Muhammed Şamil Arınç
21803683

HOMEWORK 2 REPORT

Question 1:

1.1.

Results:

Let PCi denotes Principal Component i:

```
Proportional variance of PC1  is 0.283
Proportional variance of PC2  is 0.110
Proportional variance of PC3  is 0.098
Proportional variance of PC4  is 0.061
Proportional variance of PC5  is 0.032
Proportional variance of PC6  is 0.029
Proportional variance of PC7  is 0.021
Proportional variance of PC8  is 0.021
Proportional variance of PC9  is 0.018
Proportional variance of PC10 is 0.014
```

Analysis:

It can be seen in the above values that the PVE (proportion of variance explained) of each component is decreasing. It is reasonable since the Components' importance is decreasing from 1 to 10.

Plots:

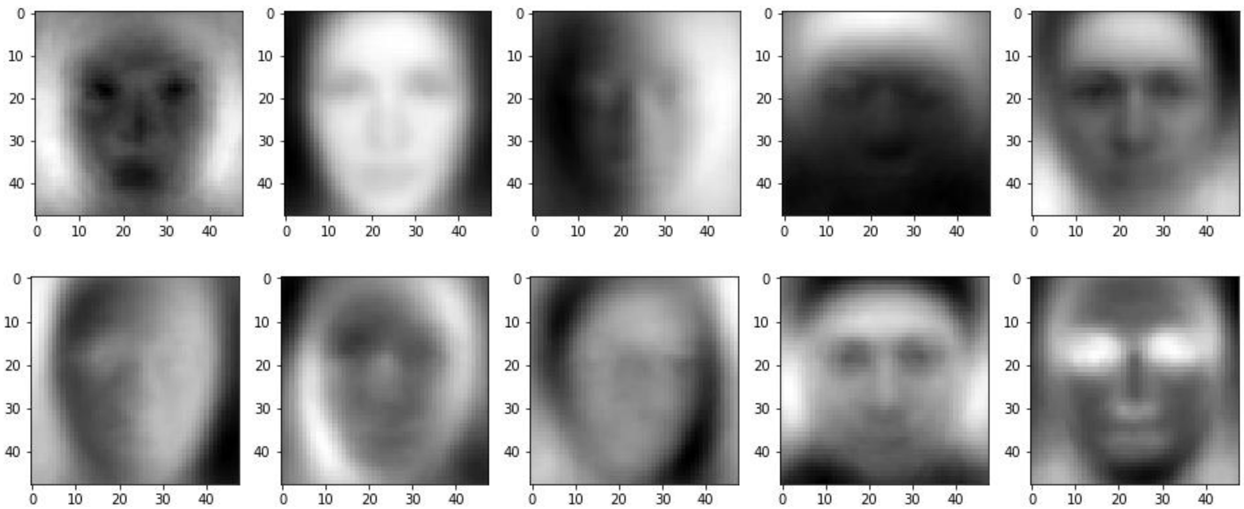

Fig.1: Eigenfaces generated

Analysis:

The data consists of 2304 features and I reshaped them to 48x48 matrices to plot them as image. The results that I obtained shows similarities with human faces. Also, interestingly the 2nd and 5th images seem like someone familiar to me. All the faces represented in the images focusing on some principal components of a human face such as the last image which focuses on the eyes of a human.

1.2.

Results:

```
When k=1,   PVE is 28.334
When k=10,  PVE is 68.708
When k=50,  PVE is 85.693
When k=100, PVE is 90.844
When k=500, PVE is 98.065
```

Plot:

