CH5019 - Mathematical Foundations of Data Science
Term Project Report

Authors:
Mayur Vikas Joshi ME16B148
Rahul Chakwate AE16B005

May 10, 2019

CONTENTS

1 PROBLEM 01

1.1 AIM

To complete the data for number of tweets during 3 different seasons IPL in different parts of the country by filling the missing value using suitable algorithm.

- **Approach 1 for Data Imputation:** Principal Component Analysis method which involves using Singular Value Decomposition to obtain linear relationships between variables.
- **Approach 2 for Data Imputation:** Linear Regression on all the variables was applied and the model was trained on the available complete data and the incomplete missing data was filled.
- **Approach 3 for Data Imputation:** Average Method which fills the values of missing variables using average of the corresponding variable over entire data set.
- **Approach 4 for Data imputation** Case wise Average Method on each variable.
- **Approach 5 for Data Imputation** This uses Naive Bayes to classify/find out the missing values of Q1 and Q2 and uses one of the above mentioned methods for finding out the missing values of other variables.

Note: The imputed data for each approach and the code used has been enclosed along with the submission

1.2 DATA SET

The data set is collected from twitter posts across various regions of India for 3 different IPL seasons. Each data sample contains two binary variables and four integer variables.

1. 1. Q1 (first column): It is a binary variable indicating whether there exists a match for CSK on the particular day or not.
2. 2. Q2 (second column): It is a binary variable indicating whether there exists a match for MI on the particular day or not.
3. 3. X1 (third column): It is an integer value corresponds to the number of tweets specified David Warner on the particular day.
4. X2 (fourth column): It is an integer value corresponds to the number of tweets specified Mahendra Singh Dhoni on the particular day.
5. X3 (fifth column): It is an integer value corresponds to the number of tweets specified Rohit Sharma on the particular day.
6. X4 (sixth column): It is an integer value corresponds to the number of tweets specified Virat Kohli on the particular day.

1.3 Problem 1 Part (i)

Total Number of data samples available = 1000 The number of data samples which have at least one of the entries missing = 238. Therefore there are 238 rows in our data matrix which have one or more values missing.
So the number of data samples available for training the model = 762

1.4 Problem 1 Part (ii)

By omitting the samples with missing values **4 categories** can be observed which are:

1. Q1=0 and Q2=0: Neither CSK nor MI play the match.
2. Q1=0 and Q2=1: Only MI plays the match.
3. Q1=1 and Q2=0: Only CSK plays the match.
4. Q1=1 and Q2=1: Both CSK and MI play the match.

This division into 4 categories has been used in subsequent parts for data imputation.

1.5 Problem 1 Part (iii)

Linear relationships between the variables (X1, X2, X3 and X4) for each of the scenario is as follows:

The part of data set which had no missing values was split into different parts corresponding to each scenario. Singular value decomposition was performed for each case on the mean shifted data and the eigen vector corresponding to the lowest eigen value was computed. This Eigen vector corresponding to the least eigen vector gives the linear relation between the variables (X1,X2,X3,X4).
a. For the whole data

$$0.40303729X_1 + 0.38386888X_2 - 0.57008739X_3 - 0.60432275X_4 = 0$$

b. If there is no match for both CSK and MI

$$0.44360406X_1 + 0.63373043X_2 - 0.50697762X_3 - 0.38023001X_4 = 0$$

c. If there is no match for CSK but there is match for MI

$$0.16912257X_1 - 0.67650838X_2 + 0.23676674X_3 + 0.67651717X_4 = 0$$

d. If there is no match for MI but there is match for CSK

$$-0.22923771X_1 - 0.30565895X_2 + 0.80237288X_3 - 0.45849802X_4 = 0$$

e. If there is match for both MI and CSK.

$$-0.5334013X_1 - 0.41029914X_2 + 0.41030843X_3 + 0.61545484X_4 = 0$$

1.6 Problem 1 Part (iv)

1.6.1 Approach 1: Singular Value Decomposition

Approach 1 for Data Imputation: The first approach for data imputation is based on the principles of Principal Component Analysis.Principal Component Analysis method involves using Singular Value Decomposition to obtain linear relationships between variables and using these relations to fill missing values.

