**Note: All assignments should be done in python only.**

**Exercise – 1:**

* Implement simple linear regression using gradient descent and apply it on ‘Datasets/ex1/ex1data1.txt’.
* Make your own individual python functions as per steps listed below and use them to produce best line fit on the dataset.
* Also, produce best line fit by using inbuilt library functions of “scikit-learn” for the linear regression and tune their parameters.

Tasks to perform:

1. Import necessary libraries
2. Read data
3. Plot data
4. Define and use function for cost computation
5. Define and use function for calculating gradient descent
6. Plot the linear model along with data to visually see how well it fits
7. Plot Error vs. Training epoch graph

Suggestions:

* Follow ex1.pdf for guidance to complete exercise.

Expected output:

1. Linear model should be best fit on data.
2. Exponentially decreasing Error vs. Training epoch graph.

**Exercise – 2:**

* Implement linear regression with multiple variables on ‘Datasets/ex2/ex1data2.txt’ for “House Price Prediction”.
* Find the cost (error) of the model near to 0.13070
* plot Error vs. Training epoch graph
* Code should be implemented in Python.

Expected output:

1. Cost (error) of the model near to 0.13070
2. Exponentially decreasing Error vs. Training epoch graph

**Exercise – 3:**

* Apply logistic regression on ‘Datasets/ex3/pima-indians-diabetes.csv’ to get at least 77% of classification accuracy.
* Code should be implemented in Python.

Suggestions:

* Use inbuilt logistic regression function of “scikit-learn” library.
* Apply data normalization if required.
* Use 10 fold cross validation to finalize classification accuracy

Expected output:

1. Classification accuracy >= 77%.
2. Classification accuracy (without data normalisation) < Classification accuracy (with data normalisation)

**Exercise – 4:**

* Apply KNN classifier on ‘Datasets/ex4/winequality-red.csv’ to get at least 50% of classification accuracy.
* Code should be implemented in Python.

Suggestions:

* Use inbuilt KNN function of “scikit-learn” library.
* Apply data normalization if required.
* Use 10 fold cross validation to finalize classification accuracy

Expected output:

1. Classification accuracy >= 50%

**Exercise – 5:**

Apply SVM classifier on datasets given in ‘Datasets/ex5/ex5data1.mat’ and ‘Datasets/ex5/ex5data2.mat’ to get at least 90% of classification accuracy.

* Code should be implemented in Python.

Suggestions:

* Use inbuilt SVM function of “scikit-learn” library.
* Apply data normalization if required.
* Use 10 fold cross validation to finalize classification accuracy

Expected output:

1. Classification accuracy >= 90%

**Exercise – 6:**

* Apply K-Means classifier on ‘Datasets/ex6/ex6data2.mat’.
* Use the same algorithm on ‘Datasets/ex6/bird\_small.png’ for Image Compression.
* Code should be implemented in Python.

Suggestions:

* Use built in K-means function of “scikit-learn” library.
* Follow K-means part of ex6.pdf for guidance to complete exercise.

Expected output:

1. K-means classifier must provide 3 Centroids (1.1535, 4.6786), (6.2737, 2.2425), (2.2096, 4.9146).
2. Compressed image should contain main features of original image.

**Exercise – 7:**

* Use YALE face database provided in folder ‘Datasets/ex7/centered/’. Dataset contains 15 persons with different facial expressions. 11 different images per persons are provided in the database.
* Use feature reduction technique (PCA or LDA) and any of the studied classifier to identify the person index from the trained classes with at least 70 % accuracy.
* Code should be implemented in Python.

Suggestions:

* Use inbuilt classification method and PCA/LDA functions of “scikit-learn” library.
* Find out the best classification algorithm which gives best classification accuracy.
* Use 10 fold cross validation to finalize classification accuracy.

Expected O/P:

1. Produce Eigen faces.
2. New image should be correctly identified with minimum 70% accuracy.