
HAMMAD KHAN MUSAKHEL

21801175

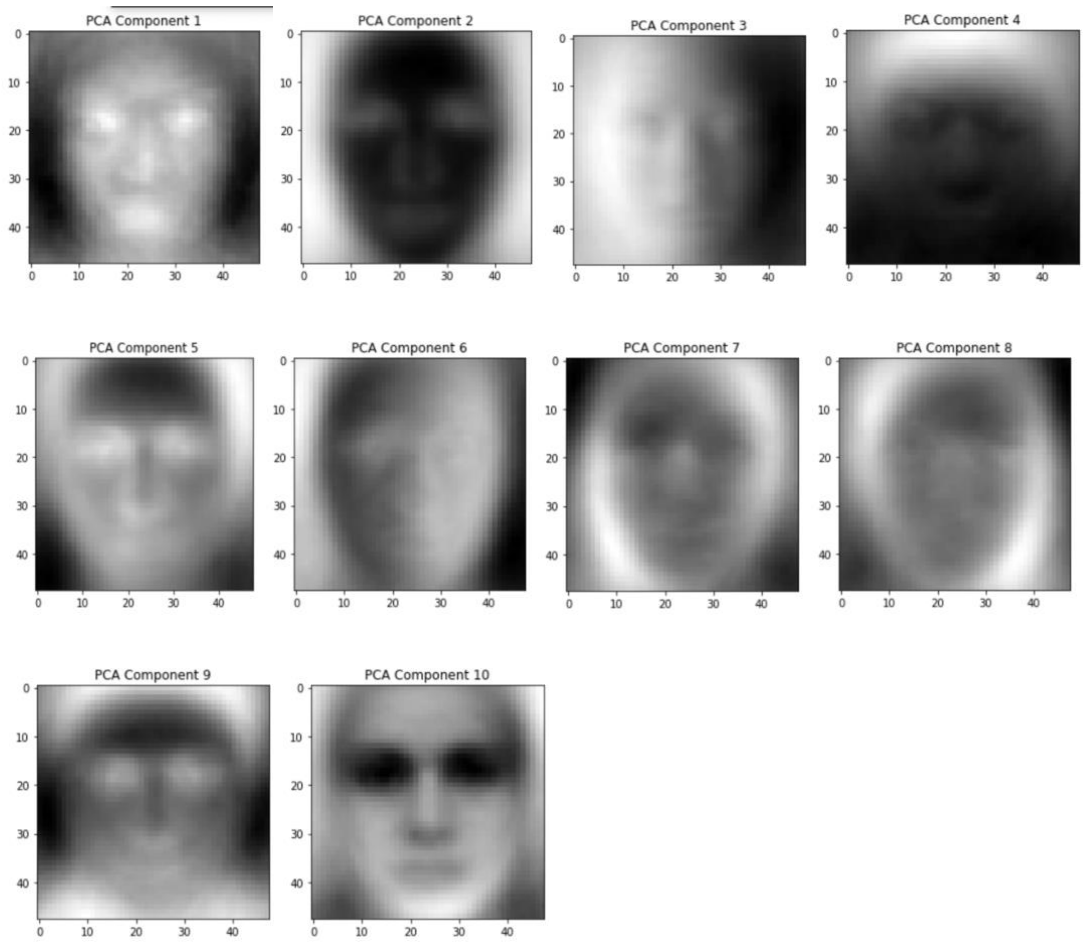
CS464-001

HW2

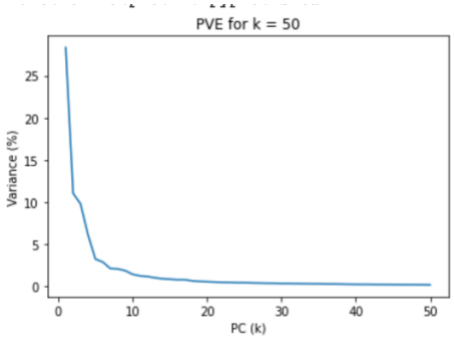
1.1) The proportion of variance explained for each PC