- In the first approach, the linear relations between different variables for different cases have been used to impute the missing data. These linear relationships between X1, X2, X3,X4 were found out by performing Singular Value Decomposition on the data set and using the eigen vector corresponding to the lowest eigen value.
- First the data samples for which only one value is missing are imputed. The data set with known values of Q1 and Q2 is considered and those samples with unknown Q1 or Q2 are removed, So we can classify the remaining data into one of the above 4 scenarios as information about which teams played the matches is known.
- Each of these is further divide into 4 categories which are: 1) Neither CSK nor MI is playing 2) Only MI plays 3) Only CSK play 4) Both CSK and MI play. So we have 16 cases which are imputed by using appropriate linear relation found using Singular Value Decomposition and computing the value corresponding to missing variable.
- Then data samples having only either Q1 or Q2 missing were imputed. If the value of X2 (Number of Dhoni Tweets) for a particular data sample is greater than the average number of Dhoni tweets (X2 value) for cases where only CSK played the match (i.e. Q1= 1 and Q2 = 0) then the missing Q1 value is set to be equal to 1 otherwise it is set to zero.
- Similarly the missing values of Q2 are filled by comparing the X3 value of incomplete data sample (Number of Rohit Sharma tweets) to the average value of X3 for complete data sample where Q1=0 and Q2=1 (Only MI plays the match). So if this X2 value for incomplete data is greater than the average X2 value of complete data for the case in which only MI play the match then Q2 is set equal to 1 else it is set equal to 0.
- Then the cases where more than one missing values present in the same data sample was considered. Here we observe in the data sample that there are only three cases: 1) X1 and X2 values are missing 2) X2 and X3 values are missing 3) X3 and X4 values are missing.
- Each of these cases is further subdivided into 4 cases on the basis of Q1 and Q2 values. Here since we have 2 missing values in each data sample we cannot use linear relation to find the missing value. We use the average values to corresponding to X1, X2, X3, X4 in each of the 12 cases to impute the data.

Enclosures:

- *'Imputed Data Approach1.csv'*

- *'Problem1_basic_code_and_SVD_method_ME16B148_AE16B005.py'*

The csv file contains completed data using approach 1. The .py file contains the basic code used for previous parts of the question and the Approach 1 code.

1.6.2 Approach 2: Linear Regression

Approach 2 for Data Imputation: Linear Regression on all the variables was applied and the model was trained on the available complete data and the incomplete missing data was filled.

- The training data comprises of 762 data samples over which the regression model is trained and wights are calculated.
- All the variables are included in this regression model. For example if we are imputing a data sample with value of X1 missing then our model trains over X2,X3,X4,Q1 and Q2.
- The model returns a continuous value for Q1 and Q2. We set the them equal to 1 if the value returned is greater than 0.5 and we set them equal to zero if the value returned is less than 0.5
- For those incomplete data samples where more than one values are missing, we fit a model using remaining variables. For example: If the values of X2 and X3 are missing in a particular data sample then we fit the model using X1, X4, Q1 and Q2.

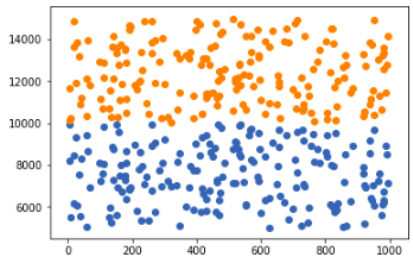

Figure 1: Dhoni Tweets and Rohit Sharma Tewets Vs. Index when only CSK plays

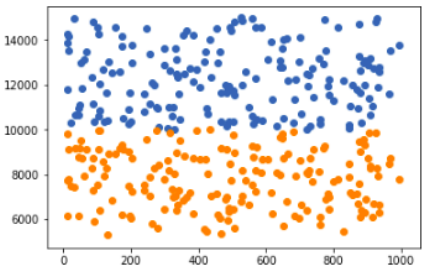

Figure 2: Dhoni Tweets and Rohit Sharma Tewets Vs. Index when only MI plays

Enclosures:

- *'Imputed Data Approach2.csv'*
- *'Problem1_linear_regression_method_ME16B148_AE16B005'*

The csv file contains completed data using approach 2. The .py file contains the Approach 2 code.

