Graduation Admission Acceptance Prediction Using PCA, KPCA and logistic regression

Daniel Zakerifar Hitesh

Abstract— In this paper, we classify and extract the admission data of 400 students using principle component analysis By using this model, every student can predict their chances of acceptance by providing key factors. Before the classification task, important features can be extracted from the data set using principle component analysis. In this way, model's complexity can be reduced and accuracy will be increased in some cases.

Key word—Admission, principle component analysis, logistic *regression, kernel* PCA.)

I. Introduction

Many students apply for the universities around the world with hope of getting acceptance from them. Determining which student is accepted and which is not is a difficult task, and besides it usually takes a lot of time and effort[2]. As a result, in this paper we consider a data from the acceptance list of previous student and investigate their situation. This dataset was built with the purpose of helping students in shortlisting universities with their profiles. The predicted output gives them a fair idea about their chances for a particular university. Classification methods can be adopted for the chance of admittance in the university. One of these classification techniques, which we are going to use in this work, is logistic regression[3].

Increase in the number of variables in a classification problem leads to increase in the model complexity and the time needed for computation. Besides, insignificant variables can even decrease the model accuracy. Therefore, in this work we are going to use principle component analysis (PCA) and Kernel PCA(KPCA) to extract the most important features, before using the classification model. The rest of this paper is organized as follows: In section 2, the data which is used in this work is presented. In section 3 and 4, 5 PCA, KPCA and logistic regression are described and the results of implementations are provided. Finally in section 6, we conclude our work.

II. Graduation Admission Data

The Data used in this work is taken from Kaggle data set [1] and consists of 400 observations. The Data includes 7 features and a class variable which presents the authentication labels. These labels include 1 for having a research and 0 for not having that the dataset contains several parameters which are considered important during the application for Masters Programs. The parameters included are: 1.GRE Scores (out of 340) 2. TOEFL Scores (out of 120) 3. University Rating (out of 5) 4, 5. Statement of Purpose and Letter of Recommendation Strength (out of 5) 6. Undergraduate GPA (out of 10) 7. Research Experience (either 0 or 1) 8. Chance of Admit (ranging from 0 to 1).

Figure 1 shows the first 4 rows of this data set, and figure 111 presents the summary of different features.

	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
0	337	118	4	4.5	4.5	9.65	1	0.92
1	324	107	4	4.0	4.5	8.87	1	0.76
2	316	104	3	3.0	3.5	8.00	1	0.72
3	322	110	3	3.5	2.5	8.67	1	0.80
4	314	103	2	2.0	3.0	8.21	0	0.65

Figure 1: **First 4 rows graduation admission data**

The first two features of the data set are plotted against each other in figure 2. Since different attributes have different measurements, we standardize the data before going into PCA and regression part.

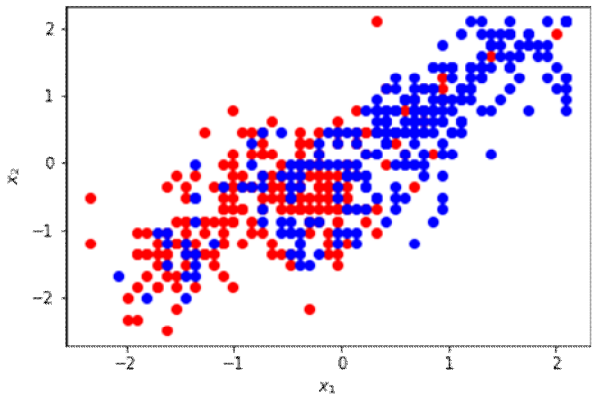

Figure : 2D visualization of data.

	Unnamed: 0	GRE Score	TOEFL Score	University Rating	SOP	\
count	500.000000	500.000000	500.000000	500.000000	500.000000	
mean	250.500000	316.472000	107.192000	3.114000	3.374000	
std	144.481833	11.295148	6.081868	1.143512	0.991004	
min	1.000000	290.000000	92.000000	1.000000	1.000000	
25%	125.750000	308.000000	103.000000	2.000000	2.500000	
50%	250.500000	317.000000	107.000000	3.000000	3.500000	
75%	375.250000	325.000000	112.000000	4.000000	4.000000	
max	500.000000	340.000000	120.000000	5.000000	5.000000	

Figure 2: SUMMARY OF Graduation admission DATA

To have a better intuition into the relationship between the attributes of the data set, bi variant scatter plot is shown in figure 4.

