Aalto University
Term Project Report

Application of Machine Learning Methods to Analyze and Predict Data

Author:

Muhammad Rafay Khan - 599838

Syed Azeem Akhtar - 545400

Professor:

Alex Jung

This term project report is submitted in fulfilment of the requirements for the course of Machine Learning: Basic Principles

November 25, 2016

Abstract

In this report, we discuss the key concepts, methods and learning related how we can predict data from a given training dataset using classification and regression methods. We have used the YELP dataset here with 50 features to classify if the user gave more than 3 stars to a certain restaurant (Classification) and if the reviews provided by the user were useful to other people. In order to test our prediction results the predicted values were submitted on kaggle to be verified against the actual dataset. In the end we have discussed the learning based on the findings and some improvements.

1. Introduction

In this project, the task is to classify if the user gave 3 or more stars to a restaurant based on the words used in the review text provided by the user. There are 50 words picked as features where classification and regression are both dependent on the occurrences of these words in the review text. We are provided with 5000 samples with 50 words (features), classified as 1 if the rating is 3 or more and 0 if it is less also the number of useful votes received used as output label for data. Machine learning methods used for the learning of the dataset is then applied to the additional test data of 1000 samples.

The training dataset is analyzed and extract important characteristics using visualization tools and analysis of features. After that different machine learning methods are applied to prediction. Regression methods are used to calculate the errors.

2. Methods

There are different machine learning methods which are suitable different types of dataset and the accuracy of the model can vary with respect to the dataset. Therefore, it is always a good practice to first understanding the dataset. Although it can be quite difficult to understand a dataset with a large number of samples but visualization of data is one of the ways which can help to in understanding the data with a graphical representation of the data. Initial analysis of data was done by using scatter plots, correlation plots and box plots. Plotting the correlation plot of the data showed that there is very low or no linear correlation between the features. Some of the inputs were difficult to classify because of the difference in the frequency of the occurrence of the words. This lead to the process of feature engineering on the dataset.

Principle Component Analysis (PCA) was applied on the normalized dataset through which a new dataset with reduced dimensions was achieved based on the most important principle components of the dataset. But it was found out that PCA was not much effective because of the reason that the data doesn't follow the normal distribution. Other feature selection techniques such as forward feature selection and backward feature selection were also applied to select only the important features from the dataset.

2.1. Visualization of Data

The main goal of data visualization is to portray the data in such a form which is easily understandable which can be effectively and clearly shown through graphical means. Some of the techniques used for the visualization of the data are briefly explained below

2.1.1. Scatter Plot

A scatter plot shows a relationship between two variables. Scatter plots can be combined in case of multivariate data. Such scatter plots are called matrix scatterplots. This approach can help show relation between covariates and target labels.

2.1.2. Histogram

Histograms are a single dimensional graphical representation of data where separation of the bins is not clear and is also used to estimate the probability distribution of the data.

2.1.3. Correlation Plot

Correlation plot are used to show the correlation between the features of a data. It helps find out the negative and positive correlation between the features. Computational cost can be reduced by removing the highly correlated features from the dataset.

2.2. Feature Engineering

Feature engineering is a domain of Applied Machine Learning which is a process of creating features which makes machine learning algorithms work. This process can be difficult and complex.

2.2.1. Principal Component Analysis (PCA)

PCA is a one of the core concepts of statistics which uses orthogonal transformation to convert of correlated variables into a set of linearly uncorrelated variables. These variables are called Principal components. Principal components could less or equal to the features of the original dataset. PCA is the most commonly used tool for the exploratory data analysis.

2.2.2. Zero-Mean Standardization of Data

Standardization of data is a process to convert data into a common format that helps in the application of machine learning methods. One of the process is to transform data into zero mean and unit variance.

$$x_{new} = (x - \mu) / \sigma$$

2.3. Regression

Regression is a method of modelling relationships between variables that is iteratively refined using measure of error in the predictions. For this problem linear regression has been used to predict new data.

2.3.1. Linear Regression

In linear regression, there are two variables one is predictor variable and the other is criterion variable. We use the predictor variable to predict the values of the criterion variable. If the criterion variable is target matrix and predictor variable is a covariate matrix we can say that

$$Y = WX$$

Where W is the weight vector which is obtained by the pseudo inverse technique, which gives

$$W = (XX)^{-1} X^T Y$$

2.3.2. Polynomial Regression

Polynomial regression is a machine learning regression technique where both dependent and independent variable as based on the degree of the polynomial.

2.4. Classification

Classification in statistics and machine learning is referred to as a problem of identifying a category (Class) for a new data observation with respect to the training dataset available. For this problem we have a testing dataset (new data) and we have to predict for a single observation if the user has given a rating of 3 or more to a restaurant.

2.4.1. Logistic Regression

Logistic regression is the most commonly used machine learning technique for classification problems. Logistic regression is a predictive analysis technique for a dataset where dependent variables are binary. It shows the relationship between the dependent variables and one or more independent variables.

