CS464 Machine Learning

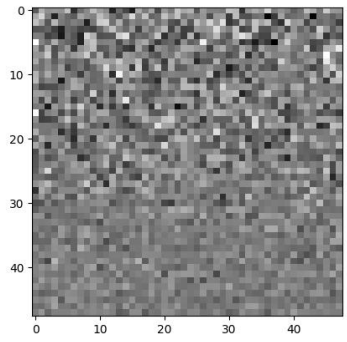
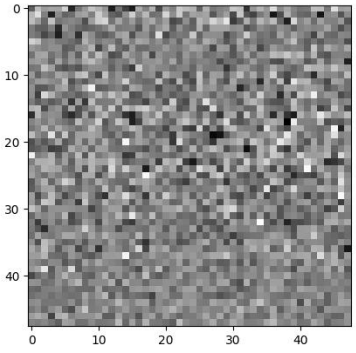
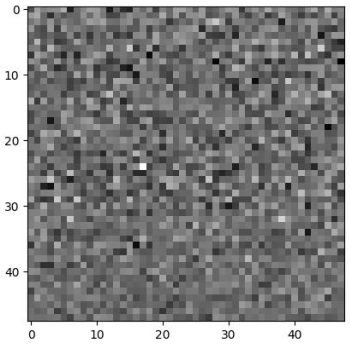
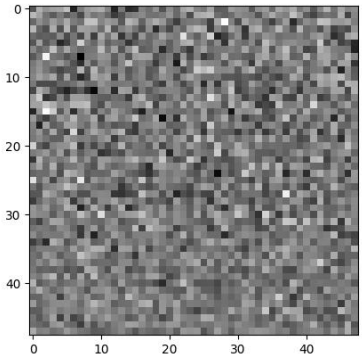
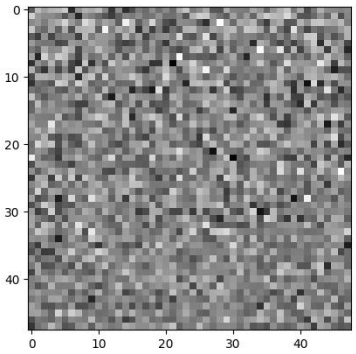
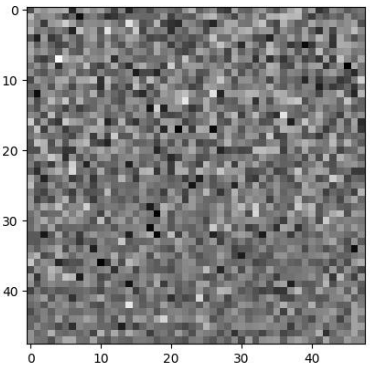
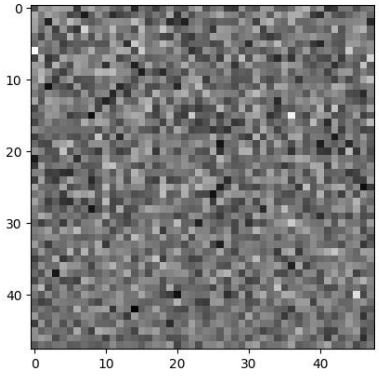
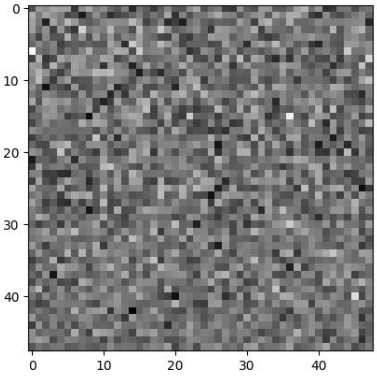
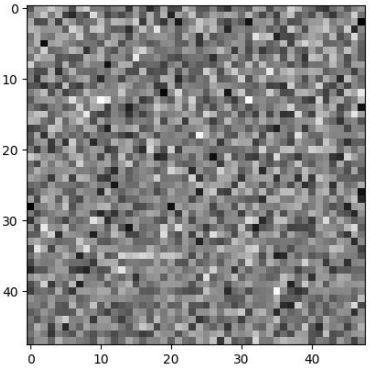
Homework 2