As shown in the Box plot for standardize data in figure 5, LOR and CGPA are the only features that have outliers

SOP has the biggest median, and University ranking and TOEFL Score have the smallest median. Based on this plot, GRE feature distribution seems to be the most similar to normal distribution..

Figure 4: bi variant scatter plot.

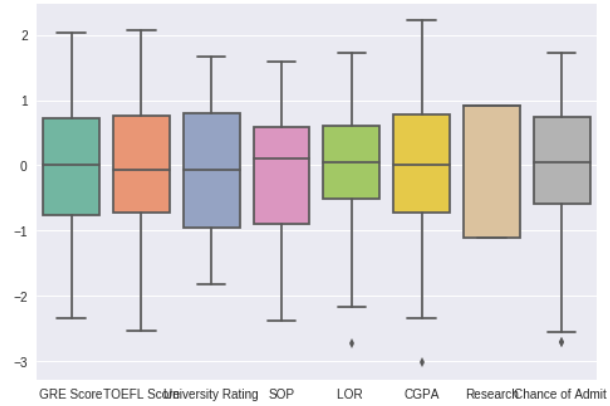

Figure 5: Box plot of the graduation admission data.

III. PRINCIPLE COMPONENT ANALYSIS

Principal component analysis (**PCA**) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables (entities each of which takes on various numerical values) into a set of values of linearly uncorrelated variables called **principal components**. If there are n observations with p variables, then the number of distinct principal components is min(p,n-1) . This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors (each being a linear combination of the variables and containing *n* observations) are an uncorrelated orthogonal basis set. PCA is sensitive to the relative scaling of the original variables. [4].

In general data matrix X, which is a n _ p matrix will be pre- processed before the PCA analysis. First, it will be zero centered, which means the mean of each column will be equal to zero. We have zero centered the admission data and calculated the covariance matrix S, which is shown in Figure 6.

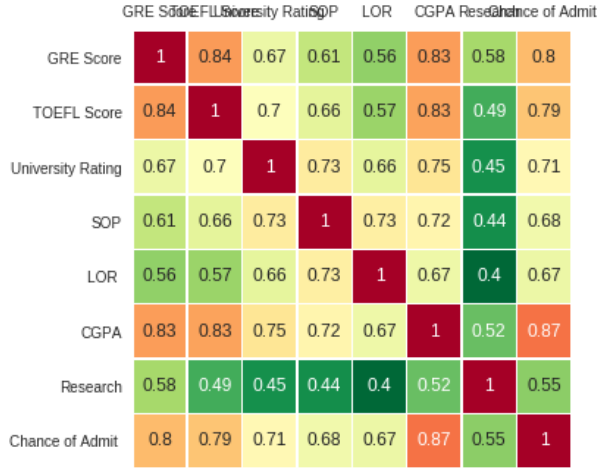

Figure 6: Covariance matrix of zero centered data.

Next step in PCA analysis is to implement the Eigen value decomposition of the covariance matrix S, as follows:

$$S = A \Lambda A^{\dagger} | \qquad (1)$$

where A is a $p \times p$ matrix of eigenvectors, and Λ is a diagonal matrix of eigenvalues. We have calculated the eigenvalues and eigenvectors of the covariance matrix for the graduation admission. The following matrix is the eigenvectors:

$$A^{\dagger} = [\begin{matrix} -.37 & -0.27 & -0.32 & -0.064 & 0.012 & 0.21 & 0.79 & 0.14 \\ -0.37 & -0.11 & -0.42 & -0.028 & -0.084 & 0.61 & -.54 & -0.012 \\ -0.35 & 0.27 & 0.13 & 0.64 & 0.64 & 0.048 & 0.009 & 0.0043 \\ -0.35 & 0.37 & 0.2 & 0.34 & -0.76 & -0.032 & .074 & 0.079 \\ -0.32 & 0.44 & 0.44 & -0.65 & 0.18 & 0.24 & 0.065 & -0.023 \\ -0.39 & -0.014 & -0.23 & -0.12 & -0.0036 & -0.45 & -0.039 & -0.76 \\ -0.27 & -0.71 & 0.63 & 0.067 & -0.043 & 0.045 & -0.11 & -0.073 \\ -0.39 & -0.072 & -0.15 & -0.19 & 0.083 & -0.56 & -0.25 & 0.63 \end{matrix}]^{T}$$

the following vector is the eigenvalues. These values tell us about the variance in particular dimensions.