In the learning phase of this model we select an optimum weight vector which maximizes the conditional log likelihood

$$\log P(\{r^t\} \mid \{x^t\}; w) = \sum (r^t \log y^t + (1 - r^t) \log (1 - y^t))$$

where $y^t := \text{sigmoid}\ (w^T x^t)$

We calculate the cost with the help of cost function and choose the weight vector which minimizes the cost

$$E(w| D) := -\sum (r^t \log y^t + (1 - r^t) \log (1 - y^t))$$

There is no analytical solution in general to this. Logistic regression amounts to solving

$$W_{opt} = \text{argmin } E(w \mid D)$$

In a high dimensional regime if $N << d$ where there is a high chance of overfitting we use the technique of regularization for the logistic regression method. And instead of E(w|D) we solve for

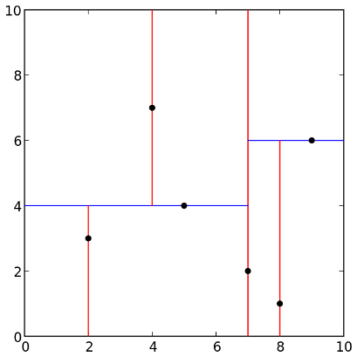
$$\text{argmin } E(w \mid D) + \text{Alpha } R(w)$$

2.4.2. KD-Trees

KD-Trees or Decision trees is a non-parametric method of regression and classification. As with any non-parametric method the main concept of KD-Trees is that similar data yields similar output. We divide the data into regions based on different decision boundaries. The decision of dividing a region into sub regions is taken based on the value of gain achieved. We keep dividing the regions in to sub regions until we find that the gain we are getting after the division is not that much or we reach a threshold value of max number of leaf nodes in a region.

$$\text{Gain}(m \rightarrow j) := \underbrace{\mathcal{H}(\mathcal{D}_m)}_{\text{entropy of leaf } m} - \underbrace{\left(\frac{|\mathcal{D}_{m,y}|}{|\mathcal{D}_m|} \mathcal{H}(\mathcal{D}_{m,y}) + \frac{|\mathcal{D}_{m,n}|}{|\mathcal{D}_m|} \mathcal{H}(\mathcal{D}_{m,n}) \right)}_{\text{entropy of leaf nodes } m_y \text{ and } m_n}$$

We select a value of decision boundaries (w) by cross validation on generalization error.



KD-Trees on univariate data

2.5. Model Assessment

Model assessment directly effects the accuracy of the model which is used for the learning of the dataset and predictions from that data. One of the main factors of model assessment is to minimize the generalization error on the new data. There could several predictor functions which are used for the prediction. If the model is extremely simple then the error will be high and the data will under-fit and if the model is too complex then the training data tends of over-fit. The error is high when the data is under-fitting and the error is nominal if the data if over-fitting. To overcome this problem model assessment techniques are used in obtain such model which is suitable for the future data. Some of the techniques are validation, cross-validation, bias and variance etc. In this problem cross-validation technique is used to tune the model to fit suitably.

2.5.1. K-fold Cross Validation

K-fold cross validation is one of the processes for the estimation of the performance of the model. In this method the training dataset is divided into K subsets, these are called k-folds. Where K-1 subsets are used for training and one of the k folds is used as a testing set. Models is learned on using the training set and then testing set is used to obtain the validation error. This processes is done for K iterations. Finally, the average error for all the k folds is computed.

2.5.2. Leave One Out Validation

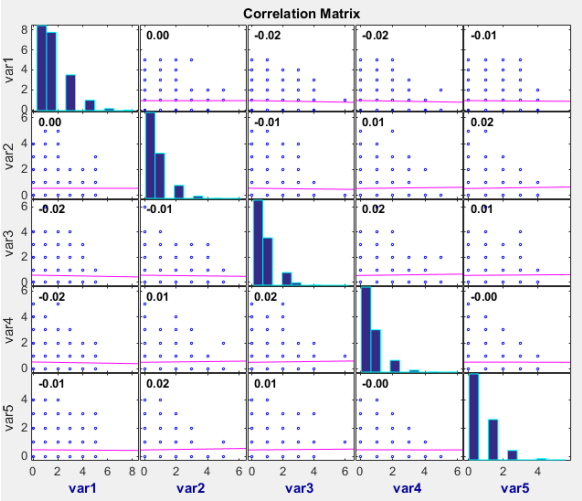
This technique is similar to K-fold cross validation but the main difference between the two techniques is that in k-fold validation the testing data split can contain several data points and each fold contains same number of data points. Whereas, in LOO only one data point which is used for testing. This could be repeated for N number of times.