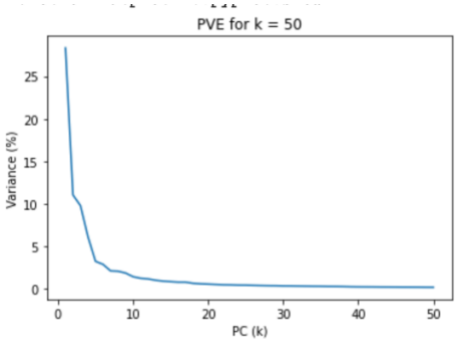
```
[28.3344749  11.02790126  9.76680318  6.10150749  3.21782866  2.86072484
  2.09555618  2.05213568  1.84182979  1.40912196]
```

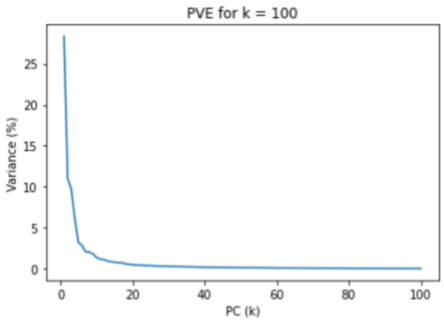
Reshape each of the principal component to a 48x48 matrix; since the PCA was done for all images in the data set, the eigen vectors correspond to a general structure of the images in the data set.

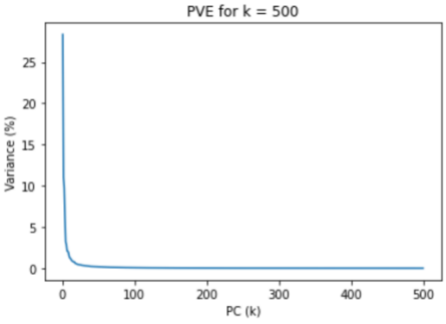



1.1 ) The figures below depict the PVE in relation with increasing K values. The increasing of Principal components shows a decrease in the Variance % as the eigen vectors are sorted in descending order. The variance decreases for higher PCs as principal components are basically the eigen vectors. It is mentioned above that eigen vectors of the covariance matrix of the dataset, and that the eigen vectors are arranged in a descending order with respect to their eigen values.