$\lambda = [5.54318562, 0.74509663, 0.57216237,$
$0.39061073, 0.26443388, 0.21392805,$
$0.15851504, 0.11206769]$

After determining the eigenvalues and eigenvectors, the

principle components can be calculated as follows:

$$Z = XA \qquad (2)$$

where X is the zero-centered data, and the columns of Z are the principle components. For the graduation admission authentication

data, the following shows the first and second principle components:

$$Z_1 = -0.37GRE - 0.27TOEFL$$
$$-0.32University - 0.064SOP$$
$$+0.012LOR + 0.21CGPA + 0.79\,\mathrm{Re}search + 0.14Chance$$

$$Z_2 = -0.37GRE - 0.11TOEFL$$
$$-0.42University - 0.028SOP$$
$$-0.084LOR + 0.61CGPA - 0.54\,\mathrm{Re}search - 0.012Chance$$

After calculating the principle components, they can be plotted against each other to have a better insight about their relationships. Figure 8 show this plot.

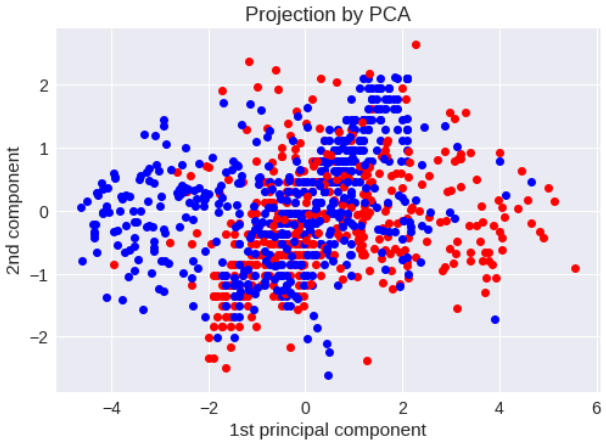

Figure 8: PC2 score Vs. PC1 score

Another important aspect of principle component analysis is that how much each attribute contributes to principle components.

For this purpose, PC coefficients are plotted against each other in Figure 9. The following items are some of the facts that can be inferred from these plots:

- Research is the feature which contributes to PC1 more than others.
- GRE and TOEFL scores have approximately similar coefficients for PC2.
- LOR is the feature which contributes to PC2 less than others.
- Some features have positive coefficients like research for PC1 and some negative like GRE Score for PC2.
- Research is much different for the other components.
- TOEFL and GRE has trivial effect on the PC1.
- Most of the variance for the PC1 is because of the research.

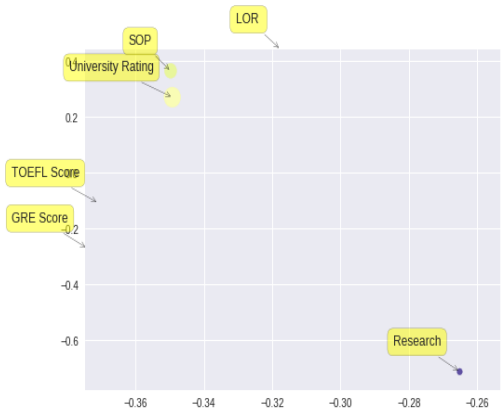

Figure 9: scatter plot of PC2 coefficient Vs. PC1 coefficient.

When the goal is to extract important information from the data matrix, the problem is to figure out how many components should be considered. There are different solutions to this problem. For instance, we can keep components whose Eigen values are larger than the average Eigen value. Another criteria which is used in this work is to calculate the explained variance for each component as follows:

$$l_j = \frac{\lambda_j}{\sum_{j=1}^{p} \lambda_j} \times 100\%, j = 1,..., p. \qquad (3)$$

