BILKENT UNIVERSITY

ENGINEERING FACULTY

DEPARTMENT OF COMPUTER ENGINEERING

CS464

Introduction to Machine Learning - Fall 2021

Homework 2

Bulut Gözübüyük 21702771

1 PCA & Eigenfaces

Question 1.1

At first, I have mean-centered the data. Thereafter, I have applied PCA and obtained the first 10 principal components. Then, reshaped each of the principal components to a 48x48 matrix which can be seen below respectively.

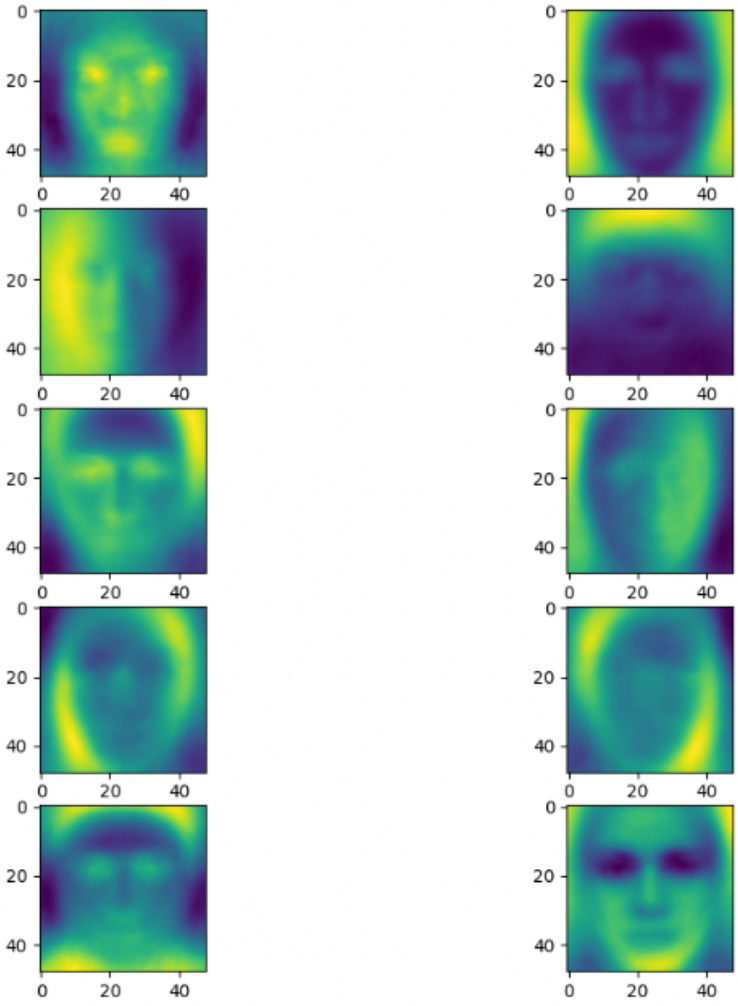


Figure 1

```
PVE for k 0 0.2833447489537039
PVE for k 1 0.11027901264243294
PVE for k 2 0.0976680318398773
PVE for k 3 0.06101507486957516
PVE for k 4 0.03217828661264683
PVE for k 5 0.02860724839829488
PVE for k 6 0.020955561849916805
PVE for k 7 0.020521356816013785
PVE for k 8 0.01841829787945829
PVE for k 9 0.0140912195672334 51
```

