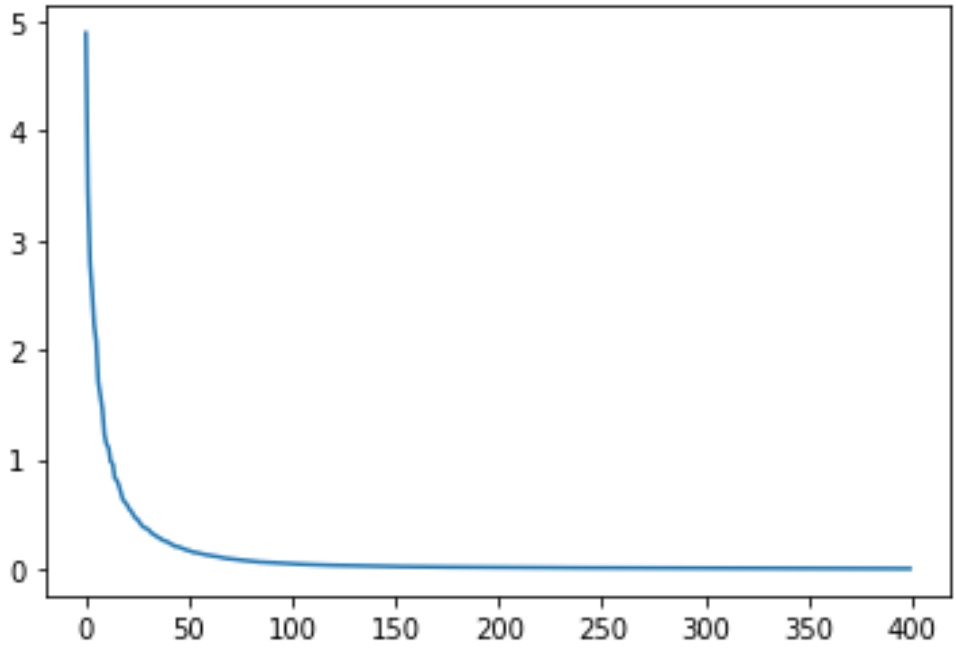
GE 461: Introduction to Data Science
Homework #1

Batıhan Akça - 21502824

Question 1:

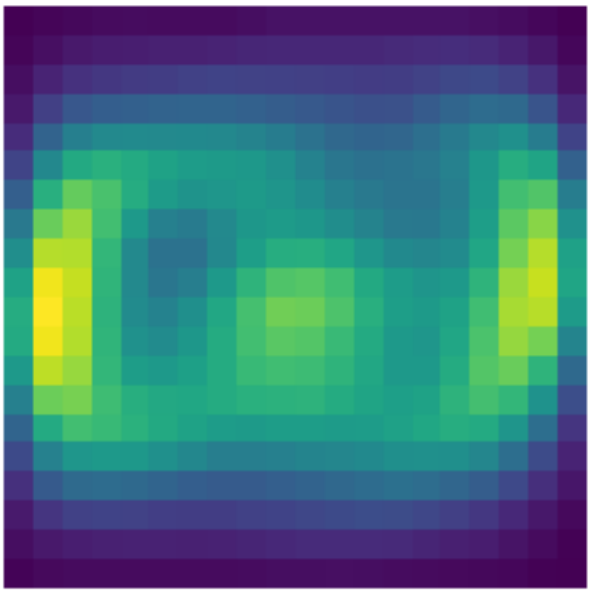
1.1



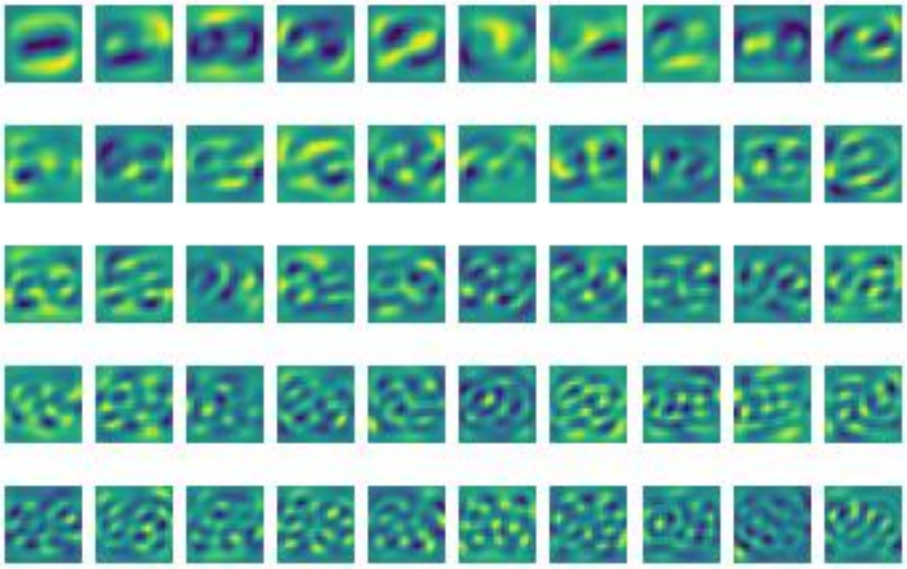
400 Eigenvalues plotted in descending order