Using the equation(3), we have calculated the explained variance for the components in graduation admisiion data set, and the result is as follows

$$l_1 = 0.6928982, l_2 = 0.09313708,$$
$$l_3 = 0.0715203$$

In figure 10,we plotted the first three component in PCA

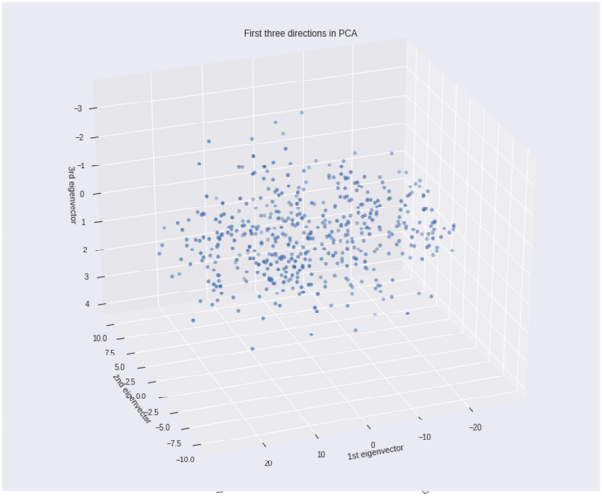

Figure 10: first three component

Based on these values, we can infer that the first two components account for more than 82% of the variance in the graduation admission authentication data set. Thus, the minimum dimension to represent our data is d = 2, and we can omit the last two components for our classification phase, which will be described in the next section.

From the Pareto plot in figure 11,12, the cumulative explained variance for the components can be inferred. As it can be easily observed, the first 2 components account for most of the variance in data.

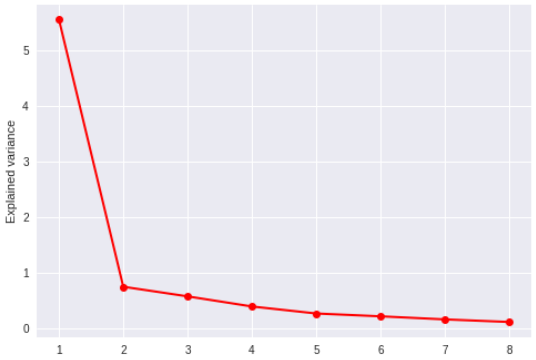

Figure 11: Explained variance plot

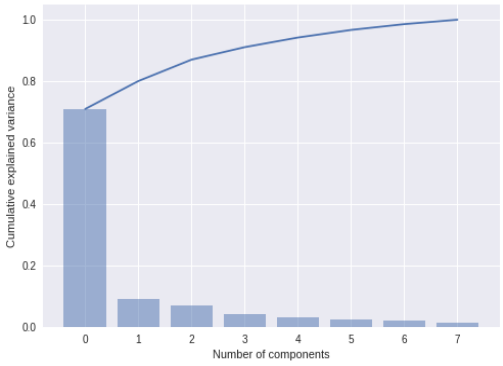

Figure 12: Pareto Chart

Another useful plot in PCA is biplot. This plot shows two kind of information at the same time: first it shows the PCA scores for observations, and second it shows how much each original variable contributes to the principle components. Observations are presented as points in the plot, and variables are presented as vectors. The axis of bi-plot are the principle components. The 2D and for the admission data set is presented in Figures 13. The following information can be inferred from these plots:

For PC1, SOP and LOR have negative coefficients. For PC2, SOP and LOR have positive coefficients, but Research and GRE have negative ones.

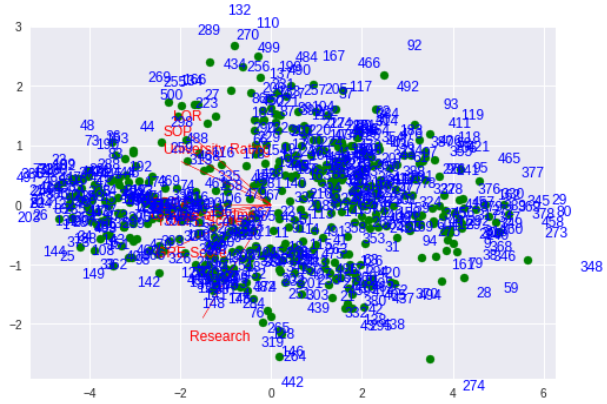

Figure 13: biplot

IV. KERNEL PCA

The "classic" PCA approach is a linear projection technique that works well if the data is linearly separable. However, in the case of linearly inseparable data, a nonlinear technique is required if the task is to reduce the dimensionality of a dataset.