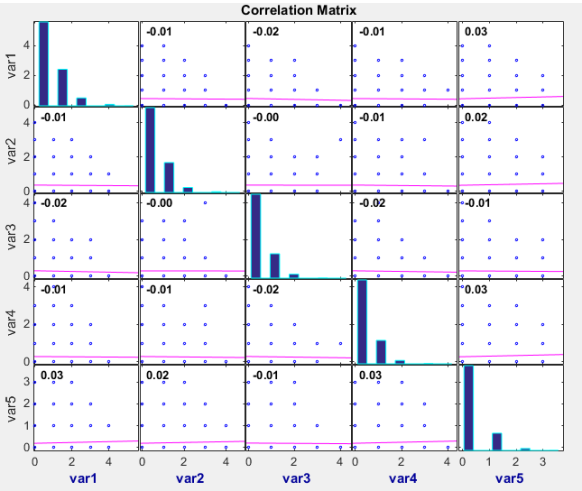
3. Experiments

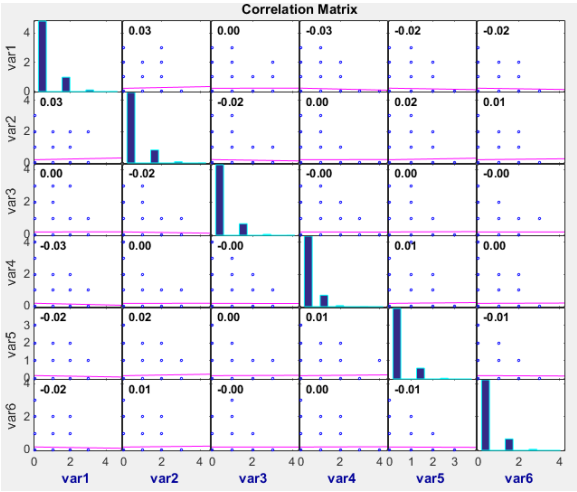
3.1. Exploratory Data Analysis

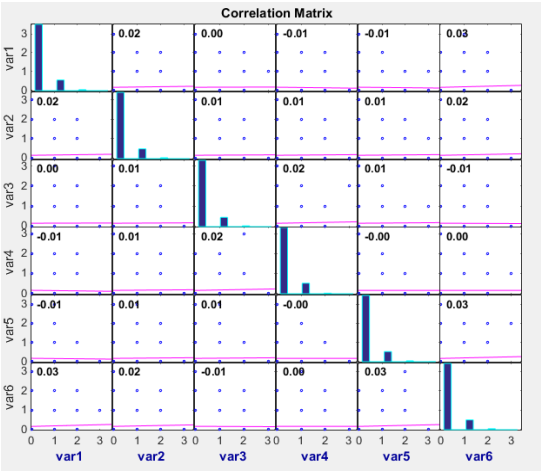
3.1.1. Correlation Matrix Plot

Here we have shown plotted correlation matrix plot for some of the features.









In the above shown figures we can easily see that the features are not linearly correlated with each other.

3.2. Feature Extraction

3.2.1. Principal Component Analysis

Principal component analysis was applied in order to reduce the dimensions of the data. It also plays a vital role in the extraction of features, extracting only the important features from the original data. Below are some figures which shows the importance of the features with respect to other features and how they affect the prediction. These figures are drawn using the PCA plot function in MATLAB named mapcaplot (). Principle components have also been calculated and scatter plots are also plotted which are shown below.