- Plot shows that the eigenvalue almost converges to zero after 50th component. Since from eigenvalue we infer the explained variance of the variables, after 50th component, variance explanation effect is reasonably decreases. Therefore, by just looking at this plot, I would choose 60 as the number of components.

1.2



Mean picture of the dataset



50 bases (eigenvectors)

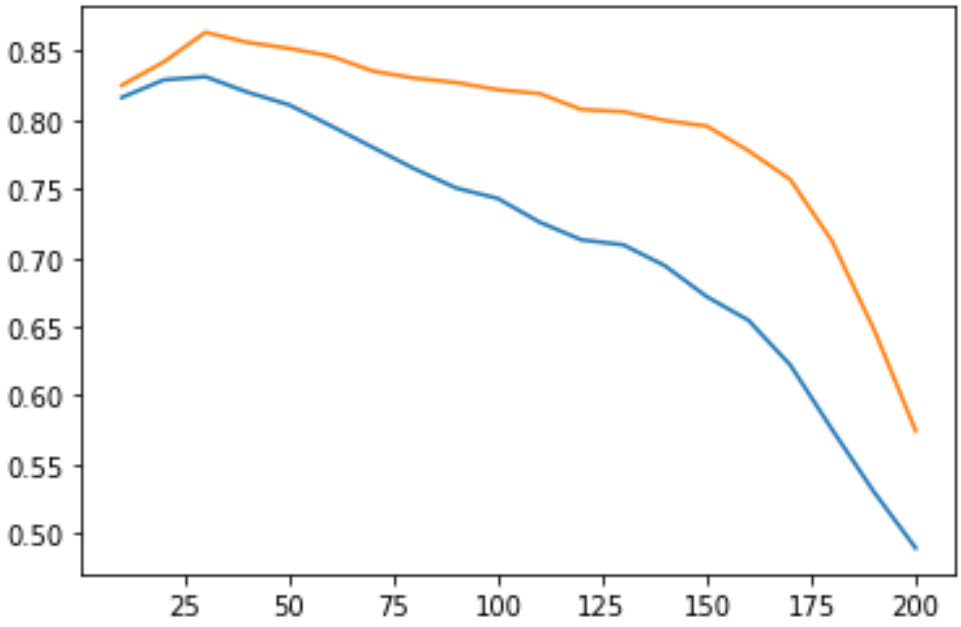
- From the first eigenvalue, it is obvious that the highest variance is on the circular shapes of the data. It is reasonable because in the dataset we have circular shaped digits as 3,6,8,9 and non-circular shaped digits as 1,7 and the mixed ones as 2,4,5. Even without using any advanced technique, a classification can be made for the digits. 50 eigenvalue and the mean digit picture support the idea that we have the high variance on circular shapes.