To understand the utility of kernel PCA, particularly for clustering, observe that, while N points cannot in general be linearly separated in $d < N$ dimensions, they can almost always be linearly separated in $d > N$ dimensions. That is, given N points, x_i if we map them to an N-dimensional space with

$$\Phi(\mathbf{x}_i) \text{ where } \Phi : R^d \rightarrow R^N, \qquad (4)$$

it is easy to construct a hyper plane that divides the points into arbitrary clusters. Of course, this Φ creates linearly independent vectors, so there is no covariance on which to perform Eigen decomposition *explicitly* as we would in linear PCA. Figure 14 shows first 2 principal components after Linear PCA and 4 Clusters.

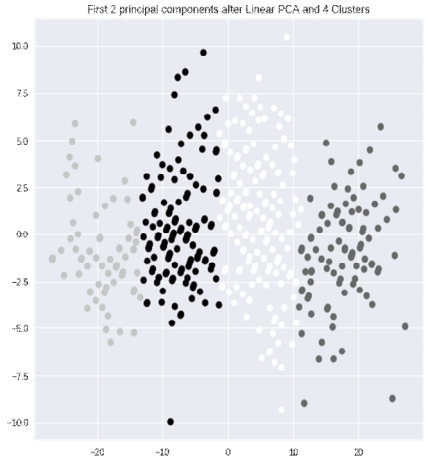

Figure 14: principal components after Linear PCA and 4 Clusters

Instead, in kernel PCA, a non-trivial, arbitrary Φ function is 'chosen' that is never calculated explicitly, allowing the possibility to use very-high-dimensional Φ's if we never have to actually evaluate the data in that space. Since we generally try to avoid working in the $\Phi(x)$ -space, which we will call the 'feature space', we can create the N-by-N kernel

$$K = K(X,Y) = (\varphi(X),\varphi(Y)) = \Phi(X)^T\Phi(X) \quad (5)$$

which represents the inner product space of the otherwise intractable feature space. The dual form that arises in the creation of a kernel allows us to mathematically formulate a version of PCA in which we never actually solve the eigenvectors and eigenvalues of the covariance matrix in the $\Phi(X)$ -space (see Kernel trick)[5]. The N-elements in each column of K represent the dot product of one point of the transformed data with respect to all the transformed points (N points). Some well-known kernels are shown in the example below.

Because we are never working directly in the feature space, the kernel-formulation of PCA is restricted in that it computes not the principal components themselves, but the projections of our data onto those components. To evaluate the projection from a point in the feature space $\Phi(X)$ onto the kth principal component v^x (where superscript k means the component k, not powers of k)

$$V^{K^T}\phi(x_i) = (\sum_{i=1}^{N} a_i{}^K \phi(x_i)^T \phi(x) \quad (6)$$

We note that $\phi(x_i)^T\phi(x)$ denotes dot product, which is simply the elements of the kernel K. It seems all that's left is to calculate and normalize the $a^k{}_i$, which can be done by solving the eigenvector equation

$$N\lambda a = Ka \quad (7)$$

where N is the number of data points in the set, and λ and a are the eigenvalues and eigenvectors of K. Then to normalize the eigenvectors $N\lambda a = Ka$'s, we require that

$$1 = (V^k)^T V^k \quad (8)$$

Care must be taken regarding the fact that, whether or not has zero-mean in its original space, it is not guaranteed to be centered in the feature space (which we never compute explicitly). Since centered data is required to perform an effective principal component analysis, we centralize' K to become

$$K' = K - 1_N K - K 1_N + 1_N K 1_N \quad (9)$$

here 1_N denotes a N-by-N matrix for which each element takes value $1/N$. We use K' to perform the kernel PCA algorithm described above.

One caveat of kernel PCA should be illustrated here. In linear PCA, we can use the eigenvalues to rank the eigenvectors based on how much of the variation of the data is captured by each principal component. This is useful for data dimensionality reduction and it could also be applied to KPCA. However, in practice there are cases that all variations of the data are same. This is typically caused by a wrong choice of kernel scale.