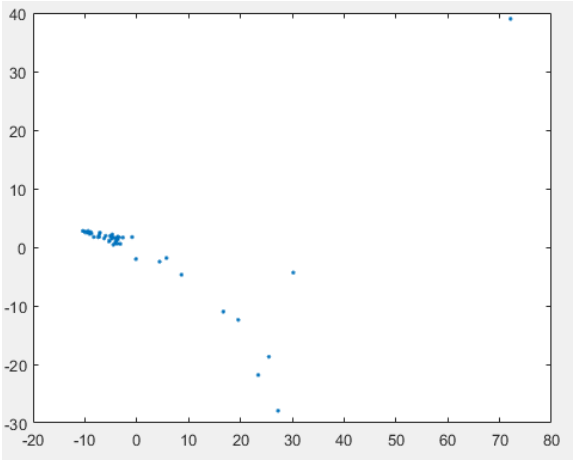


Fig: 3.2.1.1 Feature # 1 and 2

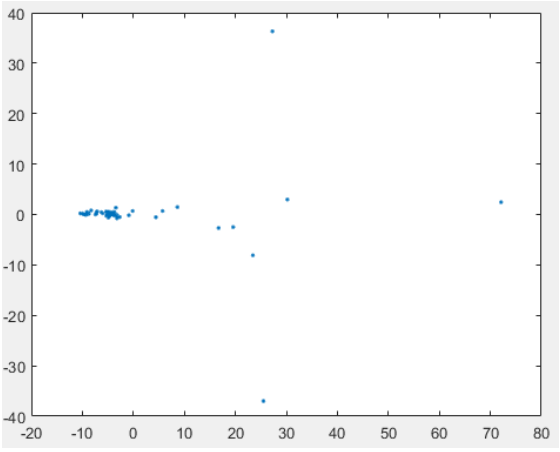


Fig: 3.2.1.2 Feature # 1 and 3

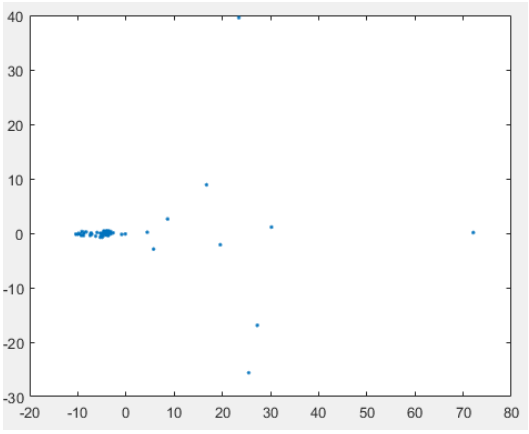


Fig: 3.2.1.3 Feature 1 and 4

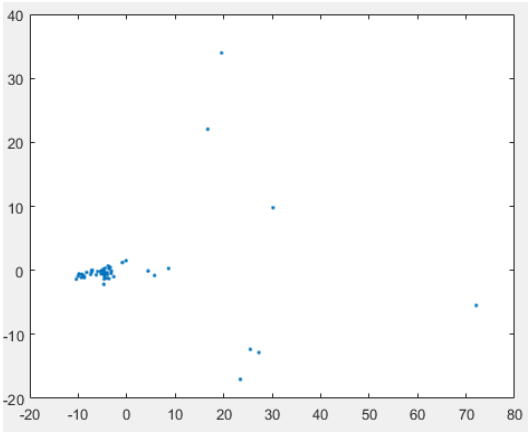


Fig: 3.2.1.4 Feature 1 and 5

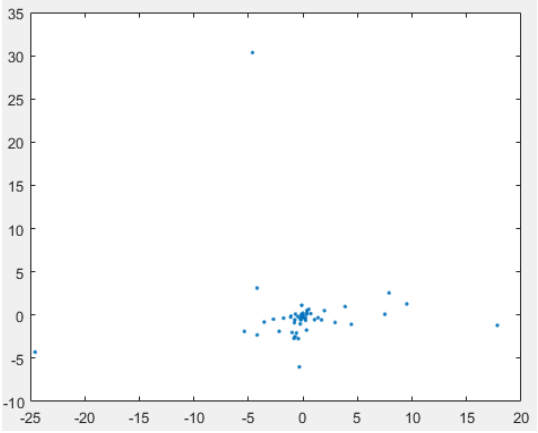


Fig: 3.2.1.5 Feature 10 and 11

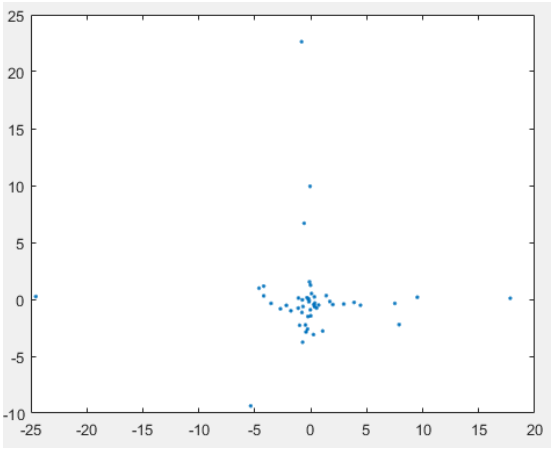


Fig: 3.2.1.6 Feature 10 and 15

The first four figures shows the relationship between feature 1 and 4 other features. Whereas, the remaining two figures shows the relationship between feature 10 with feature 11 and 15.

