithm

### The Algorithm

- Let the number of training cases be  $N$ , and the number of variables in the classifier be  $M$ .
- The number  $m$  of input variables are used to determine the decision at a node of the tree.  $m$  should be much less than  $M$ .
- Choose a training set for this tree by choosing  $N$  times with replacement from all  $N$  available training cases. Use the rest of the cases to estimate the error of the tree by predicting their classes.
- For each node of the tree, randomly choose  $m$  variables on which to base the decision at that node. Calculate the best split based on these  $m$  variables in the training set.
- Each tree is fully grown and not pruned.
- Random Forest uses the Gini index to construct decision trees. The Gini index of node impurity is the measure most commonly chosen for classification-type problems. If a dataset  $T$  contains examples from  $n$  classes, Gini index,  $Gini(T)$  is given by

$$Gini(T) = 1 - \sum_{j=1}^n (p_j)^2 \quad \dots(32)$$

Where  $p_j$  is the relative frequency of class  $j$  in  $T$

- If a dataset  $T$  is split into two subsets  $T_1$  and  $T_2$  with sizes  $N_1$  and  $N_2$  respectively, the Gini index of the split data contains examples from  $n$  classes, the Gini index ( $T$ ) is given by

$$Gini_{split}(T) = \frac{N_1}{N} gini(T_1) + \frac{N_2}{N} gini(T_2) \quad \dots(33)$$

## III. RESULTS

### Implementation Steps

This work used publically available donorschoose.org dataset. It contains 52 attributes. Pre-processing done is as described below

All the datasets of donorschoose.org like project, donation, resource etc were merged using innerjoin SQL query with project id as the primary key. Once the final

dataset has been merged, following are the preprocessing done on the various attributes in the dataset.

1. All hexadecimal values are converted into decimal values using visual basic script
2. All categorical values are converted into numerical values. i.e. vendor name, project resource type, school metro, poverty level
3. All attribute values are rounded with a maximum of two decimal value digits i.e. donation to project, donation\_optional support, total price excluding optional support, total price including optional support, item unit price
4. Random numbers fixed for donation amount for ranges between values below 10, 10 to 100 etc
5. Normalized all values using General Numeric Normalization Algorithm in Matlab

Once the preprocessing step is over it is subjected to PCA to attain good dimensionality reduction. First obtain principal component coefficients on preprocessed dataset which is a p-by-p matrix, each column containing coefficients for one principal component as given in equation (1). The columns are in order of decreasing component variance. eigen values are obtained from the covariance matrices that satisfy the equation given in (2). eigenvectors are formed from eigen values matrix as given in equation (3). All eigenvectors of asymmetric matrix are perpendicular to each other.

Preprocessed dataset is also subjected to ICA to attain dimensionality reduction. First it finds the Independent components which are latent variables as given in equation (4). Then it performs preprocessing first using centering where it subtracts its mean vector to make x a zero-mean variable as given in equation (5). It preprocesses again using whitening and eigen-value decomposition (EVD) of covariance matrix as given in equation (6). Finally it performs Fast ICA Algorithm as given in equation (9).

Preprocessed dataset is also subjected to TSVD to attain dimensionality reduction. First it performs the matrix factorization as given in equation (10). i.e it finds matrix (R) combining the two orthogonal matrices and the diagonal matrix and then it finds eigenvalues and eigenvectors of  $RR^T$  and  $R^TR$ . Finally the square root of the common positive eigenvalues of  $RR^T$  and  $R^TR$  are computed and the dimensionality reduction matrix is achieved. Table 1, shows the sample eigen vectors of PCA, ICA & TSVD

| Dimensionality reduction            | Sample Eigenvectors |        |        |        |
|-------------------------------------|---------------------|--------|--------|--------|
| <b>Principle Component Analysis</b> | 0.1523              | 0.1523 | 0.0583 | 0.0583 |
|                                     | 0.2340              | 0.2340 | 0.0660 | 0.0660 |
|                                     | 0.0252              | 0.0252 | 0.0453 | 0.0453 |
|                                     | 0.1291              | 0.1291 | 0.2812 | 0.2812 |
|                                     | 0.1505              | 0.1505 | 0.1094 | 0.1094 |
|                                     | 0.0310              | 0.0310 | 0.0451 | 0.0451 |
|                                     | 0.1544              | 0.1544 | 0.0204 | 0.0204 |
|                                     | 0.0259              | 0.0259 | 0.2126 | 0.2126 |
|                                     | 0.0302              | 0.0302 | 0.1069 | 0.1069 |
|                                     | 0.0593              | 0.0593 | 0.0070 | 0.0070 |
|                                     | 0.1473              | 0.1473 | 0.1616 | 0.1616 |