A. Kernel functions and the kernel trick

The basic idea to deal with linearly inseparable data is to project it onto a higher dimensional space where it becomes linearly separable. Let us call this nonlinear mapping function ϕ so that the mapping of a sample $\mathbf{x}$ can be written as $\mathbf{x} \rightarrow \phi(\mathbf{x})$, which is called "kernel function."
Now, the term "kernel" describes a function that calculates the dot product of the images of the samples $\mathbf{x}$ under ϕ

$$\kappa(\mathbf{x_i},\mathbf{x_j}) = \phi(\mathbf{x_i})\phi(\mathbf{x_j})^T \quad (10)$$

In other words, the function ϕ maps the original d-dimensional features into a larger, k-dimensional feature space by creating nononlinear combinations of the original features. For example, if $\mathbf{x}$ consists of 2 features:

$$\mathbf{x} = [x_1 \quad x_2]^T \qquad \mathbf{x} \in IR^d \quad (11)$$
$$\Downarrow \phi$$
$$\mathbf{x}' = [x_1 \quad x_2 \quad x_1x_2 \quad x_1^2 \quad x_1x_2^3 \quad \ldots]^T \qquad \mathbf{x} \in IR^k \; (k >> d)$$

Often, the mathematical definition of the RBF kernel is written and implemented as

$$\kappa(\mathbf{x_i},\mathbf{x_j}) = exp(-\gamma \| \mathbf{x_i} - \mathbf{x_j} \|_2^2) \quad (12)$$

where $\gamma = \frac{1}{2\sigma^2}$ is a free parameter that is to be optimized.

B. Gaussian radial basis function (RBF) Kernel PCA

In the linear PCA approach, we are interested in the principal components that maximize the variance in the dataset. This is done by extracting the eigenvectors (principle components) that correspond to the largest eigenvalues based on the covariance matrix:

$$Cov = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x_i}\mathbf{x_i}^T \quad (13)$$

Bernhard Scholkopf (Kernel Principal Component Analysis [6]) generalized this approach for data that was mapped onto the higher dimensional space via a kernel function

$$Cov = \frac{1}{N}\sum_{i=1}^{N} \phi(\mathbf{x_i})\phi(\mathbf{x_i})^T \quad (14)$$

However, in practice the the covariance matrix in the higher dimensional space is not calculated explicitly (kernel trick).

Therefore, the implementation of RBF kernel PCA does not yield the principal component axes (in contrast to the standard PCA), but the obtained eigenvectors can be understood as projections of the data onto the principal components

C. implementing the RBF kernel PCA step-by-step

In order to implement the RBF kernel PCA we just need to consider the following two steps.
1. Computation of the kernel (similarity) matrix. In this first step, we need to calculate

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = exp(-\gamma \| \mathbf{x}_i - \mathbf{x}_j \|_2^2) \qquad (15)$$

for every pair of points. E.g., if we have a dataset of 100 samples, this step would result in a symmetric 100x100 kernel matrix.

2. Eigendecomposition of the kernel matrix.

Since it is not guaranteed that the kernel matrix is centered, we can apply the following equation to do so:

$$K' = K - \mathbf{1}_N K - K\mathbf{1}_N + \mathbf{1}_N K \mathbf{1}_N \qquad (16)$$

where $\mathbf{1}_N$ is (like the kernel matrix) a $N \times N$ matrix with all values equal to $1/N$. [7]

Now, we have to obtain the eigenvectors of the centered kernel matrix that correspond to the largest eigenvalues. Those eigenvectors are the data points already projected onto the respective principal components. figure 15 shows the elbow diagram for the optimal k. [8].

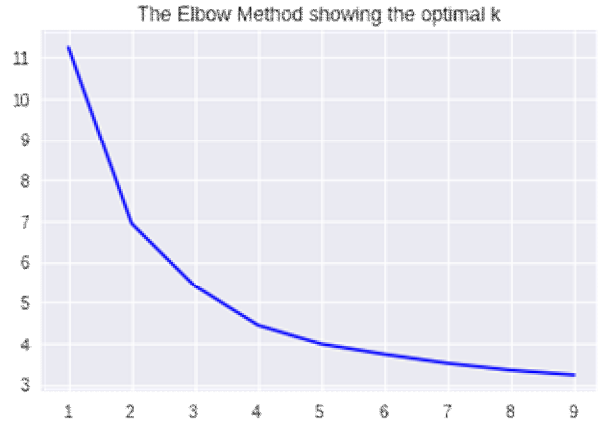

Figure 15 : elbow diagram for the optimal k

We also compare the result in PCA and KPCA. Figure 16 shows the data in real space and projected feathers by PCA and KPCA.