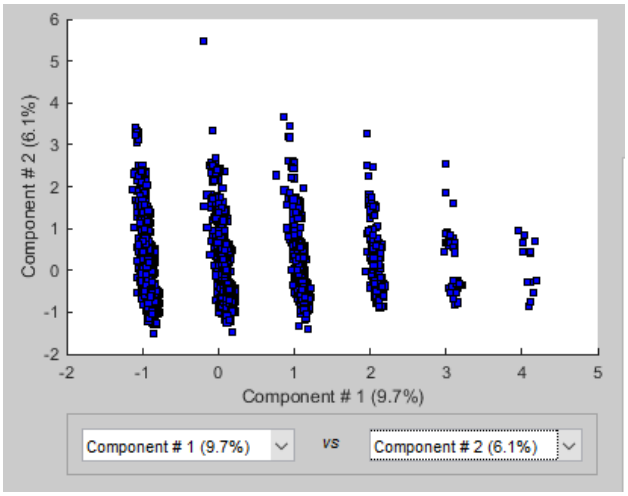


Fig: 3.2.1.7

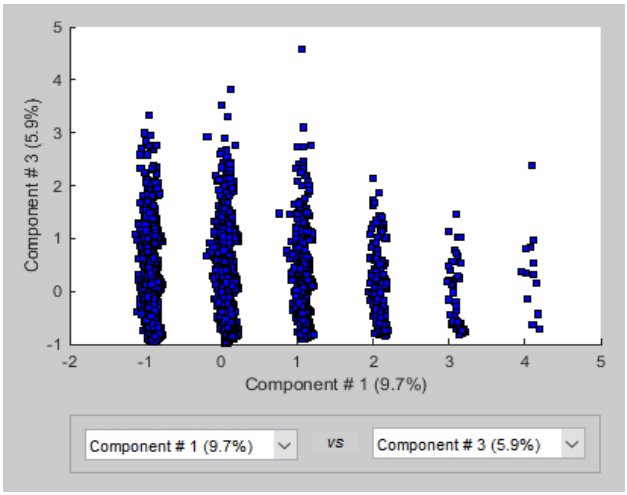


Fig: 3.2.1.8

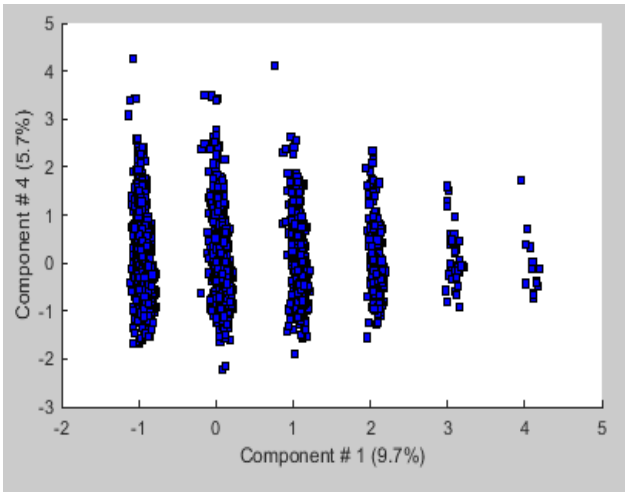


Fig: 3.2.1.9

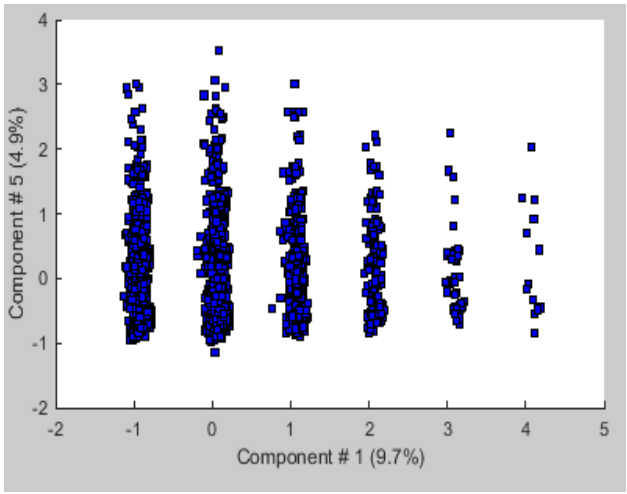


Fig: 3.2.1.10

Looking at the above four figures we can easily tell that feature 1 is the most important principal component with the highest percentage i.e. 9.7. But the problem with the data is that it doesn't follow the Gaussian distribution due to which applying the PCA doesn't affect the accuracy of predictions that much.

3.3. Application of Machine Learning Methods on Data

Different models were applied for classification and regression. Here we have only discussed the methods which gave the most accuracy for the training data.

3.3.1. Logistic Regression (Classification)

The data was split into training set and testing set using the k-fold cross validation. The logistic model was fitted to the training data to test the hypothesis regarding the relationship between the likelihood of the features and the rating got from the users. Logit function of Matlab is used to find the coefficients of the training data. The coefficients are selected at which the cost is minimum. These coefficient are then applied on the test data. Cross validations were also used in order to increase the efficiency of the model.

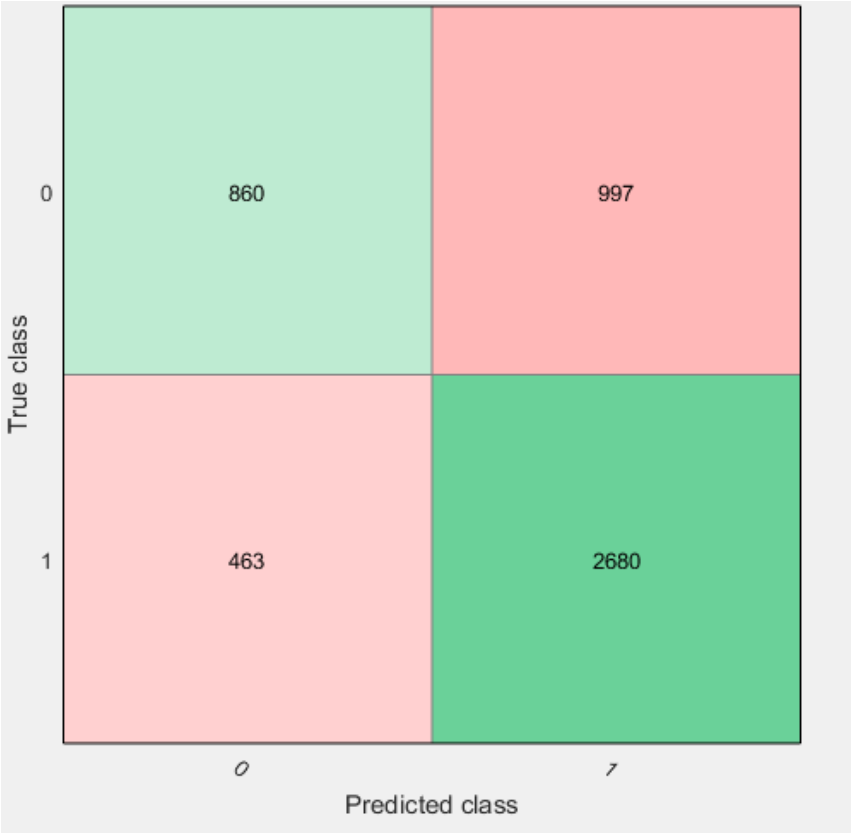


Fig 3.3.1.1 Confusion Matrix

The above Fig 3.3.1.1 shows the confusion matrix after the application of the model on the test data. The matrix shows the number of observations in each block showing the predicted class against the true class.