|   |                                       |        |        |        |        |
|---|---------------------------------------|--------|--------|--------|--------|
|   | 0.0298                                | 0.0298 | 0.3466 | 0.3466 |        |
|   | 0.0997                                | 0.0997 | 0.0206 | 0.0206 |        |
|   | 0.1309                                | 0.1309 | 0.2182 | 0.2182 |        |
|   | 0.0722                                | 0.0722 | 0.1291 | 0.1291 |        |
|   | 0.3104                                | 0.3104 | 0.1038 | 0.1038 |        |
|   | 0.0619                                | 0.0619 | 0.1230 | 0.1230 |        |
|   | 0.1513                                |        |        |        |        |
|   | <b>Independent Component Analysis</b> | 0.9984 | 0.9990 | 0.9842 | 0.9842 |
|   |                                       | 0.1862 | 0.1862 | 0.7117 | 0.7117 |
|   |                                       | 0.0679 | 0.0321 | 0.0387 | 0.1306 |
|   |                                       | 0.1306 | 0.7484 | 0.7484 | 0.0445 |
|   |                                       | 0.0445 | 0.3317 | 0.0436 | 0.0087 |
|   |                                       | 0.0823 | 0.0823 | 0.2600 | 0.2600 |
| 0.2176  |                                       | 0.2176 | 0.3749 | 0.0119 |        |
| 0.0173  |                                       | 0.0702 | 0.0702 | 0.5546 |        |
| 0.5546  |                                       | 0.2817 | 0.2817 | 0.6951 |        |
| 0.0015  |                                       | 0.0027 | 0.0012 | 0.0012 |        |
| 0.0307  |                                       | 0.0307 | 0.0167 | 0.0167 |        |
| 0.0202  |                                       | 0.0043 | 0.0051 | 0.0007 |        |
| 0.0007  |                                       | 0.0132 | 0.0132 | 0.0644 |        |
| 0.0644  |                                       | 0.0899 | 0.0059 | 0.0072 |        |
| 0.0012  |                                       | 0.0012 | 0.0113 | 0.0113 |        |
| 0.1036  |                                       | 0.1036 | 0.0499 | 0.0003 |        |
| 0.0001  |                                       | 0.0026 | 0.0026 | 0.0623 |        |
| 0.0623  |                                       | 0.0398 | 0.0398 | 0.0937 |        |
| 0.0013  |                                       | 0.0017 | 0.0007 | 0.0007 |        |
| 0.0000  |                                       | 0.0000 | 0.1009 | 0.1009 |        |
| 0.1362  |                                       | 0.0057 | 0.0064 | 0.0025 |        |
| 0.0025  |                                       | 0.0046 | 0.0046 | 0.0205 |        |
| 0.0205  |                                       | 0.0158 | 0.0041 | 0.0042 |        |
| 0.0003  | 0.0003                                | 0.0069 | 0.0069 |        |        |
| 0.0310  | 0.0310                                | 0.0228 | 0.0013 |        |        |
| 0.0008  | 0.0000                                | 0.0000 | 0.0060 |        |        |
| 0.0060  |                                       |        |        |        |        |
| <b>Truncated Singular Value Decomposition</b> | 0.0002                                | 0.0002 | 0.1118 | 0.1666 |        |
|   | 0.1943                                | 0.1555 | 0.1555 | 0.2465 |        |
|   | 0.5550                                | 0.0003 | 0.0003 | 0.2069 |        |
|   | 0.2137                                | 0.1491 | 0.0747 | 0.0747 |        |
|   | 0.2265                                | 0.3091 | 0.0003 | 0.0003 |        |
|   | 0.2701                                | 0.2704 | 0.1450 | 0.1082 |        |
|   | 0.1082                                | 0.0281 | 0.1954 | 0.0003 |        |
|   | 0.0003                                | 0.2911 | 0.3124 | 0.0713 |        |
|   | 0.0109                                | 0.0109 | 0.0579 | 0.0968 |        |
|   | 0.0002                                | 0.0002 | 0.2661 | 0.2633 |        |
|   | 0.2109                                | 0.0270 | 0.0270 | 0.3476 |        |
|   | 0.0792                                | 0.0001 | 0.0001 | 0.1990 |        |
|   | 0.1871                                | 0.1581 | 0.0219 | 0.0219 |        |
|   | 0.0389                                | 0.0908 | 0.0000 | 0.0000 |        |
|   | 0.1001                                | 0.0817 | 0.0853 | 0.0218 |        |
|   | 0.0218                                | 0.0246 | 0.1169 | 0.0002 |        |
|   | 0.0002                                | 0.0148 | 0.0644 | 0.1434 |        |
|   | 0.1441                                | 0.1441 | 0.0304 | 0.0101 |        |
|   | 0.9903                                | 0.9903 | 0.0977 | 0.0052 |        |
| 0.6309  | 0.7681                                | 0.7681 | 0.6497 |        |        |
| 0.0176  | 0.0003                                | 0.0003 | 0.2202 |        |        |
| 0.2654  | 0.0123                                | 0.1348 | 0.1348 |        |        |
| 0.0858  | 0.2405                                | 0.0003 |        |        |        |