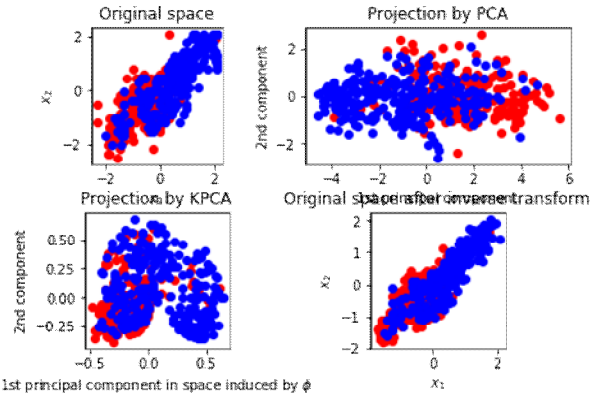

Figure 16:The real and projected observation by PCA and KPCA.

The up-left picture shows the data in the real space for the first components and the right is its projection in the first two features. As we can see the variance for the PC1 is more and the points are separated a lot more.
The bottom left shows the projection of the data by KPCA. As we can see the data are not so separated. As we can see, for our data PCA works better and as a result out data has the linea. The the last picture it is the revers of components in the original space. We also can find that by ignoring trivial factors in PCA or KPCA we do not lose too much information,

V. LOGISTIC REGRESSION

The goal of a supervised learning algorithm is to train a model using n observations and try to distinguish between m different classes. the input of the model is a vector of length d as follows:

$$x = [x_1, ..., x_d]^\dagger \in R^d \qquad (17)$$

Class label y is defined as follows:

$$x = [y^{(1)}, y^{(2)}, ..., y^{(m)}]^\dagger \qquad (18)$$

such that $y^{(i)} = 1$ if x belongs to i and $y^{(i)} = 0$ otherwise. Thus the n training samples are presented as:

$$D = (x_1; y_1); :::; (x_n; y_n): \qquad (19)$$

One of the well-known supervised learning methods is regression. Regression analysis helps to understand the relationship between a dependent and several independent variables. In other words, it shows how much the dependent variable changes according to the variations in the independent variables. The general regression model is applicable when the class attribute is continues, however, in the case of binary classes, we can use logistic regression. Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured with a binary variable (in which there are only two possible

outcomes). The goal of logistic regression is to find the best fitting model to describe the relationship between the binary characteristic of interest (dependent variable) and a set of independent variables. Multinomial logistic regression is a method that generalizes logistic regression to multi-class problems. In other words, it predicts the probabilities of different outcomes, and it is applicable when the outcomes are nominal [9]. Based on a multinomial logistic regression model, the probability that x belongs to class i is described as:

$$P(y^{(i)}=1\left\|x,\omega\right)=\frac{\exp(\omega^{(i)^T}x)}{\sum_{j=1}^{m}\exp(\omega^{(j)^T}x)},$$

for $i \in \{1,...,m\}$ (20)

in this equation, $w^{(i)}$ is the weight vector corresponding to class i. for the For binary problems, this equation is known as logistic regression. However, when we have more than two class labels it is called multinomial logistic regression. Due to the normalization problem:

$$\sum_{i=1}^{m}P(y^{(i)}=1|x,w)=1 \quad (21)$$

According to this equation, the weight vector for one of the classes need not to be calculated, because it can estimated using other weight vectors. The training data D is used to estimate the components of w, using the maximum likelihood estimator (MLE), as follows

$$l(\omega)=\sum_{j=1}^{n}\log p(y_j\left|x_j,\omega\right)=\sum_{j=1}^{n}[\sum_{i=m}^{n}y_j^{(i)}\omega^{(i)^T}x_j$$
$$-\log\sum_{i=1}^{m}\exp(w^{(i)^T}x_j)] \quad (22)$$

which is done using Newton's method or other maximization methods. In the rest of this section, we are going to use the logistic regression for the graduation admission classification problem. We are going to use both the features extracted using PCA, and all the features..

The admission data set consists of 400 observations. These observations are divided into the training and testing data set. Training part includes 80% of observations. The rest of the observations are used in testing part. The results we report in this section are obtained from the test data set. we have fitted the logistic regression model, and predicted the labels of testing data. Figure 17 shows the boundaries obtained from this model, and how much it has predicted the correct labels..