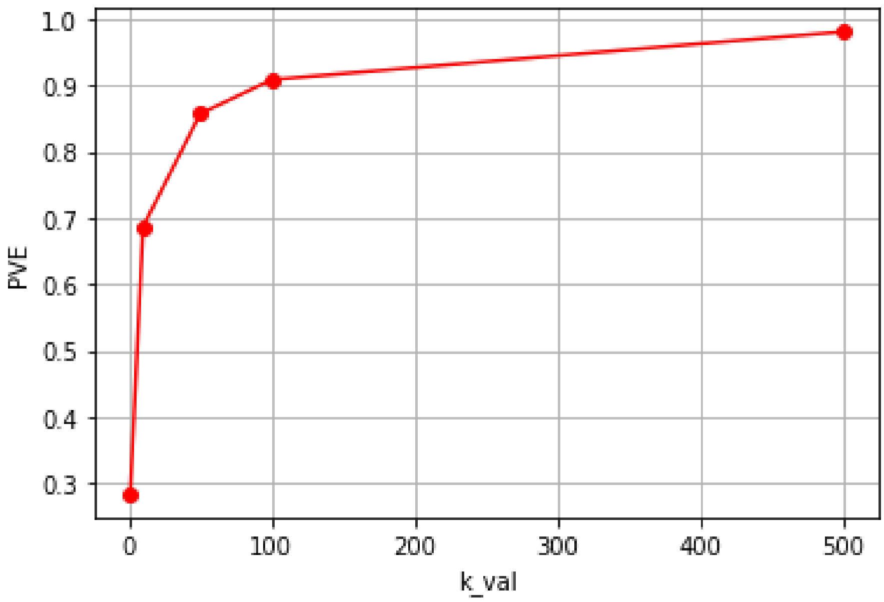

Fig. 2: k_val-PVE graph

Analysis:

As it can be seen in the above figure that increasing number of k, causes an increase in the PVE as well. That's because, the reconstructed images from the principal components that are calculated in Q1.1. There are 2304 principal components (same as the number of pixels) and when we use all of them to reconstruct an image, the image data will be totally recovered. However, as the HW2 manual asks, we used 500 of them at maximum and therefore, the PVE is 100% but very close to it. Its construction percentage is %98.26 which is very close to total construction. Hence, the data is zipped to nearly ¼ of its original size with losing just 1.8% of its data.

1.3.

Since we used PCA technique which constructs a new coordinate system whose basis is constructed with the calculated eigenvectors. Hence, in order to reconstruct the raw images from the data, the projection of the data on the new coordinate system that has been constructed should be found. After that the mean of the data will be readded to data in order to de-centralize it. To obtain the raw images following formula is used:

Muhammed Şamil Arınç
21803683

$$(X * W)\, W^T + \text{Mean(calculated in Q1.1)} = X_{new}$$

Eq.1

X is data matrix which has 4965 images consists of 2304 pixels. W is the matrix consisting of k eigenvectors as column vectors.

Reconstructed Images:

Fig. 3: Reconstructed images with each k value

Muhammed Şamil Arınç
21803683

Question 2

2.1.

The given equation can be extended as

$$J_n = y^T y - y^T XB - B^T X^T y + B^T X^T XB$$

Eq. 2

Taking the derivative w.r.t. B it becomes:

$$\frac{dJ_n}{dB} = 2X^T y + 2X^T XB$$

Eq. 3

This equation should be zero in order to get optimal results. Hence,

$$2X^T y = 2X^T XB$$

Eq. 4

Since $X^T X$ has a rank of 13 which is equal to its number of columns, it means that the matrix is invertible. Hence,

$$B = (X^T X)^{-1} X^T y$$

Eq. 5

B represents the optimal thetas of the linear regression model with respect to given data matrix X and label matrix y. The first column of the data matrix should be set to all ones and after the 1st column all columns should consist of the data of the feature number i+1 where i is the column number. That's because, we should also consider w_0 in our calculations which is a constant and not multiplied with any data value.

2.2.

The rank of the given matrix is 13 as mentioned in Q2.2. Since the rank number is equal to the columns number of the matrix, it can be said that matrix is invertible. Also, the given data has 13 features which implements that the matrix is invertible when it is multiplied with its transpose as shown inside parenthesis of Equation 5.

2.3.

The thetas matrix which consists of the optimal parameters of linear regressor is

```
[[34.55384088]
 [-0.95004935]]
```

This implies that w_0 is 34.55384088 and w_1 is -0.95004935. Since we just used LSTAT feature for this question, it can be said that the prediction of the linear regressor is $w_0 + w_1$ * LSTAT.

