

Aliaa M. Ali Abbas - 3747

Abdulrahman M. Nour - 4311

# Face recognition

March 07, 2019

## Objective

Form a face recognition system and get hands-on experience writing machine learning algorithms as well as dimensionality reduction techniques, mainly PCA & LDA.

## Goals

1. **Learning dimensionality reduction for feature engineering:** In implementing the algorithms we learn a method to be able to reduce high dimensional data which would require complex algorithms into lower dimensional data without significant loss of information
2. **Getting familiar with the python scientific computation modules:** We learn to implement our algorithms in an optimised fashion and to exploit the power of vectorization
3. Using these implementations to train a face recognition program.
4. **More on image procession:** during the bonus part we learn to mine for images and process them to get them into a certain set of features

## The models

For PCA:

After saving the eigenvalues and eigenvectors we then allow anyone to slice from it as many features according to a certain threshold for covariance (alpha)

For LDA: # Algorithm + Implementation

We also allow the user to choose the number of dimensions it wants to project on.

>>>> The discriminating model

Since both algorithms serve no purpose but to lower the number of dimensions.. The new reduced projected data must be then passed to a suitable discriminator. Now that it is reduced we can use a naive?simple classifier

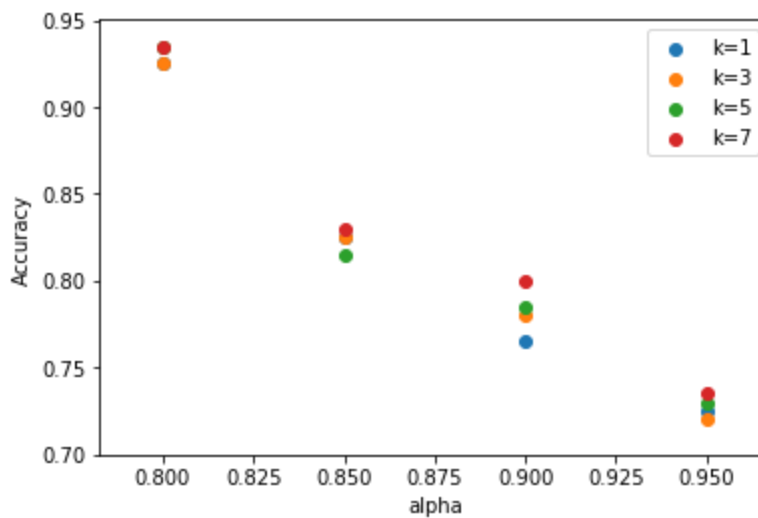
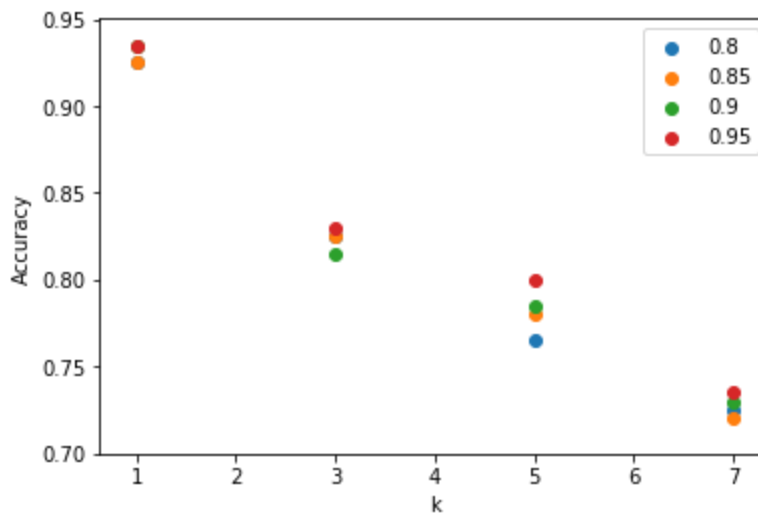
>> For the original problem: We use KNN classifier, as it would now be significantly less with the fewer features. We test the algorithm with different number of k: [1, 3, 5, 7]

>> For the bonus problem: As it is a binary classification problem we used SVM and Logistic regression classifiers. We use the default parameters for.

## Results

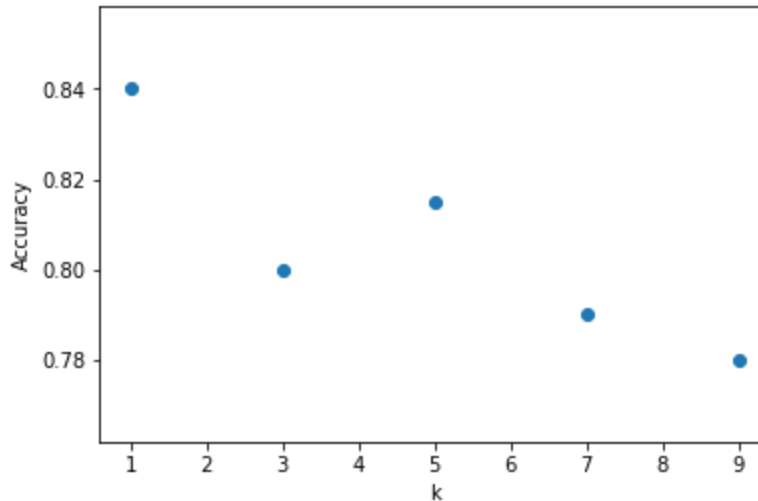
### 1. PCA + KNN

We tried for the values of  $k = [1, 3, 5, 7]$  &  $\alpha = [0.8, 0.85, 0.9, 0.95]$  and we found that the best results are for  $k = 1$  &  $\alpha = 0.95$ .



## 2. LDA + KNN

We tried for the values of  $k = [1, 3, 5, 7]$  and 39 dominant vectors.



## 3. Bonus 1

We tried the 7 train, 3 test split, and with  $\alpha = 0.75$  (best result), and we used 1-NN Classifier, and obtained the following train and test Accuracies:

Train: 100%

Test: 95.8%

## 4. Bonus 2

We used the LDA Algorithm for 2 classes, 1 = face, 0 = not face, to reduce the dimensions to 20 and then we applied SVM & Logistic Regression Classifiers.

Train: 92%

Test: 79%