Machine Learning
(Programmng Assignment - 1)

Shubhangi Ghosh
EE15B129
Department of Electrical Engineering

August 31, 2017

1 INTRODUCTION:

This assignment presents python illustrations of Linear methods for classifcation and regression. This report summarises the concepts and ideas behind the implementation of th assignment.

2 Problem 1 - SYNTHETIC DATA CREATION:

2.1 PARAMETERS:

2.1.1 COVARIANCE MATRIX:

1. A $20x20$ matrix cov is randomly generated.
2. A positive-definite matrix cov_final is created from it multiplying it with its transpose.
3. The cov matrix is generated with all positive integers. So, cov_final has on-zero off-diagonal elements and is therefore not a diagonal matrix.
4. The same covariance matrix, cov_final is used for generating both normally distrbuted classes.
5. Average standard deviation dev is calculated by averaging the square roots of the diagonal elements(variances of the parameters).

2.1.2 CENTROIDS:

The centroids are placed at a distance of $0.4dev$ apart, so that there is some overlap between the two classes.

2.1.3 TRAINING AND TEST DATA

1. 2000 samples are generated for each class.
2. Data is split nto training and test data in 70-30 ratio, uniformly from each class.

3 Problem 2 - CLASSIFICATION USING LINEAR REGRESSION

1. Before exporting the training and test data into csv files, two columns of indicator variables have been appended.

(a) **Column 1:**
Holds 1 or 0 to indicate whether or not element belongs to Class 1.

(b) **Column 2:**
Holds 1 or 0 to indicate whether or not element belongs to Class 2.

2. Linear regression is performed on both these indicator variables. Thus, two lines, $\delta_1(X) = y_1$ and $\delta_2(X) = y_2$, are fit through these data points. The LinearRegession module from sklearn.linear_model has been used.

3. If $\delta_1(X) > \delta_2(X)$,the point is classified as belonging to Class 1, and otherwise is classified as belonging to Class 2. This is because the true value of $y_1$is 1, if point belongs to Class 1, and 0 if it doesn't. Similar comditions apply to y_2. $\hat{y}_i$lies in the range of 0 to 1; thus, the closest $\hat{y}_i$is to 1, the likelier the element is to belong to Class i.

4. $\delta_1(X) = \delta_2(X)$ is the separating hyperplane between the two classes.

5. In this question, since there are only two classes, we could actually make do with one column of indicator varibles, and checking if the value predicted is > 0.5.

6. The best-fit parameters are:

   (a) Accuracy: 75.83%

   (b) Precision: 76.63%

   (c) Recall: 74.33%

   (d) F-measure: 75.46%

7. The coefficients have been turned in a $coeffs.csv$ file, with the intercept being the first coefficient.

4 Problem 3 - k-NEAREST NEIGHBOUR CLASSIFICATION

1. The KNeighboursClassifier from sklean.neighbours has been used.

2. The best fit coefficients are:

NN	Accuracy	Precision	Recall	F-Measure
1	51.75%	51.78%	50.83%	51.30%
2	52.08%	54.15%	27.17%	36.18%
3	51.42%	52.40%	52.83%	52.61%
4	52.17%	53.49%	33.17%	40.95%

3. This algorithm performs worse than linear regression on indicator variables.

4. The algorithm for this particular dataset performs well on k=1, k=3. But in general, we can say that this algorithms performs well on k=1.

5 Problem 4 - COMPLETING AN INCOMPLETE DATASET

1. Dataset completion has been done using the Imputer module of sklearn.preprocessing.

2. The first five parameters have been discarded since they are said to be non-deterministic on the O/P variable.

3. Using the sample mean is a good choice, since, we are trying to minimise Mean Square Error. So we minimise Sum of Squared Errors, $\sum(x - x_i)^2$. On differentiating and equating to 0, we find that this is minimum, when x assumes mean value. Thus, incomplete data points are made to assume mean value of the corresponding parameter.

4. The other strategies involve replacing with meadian and most_frequent. But replacing with mean makes more sense, since mean square error can be minimised.

6 Problem 5 - LINEAR REGRESSION FITTING

1. Five 80-20 train-test data splits in CandC_complete.csv from question 4 are made using test_train_split module of sklearn.model_selection.

2. Linear Regression fitting is applied on all the five datasets and Reduced Sum of Squares Error is reported.

3.

RSS1	8.51
RSS2	6.31
RSS3	271.76
RSS4	317.69
RSS5	8.17

RSS values may change for every run of code.

4. Coefficients have been reported in csv files.

7 Problem 6 - LINEAR RIDGE REGRESSION FITTING

Ridge Regression doesn't implement feature selection as relative decrease in higher weights is larger compared to relative decrease in lower weights. Thus, once the weights are reduced to an extent, reducing them further is time-consuming.
Lasso regression can implement feature selection as the weights of irrelevant parameters are reduced to 0.