1.6.3 Approach 3: Average

Approach 3 for Data Imputation: Average Method which fills the values of missing variables using average of the corresponding variable over entire data set. This is a very simple and basic method of Data Imputation.

- We first consider a data frame consisting of 762 complete data samples and use these for calculating averages.
- We calculate the average value of X1 over the entire data this complete data set and fill where value of X1 is missing in the remaining 238 incomplete data samples.
- We repeat the same process for filling the missing values of X2, X3 and X4.
- This automatically takes care of the those cases which have more than one missing values.
- We fill the missing value of Q1 by comparing the value of X2 in that data sample to a certain threshold value. This threshold value was taken as the average value of X2 in complete data for cases in which CSK plays the match.
- Similarly the value of Q2 is filled by comparing the value of X3 in that data sample to a certain threshold value. This threshold value was taken as the average value of X3 in complete data for cases in which MI plays the match.

The imputed data for this method has not been enclosed as we have also implemented a modified version of this method which is superior in the following approaches.

1.6.4 Approach 4: Case-wise Average

Approach 4 for Data imputation Case wise Average Method on corresponding variables.This is a modified version of previous method as instead of using averages over entire data set, we use average over data which is more relevant for a particular missing value. This is done by dividing the data on the basis of values of Q1 and Q2.

- We divide the complete data comprising of 762 data samples into cases depending on values of Q1 and Q2. These cases are same the one given in question.

- We then classify the each data sample of the incomplete data set comprising of 238 data samples into one of the four cases depending on Q1 and Q2 values of incomplete data samples.
- Then similar to previous approach we fill the missing value of each variable (For example: X1) using the average of X1 corresponding to the case to which the given incomplete sample belongs depending on the values of Q1 and Q2.
- Similar procedure is followed to fill all the missing values corresponding to all the variables.

Enclosures:

- *'Imputed Data Approach 4 and 5.csv'*
- *'Problem1_average_method_and_naive_bayes_method_ME16B148_AE16B005'*

The csv file contains completed data using combined version of approach 4 and 5. The .py file contains the combined code for Approach 4 and 5.

1.6.5 Approach 5: Naive Bayes

Approach 5 for Data Imputation This uses Naive Bayes to classify/find out the missing values of Q1 and Q2 and uses one of the above mentioned methods for finding out the missing values of other variables.

- This is a modification of previous methods. Here we can use any method for predicting values of the variables X1, X2, X3, X4 and we use Naive Bayes Classifier to fill the missing values of Q1 and Q2.
- In the attached code, we have used Case Wise Average method for X1,X2,X3,X4 and Naive Bayes Classifier for Q1 and Q2.
- We implemented Naive Bayes classifier using the sklearn library from python. We trained the Gaussian Naive Bayes model on the 762 complete data samples.
- Then used the trained model to give predict the values of Q1 and Q2 for the incomplete data samples with missing Q1 and Q2 values.
- A training accuracy of 93.43% was obtained in case of of Q1 and 98.29% training accuracy was obtained for Q2.

2 Problem 02

2.1 Aim

The data matrix given contains data for credit fraud detection. The task is to classify the data/transaction as legitimate or fraudulent. Classification should be done using Support Vector Machine (SVM).

2.2 Data Set

q2_data_matrix.csv: This file contains a 100 x 5 data matrix. The 5 features and their corresponding ranges are described below:

1. Age: 18-100 years
2. Transaction Amount: $ 0-5000
3. Total Monthly Transactions: $ 0-50000
4. Annual Income: $ 30000-1000000
5. Gender: 0/1 (0 - Male, 1 - Female)

q2_labels.csv: This file contains a 1000ÃŮ1 vector of 0/1 labels for whether the transaction is fraudulent or not.