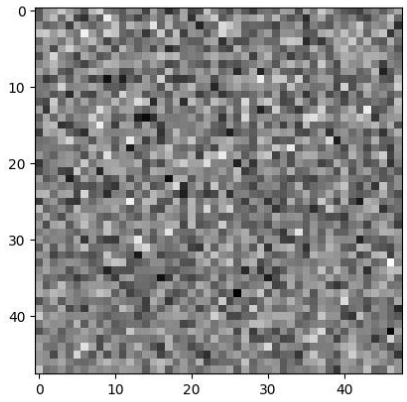
Section 2

Cansu Moran

21803665

1.1

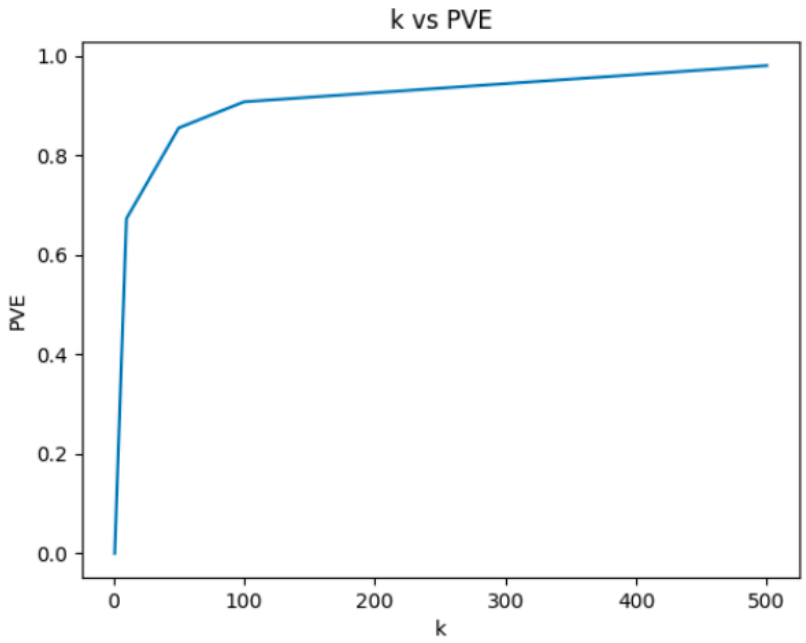


Visualization of 10 Principal Components(PC)
Order of images:
PC1 PC2 PC3
PC4 PC5 PC6
PC7 PC8 PC9
PC10

PVE by PC 1: 0.2833447489537043
PVE by PC 2: 0.11027901264243307
PVE by PC 3: 0.09766803183987764
PVE by PC 4: 0.06101507486957536
PVE by PC 5: 0.03217828661264682
PVE by PC 6: 0.02860724839829496
PVE by PC 7: 0.020955561849916787
PVE by PC 8: 0.020521356816013792
PVE by PC 9: 0.01841829787945827
PVE by PC 10: 0.014091219567233455

The principal components are in descending order in terms of the eigenvalues they are associated with. For example, PC1 had the highest eigenvalue. While making Principal Component Analysis, we try to represent the data that we have with less number of features. In order to achieve this, we try to find base vectors that can reflect the highest variances of the data. The PC with highest eigenvalue, represents the variance the best. This can also be seen from the proportion of variance explained(PVE) of the corresponding principal component. In our case, PC1 had the highest eigenvalue, and hence explained the variance in data the best. Therefore, PVE by PC1 is higher than the other. Since the PCs are in descending order, in terms of their associated eigenvalues, the PVEs explained by them are also in descending order. As the associated eigenvalue of the PC decreases, so does the ability of PC to represent the variance of the data.

1.2

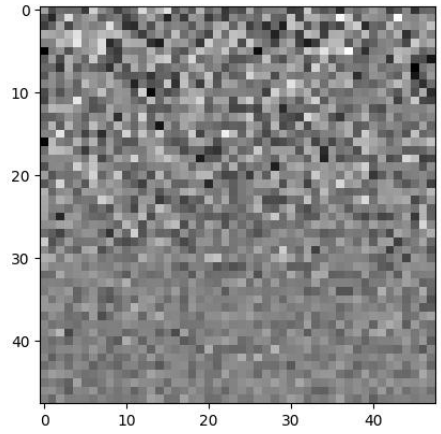


The graph shows the proportion of variance explained by k number of PCs. It can be seen at as k increases, the PVE also increases and merges to 1. This is because with more PCs, we are able to represent more variance of the data. If k is equal to the number of PCs (number of features), we expect PVE to be equal to 1, representing the entire data. It can also be seen from the graph that as k increases the difference in PVE decreases. This is because PCs with higher eigenvalue represent the variance in data better than the ones with lower eigenvalue. Therefore, as k increases, the variance explained by the added PCs loses its significance. In other words, PCs with higher eigenvalues have a higher contribution to the PVE than PCs with lower eigenvalues.