Figure 17:Logestic Regression for Test Set

As we can see, there are some green points in in the red space and some red points in the green part. These are defects. Now we should train out data to decrease the defects. After some training we have the result in the figure 18

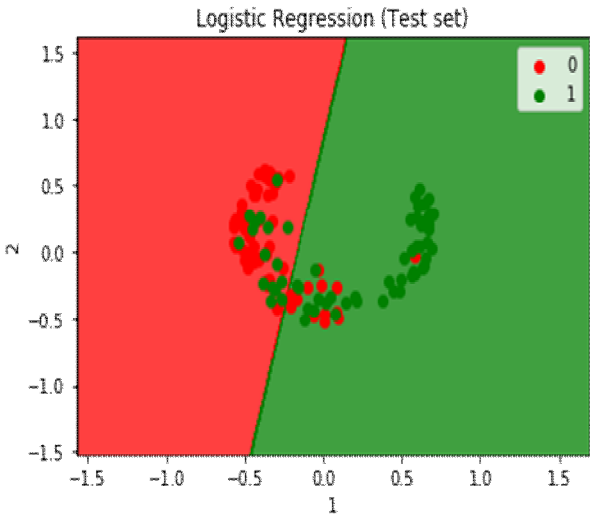

Figure 18: Logestic Regression for Training Set

We can see the now our model is better and we have less defects.

Another useful criteria to compare two models, is using the confusion matrix, which allows the visualization of the performance of an algorithm. Columns of this matrix are the predicted classes, and rows the are actual classes [10]. Confusion matrices calculated for the two cases are shown in figure 19.

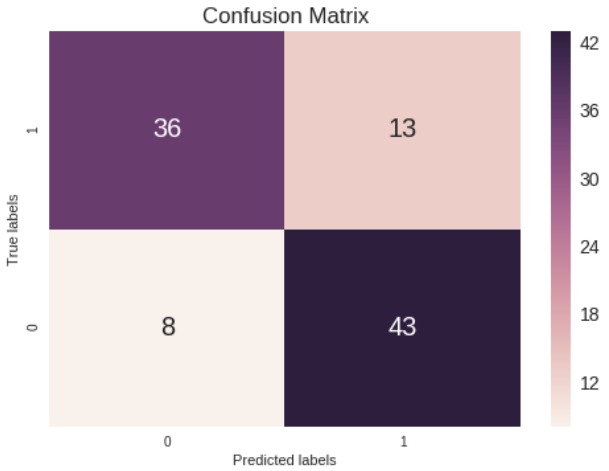

Figure 19:confusion matrix

As we can see from 49 on first object 36 are predicted correct and from the 51 of the second component, we found that 43 is predicted correct. So our training was good and now our model can predict more that 50 percent.

VI. V. CONCLUSION

In this work, we have used the university admission data of 400 students to predict the probability if acceptance of students given key factors.Our work has three phases. In the first phase, we have adopted PCA to extract important features from the university admission data set. Second, we have used Gaussian RBF kernel PCA to figure out the nature of the data. we have used logistic regression for classifying the observations. We have examined the performance of the logistic regression by considering two cases. Our results show that our data is linear. And the research is the factor that contributes to the variance more than the others.

.

[1] https://www.kaggle.com/datasets

[2] https://en.wikipedia.org/wiki/Undergraduate_educationhttps://en.wikipedia.org/wiki/Principal_component_analysis

[3] "Logisticregression," https://en.wikipedia.org/wiki/Logistic_regression.

[4] Herve Abdi , Lynne J. Williams, "Principal Component Analysis," Wiley Interdisciplinary Reviews: Computational Statistics, 2010.

[5] https://en.wikipedia.org/wiki/Kernel_trick

[6] https://dl.acm.org/citation.cfm?id=299113

[7] https://sebastianraschka.com/Articles/2014_kernel_pca.html#References

[8] https://en.wikipedia.org/wiki/Kernel_principal_component_analysis

[9] Balaji Krishnapuram, Lawrence Carin, "Sparse Multinomial Logistic Regression: Fast Algorithms and Generalization Bounds," IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 2005.

[10] "Confusionmatrix," tps://en.wikipedia.org/wiki/Confusion_matrixM. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.