Table 1. Sample eigenvectors of PCA, ICA & TSVD

The reduced dataset is applied to the Artificial Neural Network Classifier. Algorithm used is Back Propagation

Feed forward neural network. Here eigen vector values are used to feed into the input layers. The data processing can lengthen over multiple layers. The network is trained by providing it with input and matching output patterns (eigenvectors). These input-output pairs can be provided by the system which contains the self-supervised neural network. Back propagation can readjust the weight values internally in the network. This Back propagation network has two stages namely, training and testing.

Table 2, illustrates Neural network performance with different dimensionality reduction techniques. The first row, illustrates the performance of PCA with ANN. This table gives the training, validation and test performance result. The train curve depicts training of neural network for the PCA inputs. Validation curve validates the trained data. Test curve depicts its performance for the trained data. Here training set and validation set have similar characteristics. The Mean square error is comparatively small. A small amount of overfitting has also occurred and the best validation performance is at epoch 3. The second row, illustrates the performance of ICA with ANN. Here training set and validation set have somewhat similar characteristics. The Mean square error is comparatively small. Significant over fitting has occurred and best validation performance is at epoch 11. The third row, illustrates the performance of TSVD with ANN. Here training set and validation set have somewhat similar characteristics. Mean square error is also small here. No Significant over fitting has occurred and the best validation performance is at epoch 34.

| Dimensionality reductions - Classification Techniques | (Mean Square Error) |                  |            |
|---|---------------------|------------------|------------|
|   | Train Curve         | Validation Curve | Test Curve |
| PCA – ANN   | .01                 | .012             | .015       |
| ICA – ANN   | .019                | .017             | .1         |
| TSVD - ANN  | .025                | .031             | .015       |

Table 2. Dimensionality reductions with Classification Techniques - Neural Network Performance Result

The reduced Dataset is applied to Support vector Machines. Algorithm used here is Quadratic optimization. This can identify which training points  $x_i$  are support vectors in which bounding planes are crossing. SVMs construct a hyperplane that separates two classes to achieve maximum separation between the classes. Separating the classes with large margin minimizes generalization error.

Table 3, illustrates SVM performance of different dimensionality reduction with classification techniques. First it illustrates training result for PCA with SVM for different weights. Hyperplane is formed after training the data. Here bounding planes are nearer to the hyperplane. It has less margin so generalization error is somewhat reduced. Second it illustrates training result for ICA with SVM for different weights. Here hyperplane is vertical and so it is sparse. Here it has large margin and misclassification and generalization error are mostly minimized. Third it illustrates training result for TSVD with SVM. Here also hyperplane is vertical. It has less margin and generalization error is somewhat reduced.

| Dimensionality reductions – Classification Techniques | Support Vectors |      |        |
|---|-----------------|------|--------|
|   | X               | Y    | Weight |
| PCA – ANN   | .14             | .12  | .1     |
|   | .14             | .11  | .1     |
|   | .07             | .13  | .1     |
|   | .02             | .04  | .05    |
|   | .04             | .045 | .05    |
|   | .23             | .04  | .05    |
| ICA – ANN   | .2              | .2   | .2     |
|   | 0               | .2   | .2     |
|   | .7              | .15  | .2     |
|   | 0               | .29  | .3     |
|   | .29             | .7   | .3     |
|   | .85             | .22  | .3     |
| TSVD – ANN  | .32             | .32  | .3     |
|   | .1              | .1   | .1     |
|   | .19             | .1   | .1     |
|   | .2              | .09  | .1     |
|   | .3              | .09  | .1     |
|   | 0               | .2   | .2     |
| .59   | .19             | .2   |        |
| .85   | .19             | .2   |        |

Table 3. Dimensionality reductions with Classification Techniques - SVM Performance Result

The reduced dataset is applied to Random Forests. First it collects randomly sampled training dataset (eigenvectors) and from that it collects random set of attributes because if all input variables are considered, large number of computation and high chance of over fitting may occur. Using Gini index develop a tree model as given in equation (32) and (33) and this process repeats for each branch until the stop condition is reached and the accuracy is calculated.