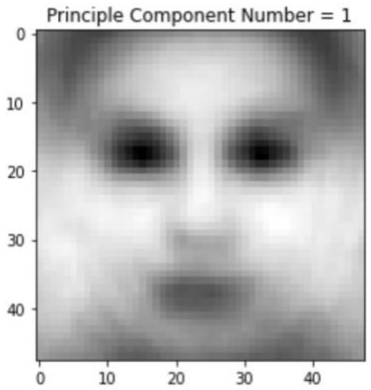



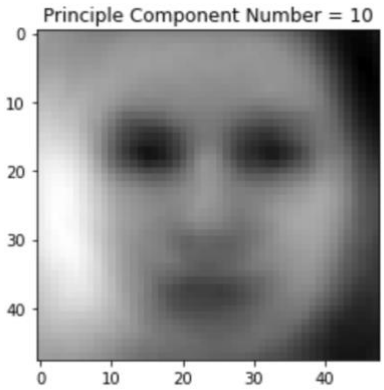



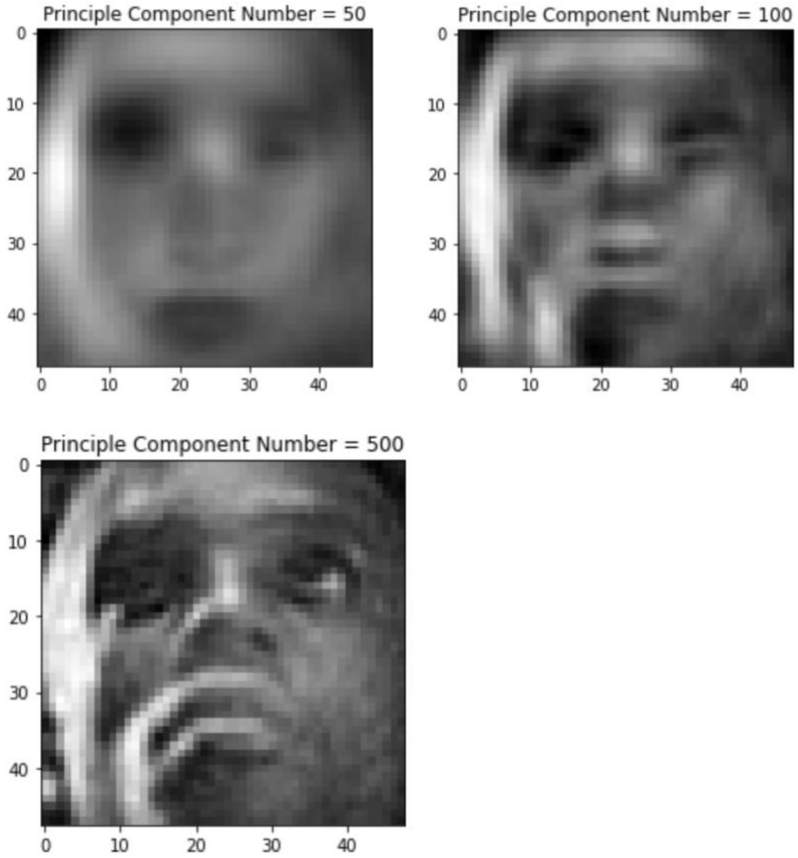




1.2 ) The reconstruction of an image using the Principal Components K, K = [1, 10, 50, 100, 500]. As we can observe that the more the eigen vectors or PCs are used, the better the image is reconstructed. This is of no surprise as more features are taken into account to reconstruct the image to its almost original shape. It starts off with a very generic image and then proceeds to a more prominent stage where the image is recognized. The variance is reduced to a much smaller for larger K and thus a more prominent reconstruction of the image is obtained.

This is the power of PCA; we have more than thousands of features but we have found out that almost 400-500 principal components are required to obtain a prominent image.



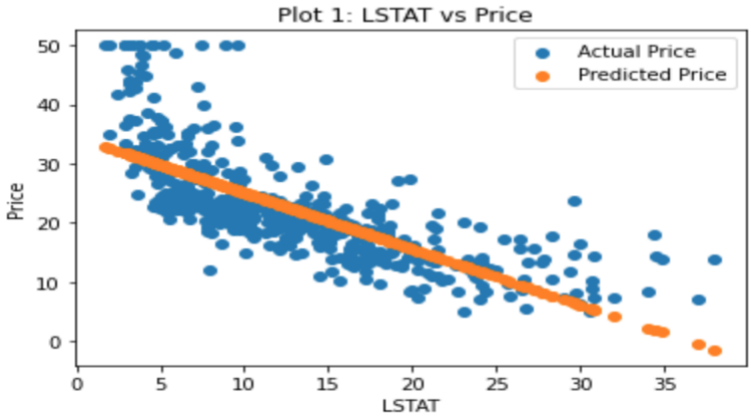



2.1) Jn = ||y-Xβ||^2 = (y-Xβ)^T (y-Xβ)
=> y^T y – 2β^T X^T y + β^T X^T Xβ
∂/∂β (Jn) = 0 - ∂/∂β(2β^T X^T y) - ∂/∂β(β^T X^T Xβ) = 0
-2X^T (-y + Xβ) = 0
β_cap = (X^T X)^-1 X^T y

2.2) The rank was 13 for the dataset; if the rank of $X$ is $m$, this means that $X$ is one-to-one when acting on $\mathbb{R}$^m. Thus, (X^T X) is one-to-one and invertible (assuming X is a n x m matrix): m is the rank of X and m <= n.