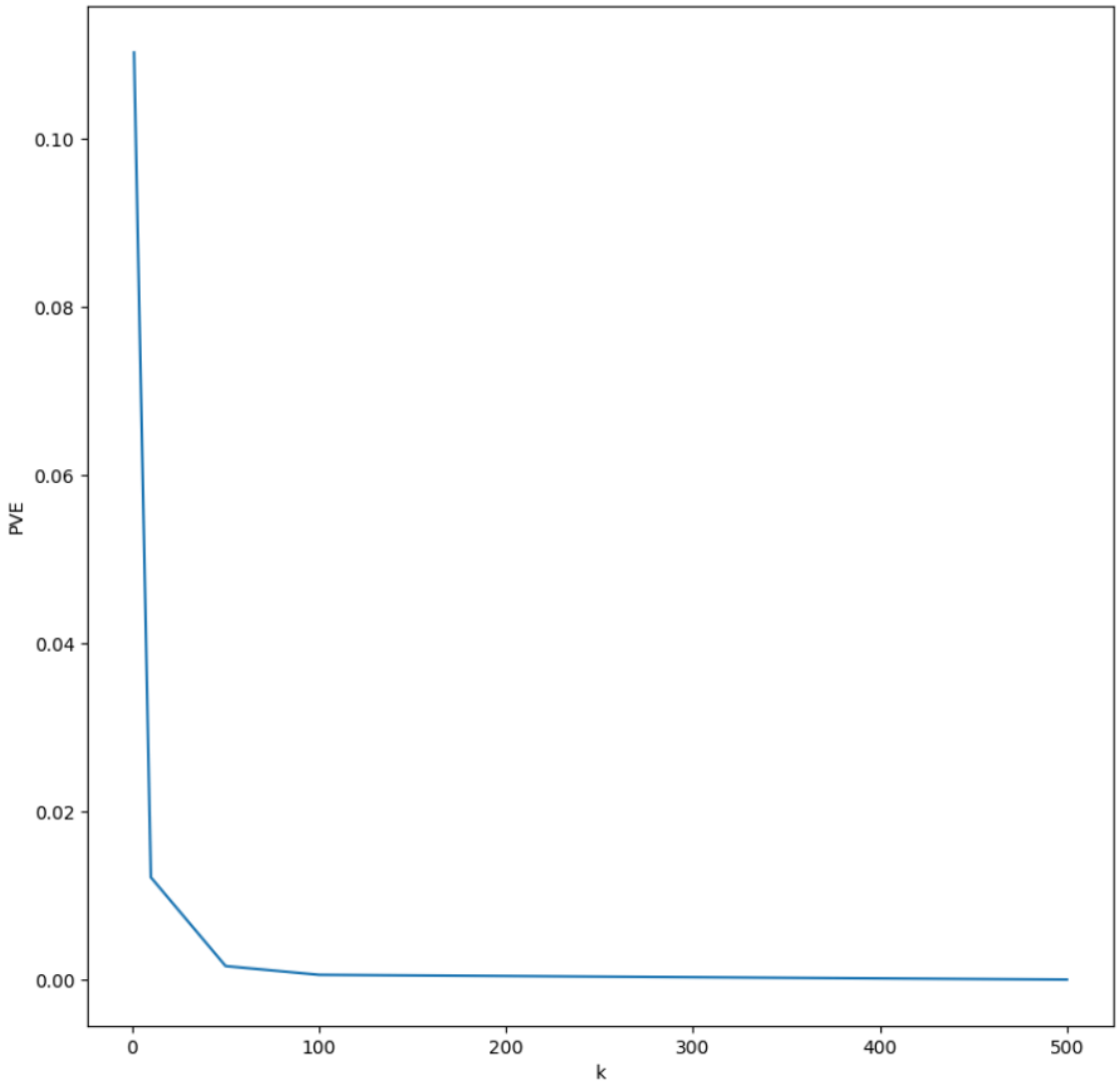
In PCA, the information stored decreases with the increase of k. Most information is stored in the 1st principal component. As can be seen in the output above the PVE decreases while k increases.

Question 1.2

I have obtained the first k principal components and PVE for k ∈ {1, 10, 50, 100, 500} can be seen below.

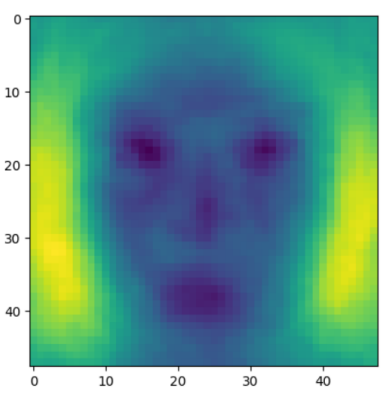
```
PVE for k 1 0.11027901264243294
PVE for k 10 0.012221631689127899
PVE for k 50 0.0016630519588116878
PVE for k 100 0.0006227885310344548
PVE for k 500 5.4658398731637264e-05
```