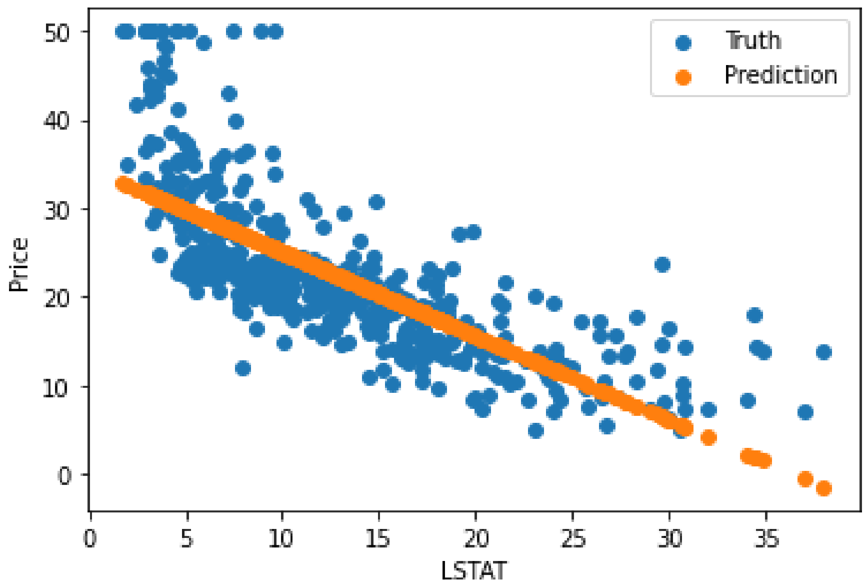

Fig. 4: LSTAT-Price graph

It can be seen in the Figure 4 that LSTAT feature has a negative effect on the price. When it increases the price decreases. Also, it can be seen that our model's predictions are linear which is what we expect in the linear regressor model. And the mean squared error of the model is MSE = `139.5435` and if we divide it into number of datas it becomes `0.2758.`

2.3.

The optimal parameters of the polynomial regressor are:

```
[[42.86200733]
 [-2.3328211 ]
 [ 0.04354689]]
```

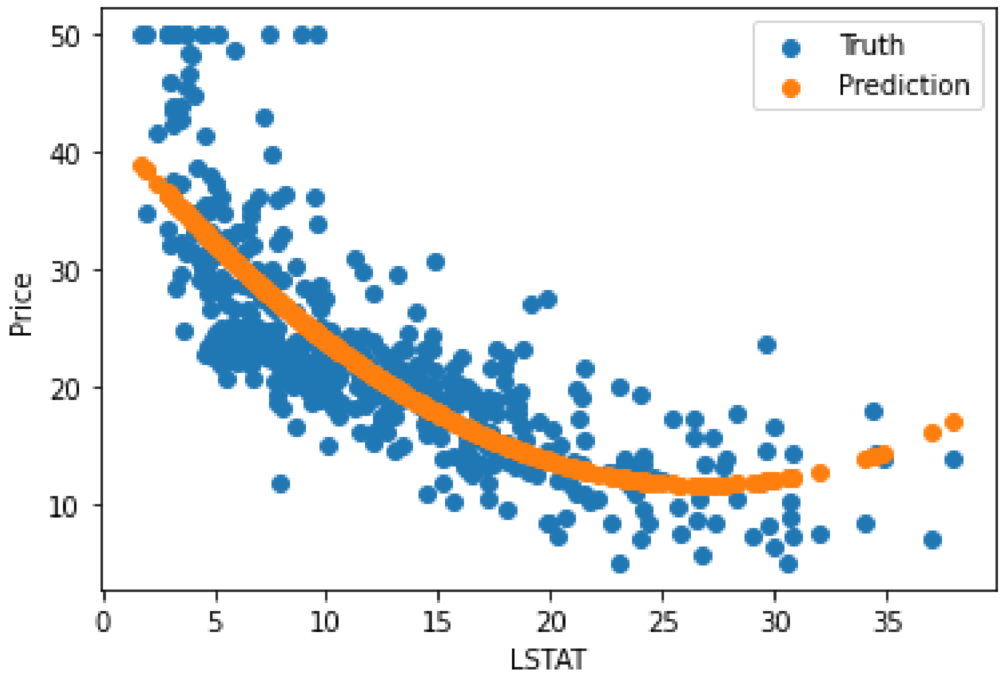

Fig. 5: LSTAT vs Price Graph

For the polynomial regressor the LSTAT feature again has a negative effect on the price however, after some value, it becomes proportional to the price with a positive manner. MSE for the polynomial regressor is `123.8840` and after dividing it into number of datas it becomes `0.2448`.