2.3)

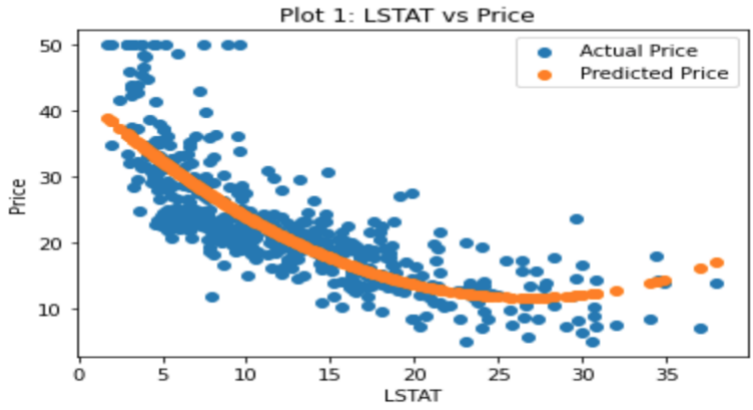
```
b0:   [[34.55384088]]
b1:   [[-0.95004935]]
The MSE is:
38.48296722989416
<function matplotlib.pyplot.show>
```




The trained beta coefficients are depicted in the figure above. As we can see that the slope is almost -1 and the intercept is at 34.55; this predicts that as we increase LSTAT, the prices of the houses will be expected to be dropped. The range of price for LSTAT being the least is 34.55. The MSE for this trained dataset is almost 39%, hinting that the predicted price with LSTAT will have an error of 39%, which is moderate. If we calculate the variance(mean) and variance(fit), we deduce that the feature LSTAT does allow us to infer/predict the price of the house with a moderate MSE.

2.4)

```
b0:   [[42.86200733]]
b1:   [[-2.3328211]]
b2:   [[0.04354689]]
The MSE is:
30.330520075853702
<function matplotlib.pyplot.show>
```



The trained beta coefficients are depicted in the figure above (used polynomial regression). As expected, the MSE is lower for this type of regression than the linear regression. The main reason is that we have more beta coefficients, and that we have a polynomial to predict the prices of the house. The addition of polynomial allows the model to have a better prediction as the curve fitting the scattered data encapsulates greater information with the flexibility of the curve. More data is taken into account in order to plot the curve, thus higher y-intercept and lower MSE for this model. The MSE is lower for this model as the curve is clearly seen to be better fitted than the straight-negative gradient line. The trend has some fluctuations as the data has random values. For such continuous set of labels, it is better to have this type of model.

3.1)

Learning rate used is: 10^-3

```
Logistic regression Accuracy:  65.92178770949721
TP:  28   TN:  90   FP:  20   FN:  41

#all values in terms of %
Precision:  58.333333333333336
Recall:  40.57971014492754
FPR:  41.66666666666667
FDR:  15.267175572519085
NPV:  68.70229007633588
F1:  47.863247863247864
F2:  43.20987654320988
```

3.2)

The time taken for this model was significantly higher than the model used in 3.1. However, it is well known that too large of a batch size will lead to poor generalization (although this is not the case in my model); It has been empirically observed that smaller batch sizes not only have faster training dynamics but also generalization to the test dataset versus larger batch sizes. The same values for accuracies and averages are given for stochastic gradient ascent, however the time taken for that is the most significant compared to mini-batch and full-batch gradient ascent.

```
Logistic regression Accuracy:  65.92178770949721
TP:  28   TN:  90   FP:  20   FN:  41
#all values in terms of %
Precision:  58.333333333333336
Recall:  40.57971014492754
FPR:  41.66666666666667
FDR:  15.267175572519085
NPV:  68.70229007633588
F1:  47.863247863247864
F2:  43.20987654320988
```

3.3) F2 and F1 scores can be used to evaluate models if the recall and precision are biased and rigid. If the labels of the dataset are uneven and one outweighs the other, then it affects the training process. F1 and F2 allows us to weigh precision and recall and if there is a balance between the both.

Likewise, FPR can be used to determine how many of the truly negative values are predicted incorrectly. FPR, when used with recall score, depicts the characteristic of the classifier being too inclined to 0 or 1, true or false, etc. In this case, accuracy might be relatively higher but the results aren't pleasing in terms of classification.

NPV is used when evaluation is preferred in terms of the correctly predicted negatives. This allows us to have a better insight into the negative/o classification of the model. It allows us to see if the model is predicting positives as negatives. The higher the value of NPV, the better negatives are evaluated.

FDR is a metric which observes the data that has been classified as positive whereas which are actually negative. Thus, if precision is not high and FDR is high, FDR can be preferred to be used for evaluation. In essence, F metrics are a positive addition to data evaluation/performance metrics.