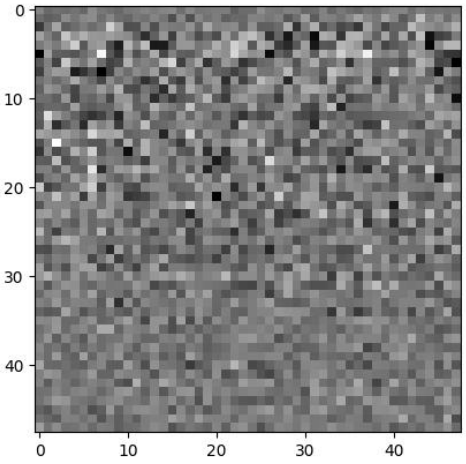
1.3

With PCA, instead of using the number of features given in the data (which is equal to the number of pixels in the image), we can use k number of PCs to represent the same data with lower dimensions. In order to achieve this, we can first take the dot product of the chosen k eigenvectors with the original image, to obtain the projection of original data onto the k PCs. Then, we can transform the reduced dimensionality projection back into original space by taking the dot product of the reduced dimensionality projection of the image with the transpose of the used PCs.

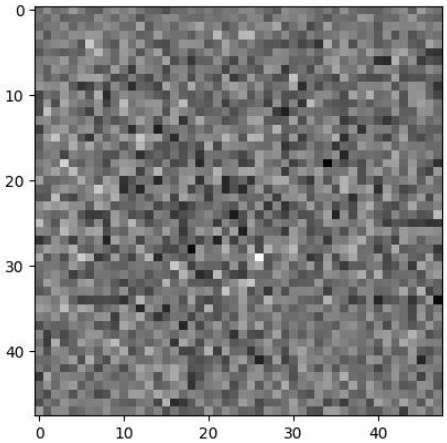
K = 1


0
10
20
30
40
0
10
20
30
40

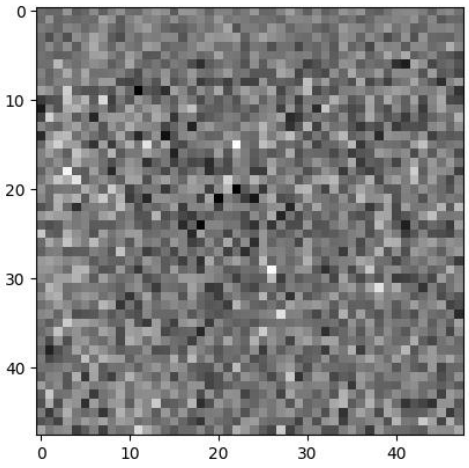
K = 10


0
10
20
30
40
0
10
20
30
40

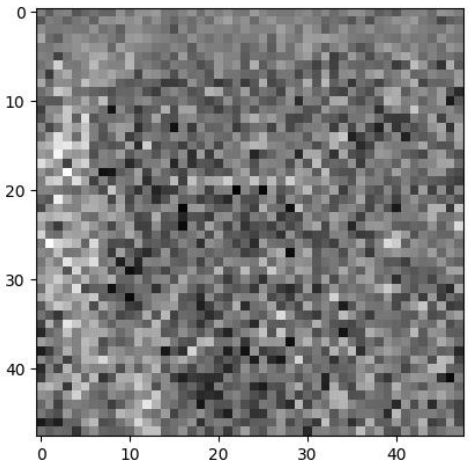
K = 50


0
10
20
30
40
0
10
20
30
40

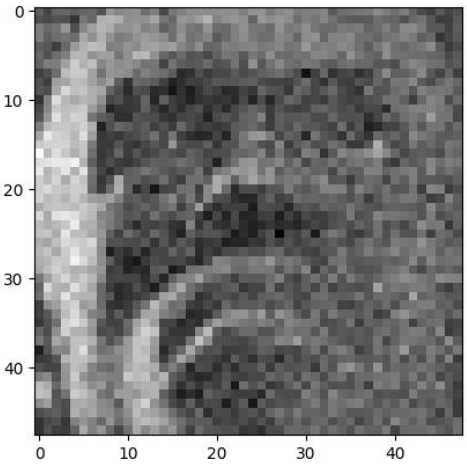
K = 100



K = 500



K = 2000



With higher k, we are able to represent more variance in the data. Therefore, while we can represent the main features of the face with a smaller number of k, as k increases, we can represent more details of the face. If k is equal to the number of features, then we expect our representation to be the same as the original image.