1.3

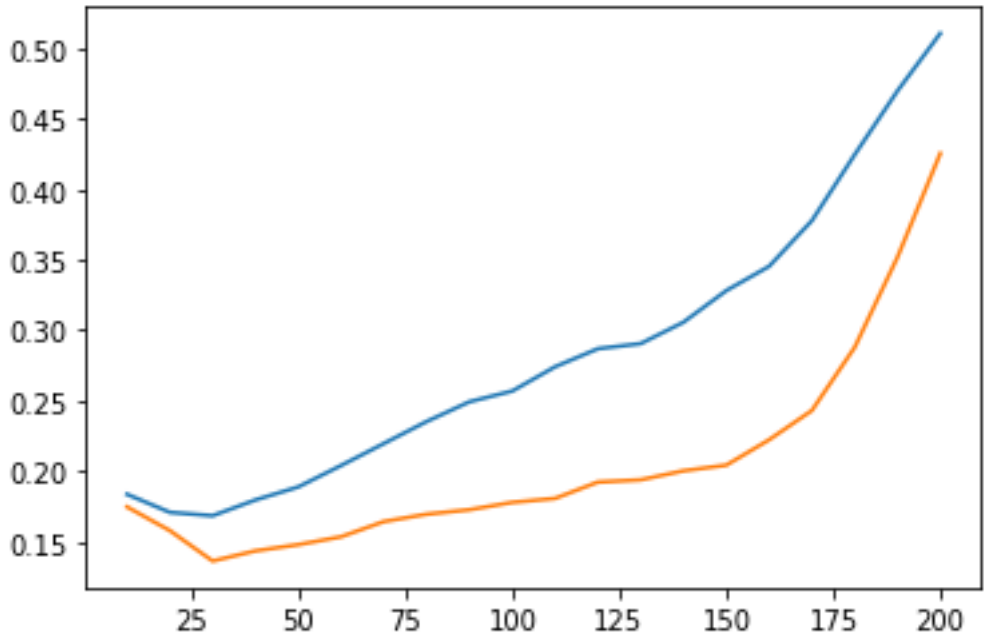
TEST		TRAIN	
10	0.816327	10	0.82527
20	0.829132	20	0.842063
30	0.831533	30	0.863655
40	0.820328	40	0.856457
50	0.811124	50	0.852059
60	0.795918	60	0.846461

70	0.780312	70	0.835666
80	0.764706	80	0.830468
90	0.7507	90	0.827269
100	0.743097	100	0.822071
110	0.72589	110	0.819272
120	0.713085	120	0.807677
130	0.709484	130	0.806078
140	0.694278	140	0.79968
150	0.671869	150	0.795682
160	0.654662	160	0.777689
170	0.622249	170	0.756897
180	0.57543	180	0.712115
190	0.530212	190	0.648141
200	0.489396	200	0.57457

Gaussian Classifier Prediction Accuracies with Reduced Dimensions



PCA (component number vs classification accuracy)
Orange=Train, Blue=Test

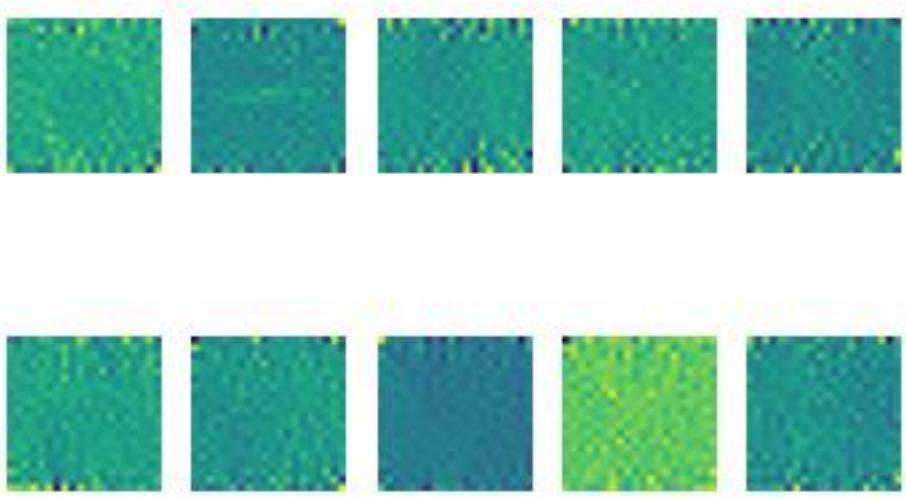


PCA (dimension number vs classification error)
Orange=Train, Blue=Test

- As expected, train data's prediction results are always overperforming against test data's results. In the first part I predicted the optimum number of components as around 50, but results show that accuracy peak when 30 components are used in the analysis. Dimension numbers are increasing from 10 to 200 by 10 therefore we cannot say the optimum number is the 30 but we are sure that it is between 20 and 40 from the plots. Increasing error term is not surprising because by adding more components we may give extra weight to fewer necessary features in the data in terms of prediction, therefore the results are consistent with the theory of the PCA.