#### IV. PERFORMANCE ANALYSIS

Accuracy is the most common performance metric used in A+ project tasks. It is calculated as follows and is given by

$$\text{Accuracy} = \frac{\text{Total no. of correctly analysis project}}{\text{Total no. of projects}} \dots(34)$$

In order to evaluate the performance of the ANN and SVM algorithm, we validated each algorithm in comparison to other algorithms

Table 4, gives the comparison of result between the PCA-ANN, PCA-SVM, ICA-ANN, ICA-SVM, TSVD-ANN and TSVD-SVM. The results show that, TSVD-SVM performs better than all the methods in terms of accuracy.

TABLE 4. COMPARISON BETWEEN PCA-ANN, PCA-SVM, PCA-RF, ICA-ANN, ICA-SVM, ICA-RF, TSVD-ANN TSVD-SVM AND TSVD-RF

| Method    | Accuracy |
|-----------|----------|
| PCA – ANN | 87.3%    |
| PCA – SVM | 85.9%    |
| PCA – RF  | 96.2%    |

|            |       |
|------------|-------|
| ICA – ANN  | 88.8% |
| ICA – SVM  | 76.4% |
| ICA – RF   | 96.3% |
| TSVD - ANN | 94.4% |
| TSVD - SVM | 94.8% |
| TSVD – RF  | 88.4% |

- [11]. Gulen Toker, “Text Categorization using k-NN Classification”, <http://www.ceng.metu.edu.tr/~e120321/paper.pdf>
- [12]. Simon Tong, Daphne Koller, 2001, “Support vector Machine Active Learning with Applications to Text Classification”, *Journal of Machine Learning Research* (2001) 45-66.

## V. CONCLUSION

The accuracy rate using Principle component analysis, Independent component Analysis and Truncated Singular value decomposition dimensionality reduction technique with artificial neural network (ANN using Back propagation algorithm), Support vector machine (SVM using Quadratic optimization algorithms) and Random Forest are compared. By this analysis the result concluded that ICA with RF gives high accuracy rate compared to all other methods.

In future work, comparison has to be made on various dimensionality reduction (PCA, TSVD, ICA, LSI, RP) and classification techniques (ANN, SVM, Naïve Bayes, KNN, RF) to find the project that deserve A+ and also identify the best dimensionality reduction and classification technique based on the following measures: F-measure, precision, recall, ROC Curve.

## REFERENCES

- [1]. Vivek Narayanan, Ishan Arora and Arjun Bhatia, 2013, “Fast and accurate sentiment classification using an enhanced Naive Bayes model”. *Intelligent Data Engineering and Automated Learning IDEAL 2013* Volume 8206, 2013, pp 194-201.
- [2]. Abbasi, A. Chen.H, 2008, “Sentiment analysis in Multiple Languages: Feature selection for opinion classification in Web forums”, *Journal ACM Transactions on Information Systems (TOIS)* Volume 26 Issue 3, June 2008.
- [3]. Cambria, E., Mazzocco, T., Hussain, A., Eckl C., 2011, “Sentic Medoids: Organizing Affective Common Sense Knowledge in a Multi-Dimensional vector Space”, *ISNN 2011*, Volume 6677, 2011, Pp 601-610. Berlin: Springer.
- [4]. Nisha Jebaseeli A, Dr. Kirubakaran E, 2012, “M-Learning Sentiment Analysis with Data Mining Techniques”, *International Journal of Computer Science and Telecommunications* [Volume 3, Issue 8, August 2012].
- [5]. Long-Sheng Chen, Hui-Ju Chiu, 2009, “Developing a Neural Network Based Index for Sentiment Classification”, *International MultiConference of Engineers & Computer Scientists 2009*, p744.
- [6]. Nisha A, Kirubakaran E, 2013, “Neural Network Classification Algorithm with M-learning reviews to improve the classification accuracy”, *International Journal of Computer Applications* (0975 – 8887) Volume 71– No.23, June 2013.
- [7]. Cambria, E., Mazzocco, T., Hussain, A., 2013, “Application of multi-dimensional scaling and artificial neural networks for biologically inspired opinion mining”, *Biologically Inspired Cognitive Architectures* (2013) Volume 4, pages 41– 53.
- [8]. Alexander Pak, Patrick Paroubek, 2010, “Twitter as a Corpus for Sentiment Analysis and Opinion Mining”, <http://kaikuehne.github.io/war/PakandParoubek> (2010, Twitter as a Corpus for Sentiment Analysis and OpinionMining.pdf
- [9]. Daniel Pomerantz, 2007, “Sentiment Classification of Moviereview”, <http://www.cim.mcgill.ca/~dpomeran/mlproject.pdf>
- [10]. Daniel Pomerantz, 2014, “Sentiment Classification of Movie review”. *International Journal of Innovative Research in Computer Science & Technology (IJIRCST)* ISSN: 2347-5552, Volume-2 Issue-2, March 2014.