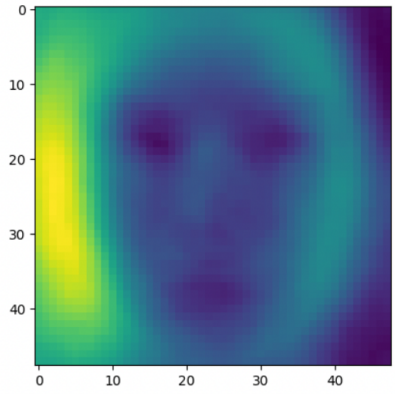
The plot of k vs. PVE

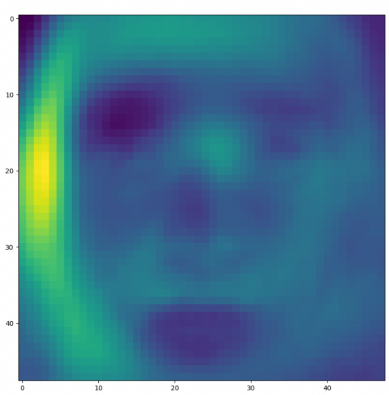


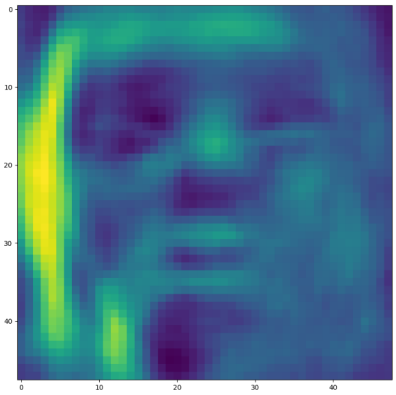
Question 1.3

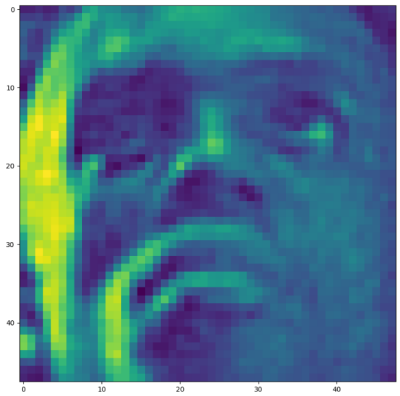
Below, reconstructed images can be seen for k ∈ 1, 10, 50, 100, 500 respectively. As can be seen, the reconstruction result is better if the number of principle count increases because more principal component means more information. Reconstruction can be done with help of this formula $= XX^TY$, X is the subset of eigenvectors and Y is the image.











2 Linear & Polynomial Regression

Question 2.1

Derivation
Weight → B
Input → X
Label → Y

$$J_n = ||y - X\beta||^2 = (y - X\beta)^T(y - X\beta)$$

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

The best B value can be found with the derivative.

$$\frac{\partial J_n}{\partial B} = -2y^T X + 2B^T X^T X = 0$$

$$y^T X = B^T X^T X$$

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

Question 2.2

The rank is 13 according to the output of the script. A rank is a number of linearly independent rows or columns of a matrix. It can be said that our matrix is invertible since the rank is 13 and our matrix's size is 13x13. Thus, a full rank matrix can be invertible.