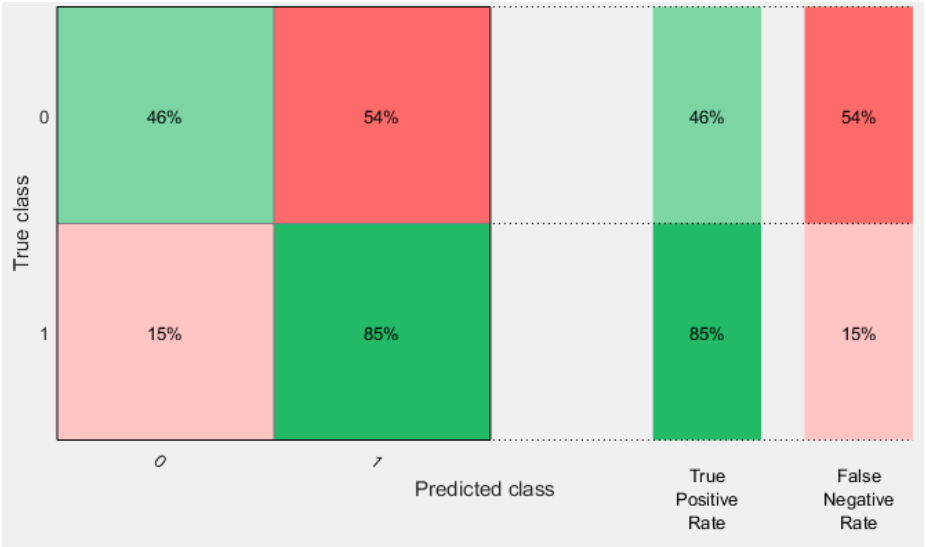


Fig 3.3.1.2

The above Fig 3.3.1.2 shows the distribution of true positive class rate and false negative class rate in terms of percentage of data.

Fig 3.2.1.3

The above Fig 3.2.1.3 represents the positive predicted value percentage and the false discovery rate.

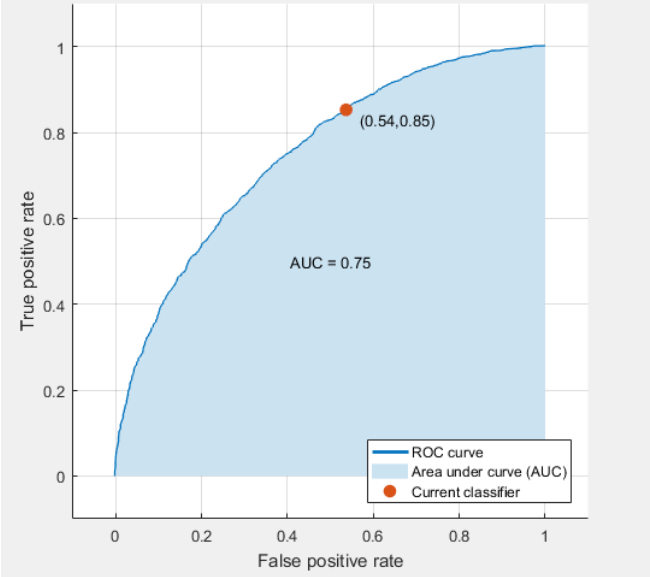


Fig 3.2.1.4 ROC Curve

The ROC Curve in the Fig 3.2.1.4 shows the curve between false positive rate and the true positive rate. It shows that the TP rate is 0.85 whereas the FP rate is 0.54.

3.3.2. Polynomial Curve Fitting (Regression)

Initially Multivariate curve fitting regression method was applied on out training data since it is the basic and the most commonly used method of regression. Weight vectors were calculated using the pseudo inverse technique as mentioned in the methods above using the polynomial of degree 1, selecting 30% data of training data as validation set and the rest as the testing set and the results were encouraging but the model based on polynomial of degree 1 was very simple for the training dataset. For this experiment we got the following results:

Empirical Accuracy: 89%
Validation Accuracy: 82%
Generalization Accuracy: 81%

Seeing the empirical accuracy to be not very high, it lead us to believe that the model is under-fit and we can use higher degree polynomial to increase the accuracy. Increasing the degree of polynomial increased the validation accuracy till degree 5 and started to decrease as the model started to over-fit.

3.3.3. KD-Trees

Once we make a decision tree based model for learning, we use it for learning the values associated to test data. If it is a classification task we assign the majority class inside one region. If it is a regression task we assign the mean value of all the values inside a region.

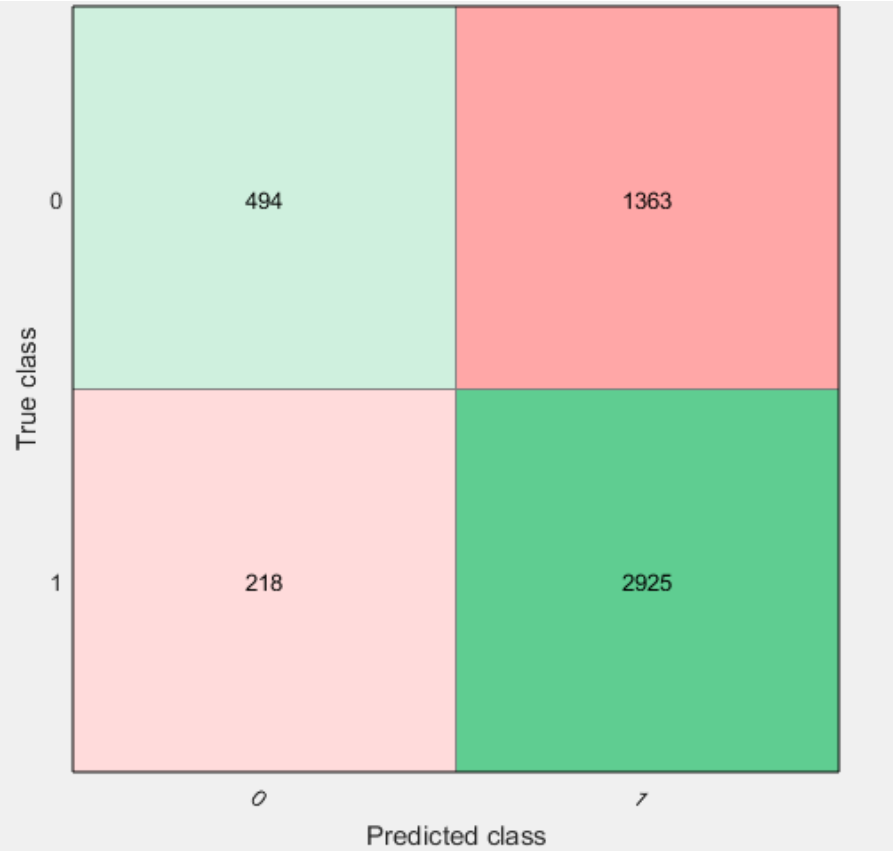


Fig 3.3.3.1

The figure 3.3.3.1 shows the confusion matrix for the decision tree classification of the training set with K-Fold cross validation of 10 sets.