In the given images, because the reconstructed image was mostly noise when the k was a smaller value, the reconstructed image when k = 2000 was also included, to demonstrate how the reconstruction becomes clearer when k increases.

2.1

X represents our data matrix where each column is a feature and each row is a data sample.
Y represents the ground truth labels of the data samples. Each row in Y has the ground truth label of the corresponding row in X.
β represents the weights as a matrix. Each row represents the weight given to the feature in X. If a row in β has the index a, then the weight in that row is the weight of the feature in column a of X.

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

$\partial J(\beta)/\partial\beta = 0$ -> to minimize the loss function

$\partial J(\beta)/\partial\beta = -2X^Ty + 2X^TX\beta = 0$

$2X^Ty = 2X^TX\beta$

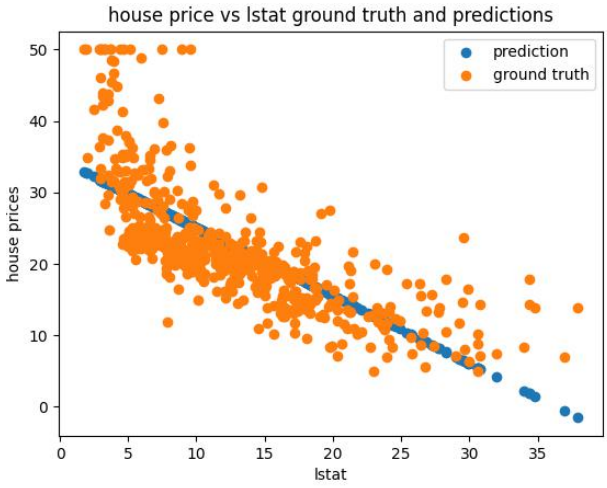
$\beta = (X^TX)^{-1}X^Ty$

2.2

X^TX has the rank (number of features + 1). Here we assume that we have appended a column of ones to X to account for the bias term. On our example, X^TX has the rank 13 + 1 = 14.

Assume X is an nxm matrix. Then the matrix X^TX will be invertible if and only if m <= n and Rank(X) = m. In our case, n represents the number of samples and m represents the number of features + 1. If we have less features than samples and the rank of X is equal to the number of features + 1, then the matrix X^TX will be invertible.

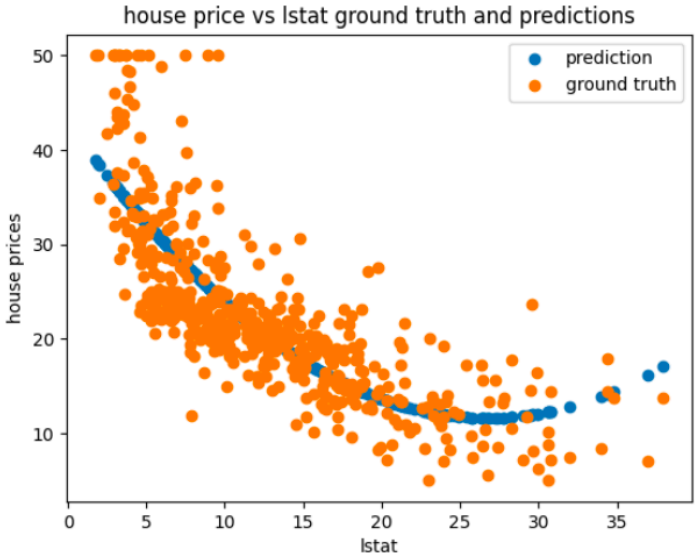
2.3




```
[[34.55384088]
 [-0.95004935]]
Mean Squared Error:  38.48296722989416
```

The coefficients found are 34.55384088 for the bias term and -0.95004935 for the feature “LSTAT”. Coefficients determine how each feature (and bias) are related to the predicted output. Here since we have a minus LSTAT coefficient we can say that the predicted output will be negatively proportional to the LSTAT values. If we look at the ground truth house prices, this observation holds. Here we can see that the house prices decreased with the increasing LSTAT values. However, we can also see from the model that linear regression doesn’t really represent the data as the data is not scattered linearly but rather polynomially. Therefore we can assume that using a polynomial regression would give better results and hence lower MSE for the given dataset.