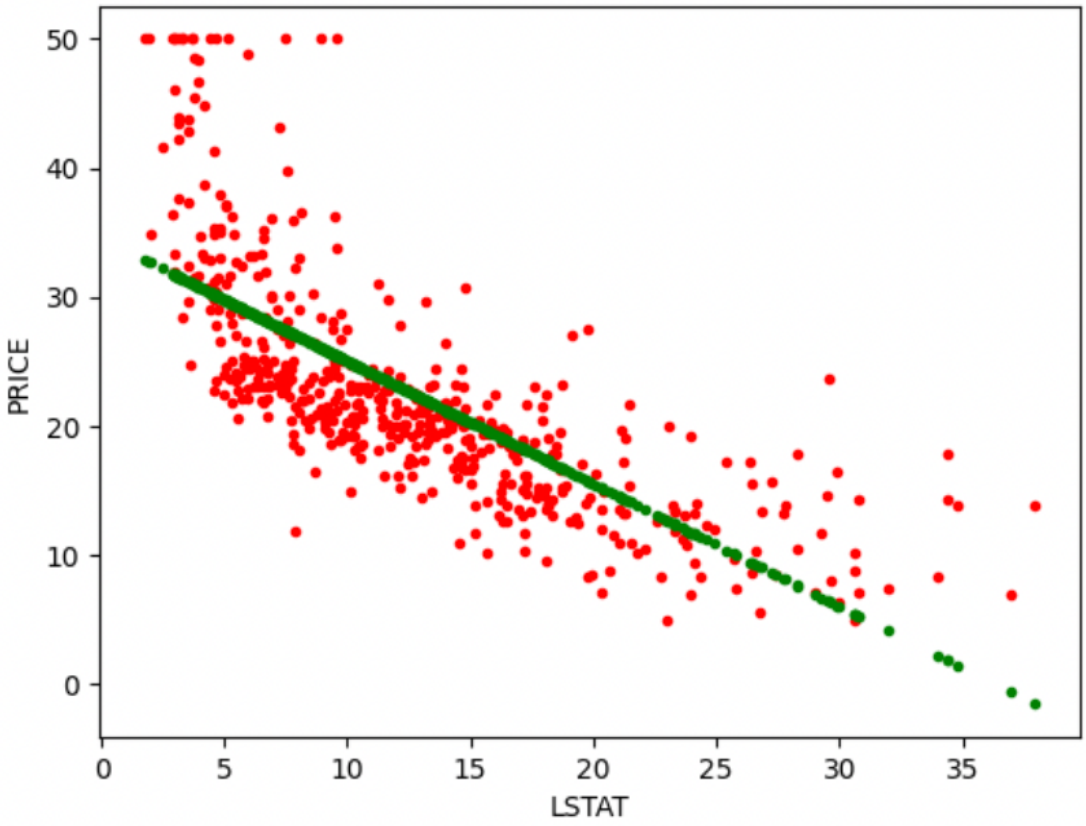
Question 2.3

Bias
34.55384088

LSTAT
-0.95004935

```
rank is 13
Bias and LSTAT [[34.55384088]
 [-0.95004935]]
MSE 38.48296722989415
```

The best line can be found using the formula that we have derived in question 2.1 $B = (X^T X)^{-1} X^T y$