- 0: The transaction is legitimate
- 1: The transaction is fraudulent

2.3 Required Python Libraries

- numpy
- pandas
- sklearn.svm.SVC
- sklearn.metrics.confusion_matrix
- sklearn.metrics.f1_score

2.4 Pseudo Code and Methods

This section provides function-by-function pseudo code for classification using SVM and hyperparameter tuning.
SVM_func():

- Takes input as training data, labels, kernel name, regularization parameter and kernel parameter.
- It calls the sklearn.svm.SVC method by appropriately passing the required arguments for the given kernel. It fits the training data and labels.
- returns a classifier object.

best_hyperparams():

- Takes training data, labels, kernel name as input.

- This data is further split into train and validation sets.
- The hyperparameter tuning is done by grid search method.
- For all kernels, regularization parameters list is common.
- For polynomial kernel, the range of polynomial degrees is provided.
- For rbf kernel, the range of lambda value in the exponent is given.
- Computes the classifier, predicts on train and val sets.
- Parameters for which validation loss is least are returned

main():

- Full data is shuffled and split into train and test data sets.
- For different kernels, best hyperparameters are found out.
- Final classifier is trained using these best hyperparameters.
- Train, test loss and acuracies are reported.
- For Linear kernel, Confusion Matrix and F1 Scores are also computed.

2.5 Linear SVM

In Linear SVM, the data matrix is used directly as an input to the SVC function.

2.5.1 Results

The Confusion Matrix, F1 Score and Accuracy of Linear SVM are given below in the table.

Table 1: Confusion Matrix for Train Data

_	Legitimate	Fraudulent
Legitimate	93.1%	6.9%
Fraudulent	20.6%	79.4%

Table 2: Confusion Matrix for Test Data

_	Legitimate	Fraudulent
Legitimate	90.5%	9.5%
Fraudulent	17.8%	82.2%

Table 3: F1 Score for Train and Test Data

F1 Score Train	F1 Score Test
0.8414	0.8259

Table 4: Accuracy for Train and Test Data

Train Accuracy	Test Accuracy	Train Loss	Test Loss
88.23%	87.00%	0.1176	0.1300

2.5.2 Conclusion of the above results

- The Confusion Matrix shows that most of the data is classified correctly for train as well as test data.
- Legitimate case is classified more accurately than the Fraudulent case.
- Classification for train data is more accurate than for the test data due to some amount of overfitting. But due to proper hyperparameter optimization, this overfitting has reduced substantially.
- F1 Score for Train data is slightly greater than the test data
- Train Accuracy is slightly better than the test accuracy due to some amount of over fitting. Overall, about 87% of the test data is classified correctly.

2.6 Comparison of Linear SVM with Kernel SVM

2.6.1 Observations and Results

Linear Kernel:

Table 5: Accuracy, loss and best parameters for Linear Kernel

Train Accuracy	Test Accuracy	Train Loss	Test Loss	Best Regularization Parameter
88.23%	87.00%	0.1176	0.1300	1e-05

Polynomial Kernel:

Table 6: Accuracy, loss and best parameters for Polynomial Kernel

Train Accuracy	Test Accuracy	Train Loss	Test Loss	Best Reg Parameter	Best Degree Parameter
64.83%	58.00%	0.3517	0.4200	1e-05	4

RBF Kernel (Gaussian Kernel):

Table 7: Accuracy, loss and best parameters for RBF Kernel

Train Accuracy	Test Accuracy	Train Loss	Test Loss	Best Reg Parameter	Best Lambda Value
65.28%	59.45%	0.3451	0.4114	1e-05	0.001

For this dataset, Linear Kernel gives the best results.

2.6.2 CONCLUSION OF THE ABOVE RESULTS

- The Linear kernel SVM performs better classification than polynomial or Gaussian kernel SVM.
- This indicates than the train data has linear relationship and is almost linearly separable.
- RBF or Gaussian kernel performs slightly better than polynomial kernel.
- The best regularization parameter comes out to be 1e-05 for all kernels. This implies that minimum regularization is required to fit the data and there is no inherent over-fitting.