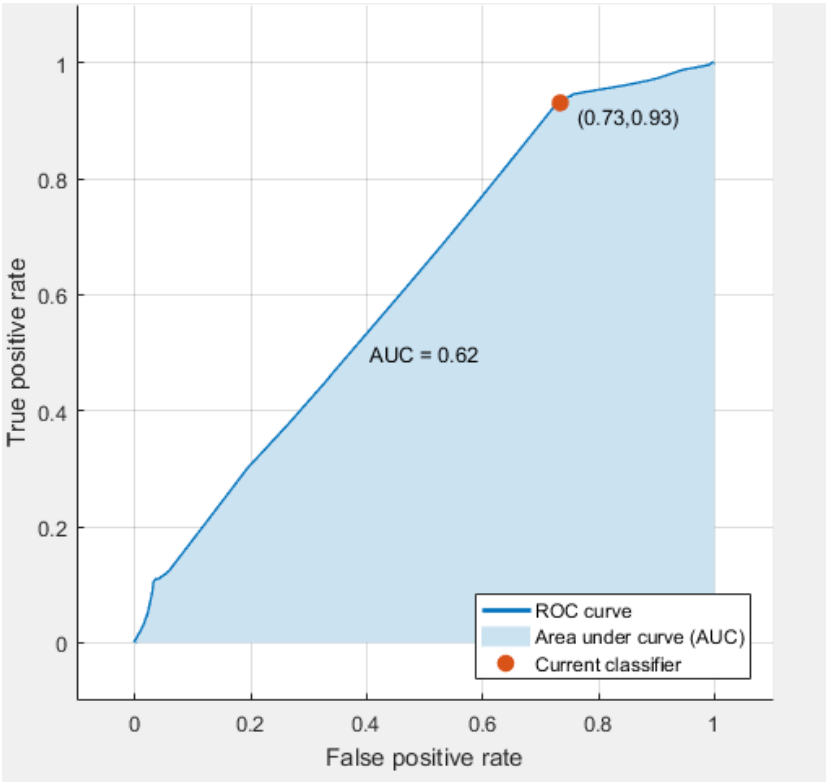


Fig 3.3.3.2 ROC Curve

The figure 3.3.3.2 shows the ROC curve for the decision tree classification. We can see the TP class prediction for classification is nearly accurate using this model i.e. the model has predicted 93% of the values correctly as positive.

4. Results

4.1. Logistic Regression

Initially Logistic regression was applied using all the features of the training dataset. 10 fold cross validation was also used to make the model efficient. After training the model it was applied on the testing dataset. The accuracy of model achieved from the data was 68.2% which shows that the model is not very efficient.

Then we applied PCA on out training set to reduce the features. We selected 42 features from our original data then applied the logistic regression method on the reduced data. The accuracy achieved from the reduced was 70.2 which was not that big of a difference from

the original dataset. This was because the data didn't follow the normal distribution and the features has no or very less correlation between them.

4.2. Polynomial Curve Fitting

We realized the fact that 30% data for validation is not the best of approaches for validation, we then changed the validation method to leave one out and then tried again with all the degrees. The best degree turned out to be 3 though there wasn't much difference in errors. We decided to use that as the final setting of parameters and calculated the final accuracy on that which turned out to be 96.13%.

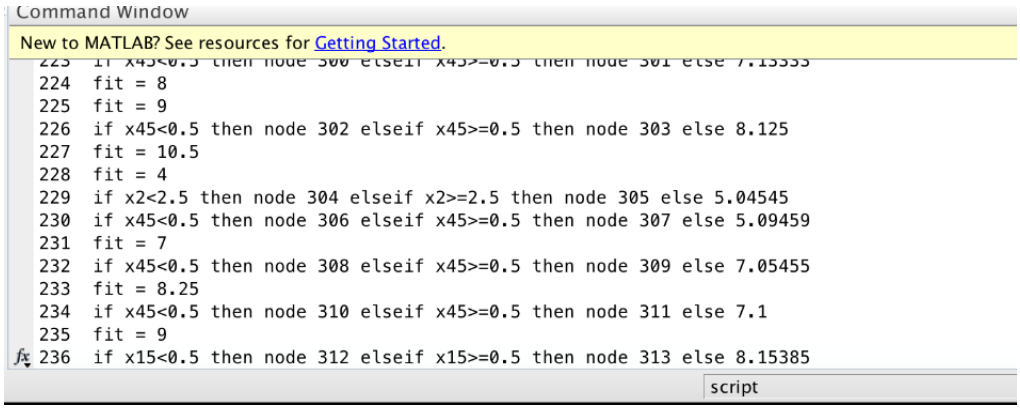
```
The validation mean squared error for degree1 is 237.9746
The validation mean squared error for degree2 is 196.9922
The validation mean squared error for degree3 is 188.1287
The validation mean squared error for degree4 is 376.9391
The validation mean squared error for degree5 is 2596.9998
The validation mean squared error for degree6 is 21897.5008
The validation mean squared error for degree7 is 170434.5576
The validation mean squared error for degree8 is 1254496.2656
The test mean squared error for degree 3 is 0.03867
Accuracy= 96.133
```