2.4



```
[[42.86200733]
 [-2.3328211 ]
 [ 0.04354689]]
Mean Squared Error:  30.330520075853723
```

The coefficients found are 42.86200733 for the bias term, -2.3328211 for the feature “LSTAT” and 0.04354689 for the square of the LSTAT value. Similar to linear regression, since we still have a minus LSTAT coefficient which shows that the predicted output will be negatively proportional to the LSTAT values. However the square of LSTAT has a positive coefficient showing that it is directly proportional to house values. From the graph we can see that the polynomial regression represents the given data better. This is also seen from the MSE, as the MSE of the polynomial regression is lower than MSE of linear regression.

3.1

```
Learning rate:  0.1
Accuracy:  0.6312849162011173
True pos:  5
False pos:  2
True neg:  108
False neg:  64
Precision:  0.7142857142857143
FDR:  0.2857142857142857
Recall:  0.07246376811594203
NPV:  0.627906976744186
FPR:  0.01818181818181818
F1:  0.13157894736842105
F2:  0.25773195876288657
```

```
Learning rate:  0.01
Accuracy:  0.7039106145251397
True pos:  50
False pos:  34
True neg:  76
False neg:  19
Precision:  0.5952380952380952
FDR:  0.40476190476190477
Recall:  0.7246376811594203
NPV:  0.8
FPR:  0.3090909090909091
F1:  0.6535947712418301
F2:  0.6172839506172839
```

```
Learning rate:  0.001
Accuracy:  0.6983240223463687
True pos:  31
False pos:  16
True neg:  94
False neg:  38
Precision:  0.6595744680851063
FDR:  0.3404255319148936
Recall:  0.4492753623188406
NPV:  0.7121212121212122
FPR:  0.14545454545454545
F1:  0.5344827586206896
F2:  0.603112840466926
```

```
Learning rate:  0.0001
Accuracy:  0.659217877094972
True pos:  27
False pos:  19
True neg:  91
False neg:  42
Precision:  0.5869565217391305
FDR:  0.41304347826086957
Recall:  0.391304347826087
NPV:  0.6842105263157895
FPR:  0.17272727272727273
F1:  0.46956521739130436
F2:  0.5335968379446641
```

```
Learning rate:  1e-05
Accuracy:  0.6145251396648045
True pos:  0
False pos:  0
True neg:  110
False neg:  69
division by zero: precision
division by zero: fdr
Recall:  0.0
NPV:  0.6145251396648045
FPR:  0.0
division by zero: f1
division by zero: f2
```

When learning rate was 1e-05, True positive and true negative count was 0. Since both precision and FDR have true positive + true negative result as a denominator, we can't calculate these

values(divide by zero error). Similarly, F1 and F2 measures include recall + precision as a denominator and in this case they are zero as well. To avoid this issue, if there is a divide by zero error in calculation, an error message is printed with the associated value that can't be calculated.

Best learning rate (according accuracy), was obtained when learning rate was 1e-02. Therefore this learning rate was used in the following mini-batch and stochastic gradient descent algorithms.

3.2

Logistic regression with mini-batch gradient descent results (learning rate: 1e-02):

```
Accuracy:  0.6983240223463687
True pos:  31
False pos:  16
True neg:  94
False neg:  38
Precision:  0.6595744680851063
FDR:  0.3404255319148936
Recall:  0.4492753623188406
NPV:  0.7121212121212122
FPR:  0.14545454545454545
F1:  0.5344827586206896
F2:  0.603112840466926
```

Logistic regression with stochastic gradient descent results (learning rate: 1e-02):

```
Accuracy:  0.6927374301675978
True pos:  48
False pos:  34
True neg:  76
False neg:  21
Precision:  0.5853658536585366
FDR:  0.4146341463414634
Recall:  0.6956521739130435
NPV:  0.7835051546391752
FPR:  0.3090909090909091
F1:  0.6357615894039734
F2:  0.6045340050377833
```

3.3

Accuracy might not be trustworthy when there is an imbalanced dataset. For example, if we have 90% of our samples with label 1 and the remaining with label 0, if we predicted all of the samples to 1, we would get 90% accuracy. However, if we have another dataset, where there 90% of the samples have label 0 and the remaining label 1, the same model would only give 10% accuracy. In such cases accuracy can be misleading. Moreover, even if we use precision and recall, our metrics would still be biased towards positive labels, due to the main definition of these metrics. Therefore, they might not accurately reflect the success of the model in terms of negative labels. In such cases, we can utilize other performance metrics to obtain a better understanding of the model performance. The performance metric we choose should be aligned with the overall structure of the problem and the data.