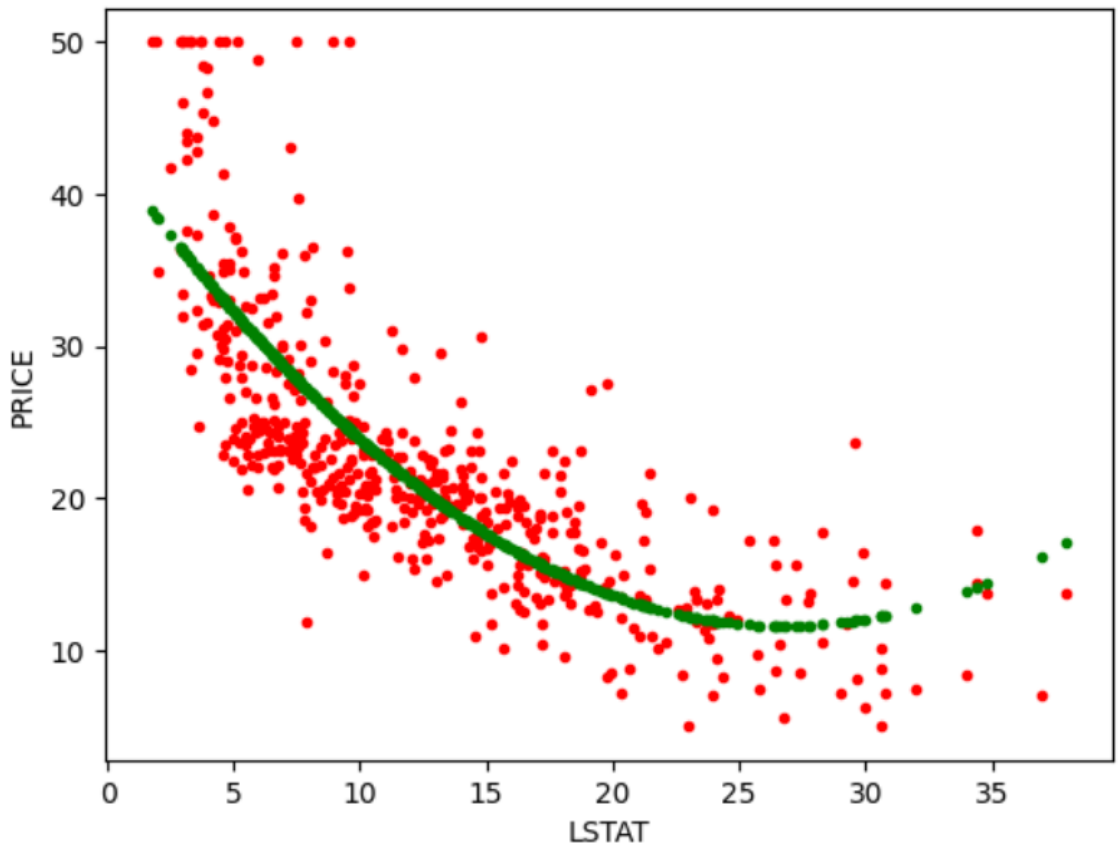


Question 2.4

```
Bias and LSTAT and LSTAT2 [[42.86200733]
 [-2.3328211 ]
 [ 0.04354689]]
MSE 30.330520075853716
```

Bias 42.86200733
LSTAT -2.3328211
LSTAT^2 0.04354689

Since we have a square we have a polynomial shape in our results. Thus, our MSE is lower now.



3 Logistic Regression

Question 3.1

```
learning rate: 1e-05 | accuracy: 0.6815642458100558
confusion matrix: [[31, 38], [19, 91]]
precision:  0.4492753623188406  recall: 0.62
npv:  0.8272727272727273 fpr: 0.29457364341085274 fdr: 0.5507246376811594
f1: 0.5210084033613446 f2: 0.3256302521008404
----------------------------------------
learning rate: 0.0001 | accuracy: 0.6927374301675978
confusion matrix: [[33, 36], [19, 91]]
precision:  0.4782608695652174  recall: 0.6346153846153846
npv:  0.8272727272727273 fpr: 0.28346456692913385 fdr: 0.5217391304347826
f1: 0.5454545454545455 f2: 0.3409090909090909
----------------------------------------
learning rate: 0.001 | accuracy: 0.6927374301675978
confusion matrix: [[33, 36], [19, 91]]
precision:  0.4782608695652174  recall: 0.6346153846153846
npv:  0.8272727272727273 fpr: 0.28346456692913385 fdr: 0.5217391304347826
f1: 0.5454545454545455 f2: 0.3409090909090909
----------------------------------------
learning rate: 0.01 | accuracy: 0.6145251396648045
confusion matrix: [[29, 40], [29, 81]]
precision:  0.42028985507246375  recall: 0.5
npv:  0.7363636363636363 fpr: 0.3305785123966942 fdr: 0.5797101449275363
f1: 0.45669291338582674 f2: 0.2854330708661417
----------------------------------------
learning rate: 0.1 | accuracy: 0.6927374301675978
confusion matrix: [[36, 33], [22, 88]]
precision:  0.5217391304347826  recall: 0.6206896551724138
npv:  0.8 fpr: 0.2727272727272727 fdr: 0.4782608695652174
f1: 0.5669291338582677 f2: 0.35433070866141736
----------------------------------------
```

The choosen learning rate is 0.001.

Question 3.2

Mini batch n = 100

```
----------------------------------------
mini batch n = 100
learning rate: 0.001 | accuracy: 0.6927374301675978
confusion matrix: [[31, 38], [17, 93]]
precision:  0.4492753623188406  recall: 0.6458333333333334
npv:  0.8454545454545455 fpr: 0.2900763358778626 fdr: 0.5507246376811594
f1: 0.5299145299145299 f2: 0.3311965811965813
----------------------------------------
```

Stochastic gradient ascent algorithm

```
----------------------------------------
stochastic gradient ascent algorithm
learning rate: 0.001 | accuracy: 0.6703910614525139
confusion matrix: [[28, 41], [18, 92]]
precision:  0.4057971014492754  recall: 0.6086956521739131
npv:  0.8363636363636363 fpr: 0.3082706766917293 fdr: 0.5942028985507246
f1: 0.48695652173913045 f2: 0.30434782608695654
----------------------------------------
```

Question 3.3

If there is an unequal class distribution, the F1 and F2 Scores would be a preferable measure to employ. It can be stated that NPV would be useful if true negatives are the majority among negatives. FPR is better if false positives are higher than true negatives. Lastly, if false positives are the majority in positives FDR would be a better choice.