Question 3

3.1. – 3.2.

Accuracies and weights for different alphas are:

```
For 1e-5
array([[-0.16709404],
       [-0.50547407],
       [-0.10840616],
       [ 0.06074037]])

Train accuracy: 0.6166

Test:
Confusion Matrix:
 [[  0  69]
 [  0 110]]
Accuracy: 0.615
Precision: nan
Recall: 0.000
Negative Predictive Value: 0.615
False Positive Rate: 0.385
F1 and F2 scores: nan, nan
```

```
For 1e-4
array([[ 0.57219593],
       [-1.45638612],
       [-0.48654093],
       [ 0.44683419]])

Train accuracy: 0.6840

Test:
Confusion Matrix:
 [[27 42]
 [19 91]]
Accuracy: 0.659
Precision: 0.587
Recall: 0.391
Negative Predictive Value: 0.684
False Positive Rate: 0.316
F1 and F2 scores: 0.470, 0.419
```

```
For 1e-3
array([[ 1.70102622],
       [-1.97755776],
       [-2.73013922],
       [ 1.28611756]])
```

```
Train accuracy: 0.7121

Test:
Confusion Matrix:
 [[31 38]
 [16 94]]
Accuracy: 0.698
Precision: 0.660
Recall: 0.449
Negative Predictive Value: 0.712
False Positive Rate: 0.288
F1 and F2 scores: 0.534, 0.480
```

```
For 1e-2
array([[ 2.61221239],
       [-1.99087256],
       [-3.21911101],
       [ 1.71843768]])

Train accuracy: 0.6601

Test:
Confusion Matrix:
 [[50 19]
 [34 76]]
Accuracy: 0.704
Precision: 0.595
Recall: 0.725
Negative Predictive Value: 0.800
False Positive Rate: 0.200
F1 and F2 scores: 0.654, 0.694
```

```
For 1e-1
array([[  0.05423731],
       [-40.58875457],
       [-38.83425394],
       [ 28.58597096]])

Train accuracy: 0.6264

Test:
Confusion Matrix:
 [[  5  64]
 [  2 108]]
Accuracy: 0.631
Precision: 0.714
Recall: 0.072
Negative Predictive Value: 0.628
False Positive Rate: 0.372
F1 and F2 scores: 0.132, 0.088
```

```
For mini-batch with n=100
array([[ 1.78858485],
       [-1.97215461],
       [-2.94735654],
```

```
       [ 1.6090071 ]])

Train accuracy: 0.7051

Test:
Confusion Matrix:
 [[33 36]
 [16 94]]
Accuracy: 0.709
Precision: 0.673
Recall: 0.478
Negative Predictive Value: 0.723
False Positive Rate: 0.277
F1 and F2 scores: 0.559, 0.508



For schoastic n=1
array([[ 1.59864208],
       [-1.90497177],
       [-2.6494684 ],
       [ 1.75589173]])

Train accuracy: 0.7093

Test:
Confusion Matrix:
 [[32 37]
 [17 93]]
Accuracy: 0.698
Precision: 0.653
Recall: 0.464
Negative Predictive Value: 0.715
False Positive Rate: 0.285
F1 and F2 scores: 0.542, 0.492
```

The learning rate 1e-3 reaches the maximum accuracy in train and 1e-2 for test, therefore, the learning rate is taken as 1e-3 is used for mini batches.

3.3.

In some cases, such as rare diseases the accuracy might seem very high if we predict all the data as negative. Therefore, the NPV, FDR, and FPR become very useful to prevent such kind of misunderstood accuracies. However, if we have a nearly uniformly distributed data also the accuracy is also good metric. Also, the F values can be very useful for arranging a distribution. When beta goes high, we can predict the labels more confidently.