Question 2:

2.1



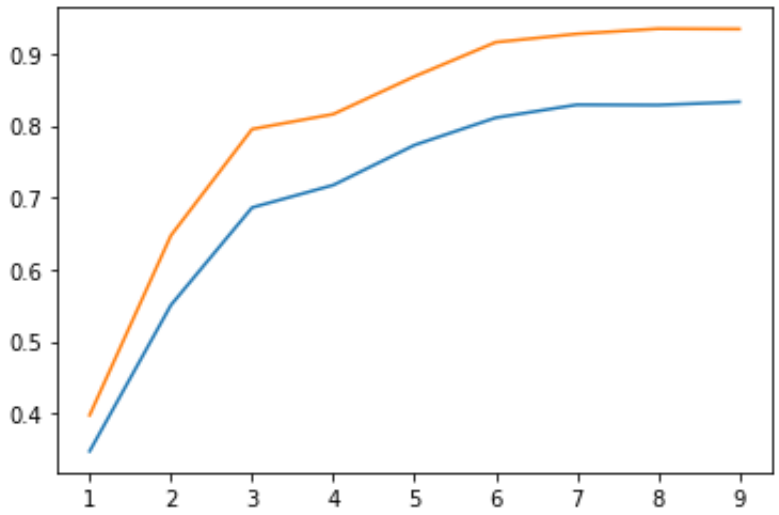
10 Bases of LDA

- We see some similar and very distinct bases. It is expected because by LDA we are giving classes a huge weight unlike the PCA method. Those similar are and distinct images may show how similar some of the digits and some are not. For example, 4th picture of the 2nd row, may show how the digit 8 includes two circular shape unlike the others.

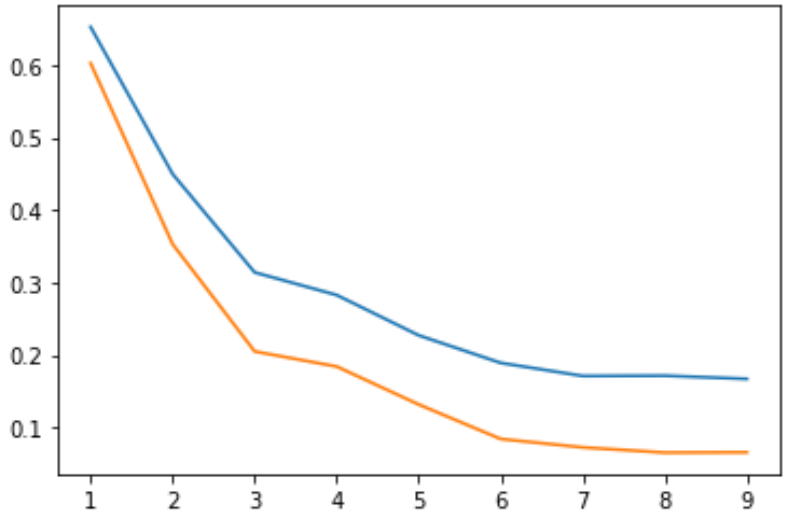
2.2

	TEST		TRAIN
1	0.347739	1	0.397841
2	0.55062	2	0.647341
3	0.686275	3	0.795282
4	0.717487	4	0.816074
5	0.773109	5	0.868453
6	0.811124	6	0.916034
7	0.829132	7	0.927629
8	0.828731	8	0.934826
9	0.833133	9	0.934426

Prediction Accuracies on Test and Train Data with Reduced Dimensions



LDA (dimension number vs classification accuracy)
Orange=Train, Blue=Test



LDA (dimension number vs classification error)
Orange=Train, Blue=Test

- The prediction accuracy doubles itself when we increased the dimension number from 1 to 2 and reaches 77% at 6 then loses the momentum. That may show us we have at least 2 basic attributes on the digits that identify its label and plus 4 attributes to reach a satisfactory classification. In a high dimensional space, we can cluster very accurately our data points without needing the maximum number of dimensions we can obtain by LDA.

Used Libraries in the Code:

NumPy

Travis E. Oliphant. **A guide to NumPy**, USA: Trelgol Publishing

Stéfan van der Walt, S. Chris Colbert and Gaël Varoquaux. **The NumPy Array: A Structure for Efficient Numerical Computation**, Computing in Science & Engineering, **13**, 22-30 (2011)

Matplotlib

John D. Hunter. **Matplotlib: A 2D Graphics Environment**, Computing in Science & Engineering, **9**, 90-95 (2007)

Scikit-learn

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesnay. **Scikit-learn: Machine Learning in Python**, Journal of Machine Learning Research, **12**, 2825-2830 (2011)