4.3. KD-Trees

We used MATLAB's own implementation of decision trees considering the fact that it provides a wide range of attributes and control over the parameters and validation options. The function we used was fitrtree for regression (See appendix for the full code). We first trained the tree without using any validation and the results we obtained was far from accurate (74% accuracy).

We then experimented with the different validation methods to minimize the generalization and the best we could get from it was 82% accuracy with KFolds cross validation.

Below is given a small view on the learned regression tree without validation.



5. Conclusion

Initially the base learner method was applied in which all the data points were assigned the class of '1'. The reason of assigning '1' as the base class was because the number of data points in the training set falling in this class was greater than the '0' class. The accuracy achieved after applying this method was only 62%.

5.1. Classification

For classification logistic regression gave us an accuracy of 69.2% when all the features were used. This model predicted better results compared to the base learners but still wasn't good enough to be called as the perfect model. After applying PCA and using the reduced data consisting of only the principal components produced and accuracy of 70.6% accuracy which was not much of a difference.

The second modelling algorithm used for classification was the decision trees which gave an accuracy of 70.8% percent accuracy. Hence, we cannot say that either one of the classification techniques used was perfect for this dataset. We can use either of the make predictions such dataset.

5.2. Regression

For regression multivariate polynomial curve fitting technique was used through which gave an accuracy of 89% using a first degree polynomial and using 30% of the data. Using this as base higher degree polynomials were used to achieve better accuracy and reduce generalization error. It was deduced that the third degree polynomial was best suited for the dataset along with the application of two different cross-validation techniques, K-fold and LOO. The accuracy achieved using this model gave an accuracy of 96.13%.

Decision trees were also used to train the data and the accuracy of 82% was achieved from this techniques along with a 10 fold cross-validation. Hence, we can easily conclude that multivariate polynomial curve fitting technique is most suitable for the regression dataset.

6. References

- Lecture Slides used in this course
- Introduction to Machine Learning, Ethem Alpaydin
- MATLAB Tools guide
- http://www.statisticssolutions.com/using-logistic-regression-in-research/

7. Appendices

7.1. Code (Multivariate Curve fitting):

```
clear;
training_data = csvread('RegressionTraining.csv',1,1);
errors = [];
feat=50;
[n, variates] = size(training_data);
variates = variates - 1;
for degree = 1:8;
    avError = 0;
    perm = randperm(n);
    for i = 1:n
        training_idx = perm([1:i-1 i+1:end]);
        test_idx = perm(i);
        trainingData = training_data(training_idx, 1:end-1);
        trainingData = trainingData(:, 1:feat);
        validationData = training_data(test_idx, 1:end-1);
        targets = training_data(training_idx,51);
        outputVal = training_data(test_idx,51);

        %%Find the PowerMatrix
        A = getBusted(feat,degree,n);
        repeatMat = repmat(trainingData, 1, degree+1);
        phi = repeatMat.^A;

        %% Find the weights by pseudoinverse technique
        weight = pinv(phi' * phi) * phi' * targets;
        %% Validation
        A = getBusted(feat,degree,2);
        phi_val = repmat(validationData,1,degree+1);
        phi_val = phi_val.^A;
```

```
        %% Find the output and the error
        output = phi_val * weight;
        errorTest = mean((outputVal - output).^2);
        avError = avError + errorTest;
    end;
avError = avError;
text = ['The validation mean squared error for degree',num2str(degree), ' is
',num2str(avError)];
disp(text)
errors = [errors avError];
end;
[m,i] = min(errors);
i = 3;
testingData = csvread('RegressionTesting.csv',1,1);
testOutput = csvread('RegressionSolution.csv',1,1);
A = getBusted(feat,i,n);
repeatMat = repmat(trainingData, 1, i+1);
phi = repeatMat.^A;
weight = pinv(phi' * phi) * phi' * targets;
A = getBusted(feat,i,1000+1);
phi = repmat(testingData,1,i+1);
phi = phi.^A;
output = phi * weight;
errorTest = mean((testOutput - output).^2);
text = ['The test mean squared error for degree ',num2str(i), ' is
',num2str(errorTest)];
disp(text);
disp(['Accuracy= ', num2str(100-errorTest *100)]);
```

7.2. Code (KD-Regression):

```
clear;
data = csvread('regression_dataset_training.csv',1,1);
```

```
trainingData = data(:, 1:50);
target = data(:, 51);
testingData = csvread('regression_dataset_testing.csv',1,1);
testOutput = csvread('regression_dataset_testing_solution.csv',1,1);
tree = fitrtree(trainingData,target, 'Leaveout','on');



view(tree);
Result(1:1000,1)=0;

for i = 1: 1000;
    xNew = testingData(i,:);
    y_hat = predict(tree.Trained{1}, xNew);
    Result(i,1) = y_hat;
end;
errorTest = mean((testOutput - Result).^2);
disp(num2str(errorTest))
```