# Machine Learning
## Term Project

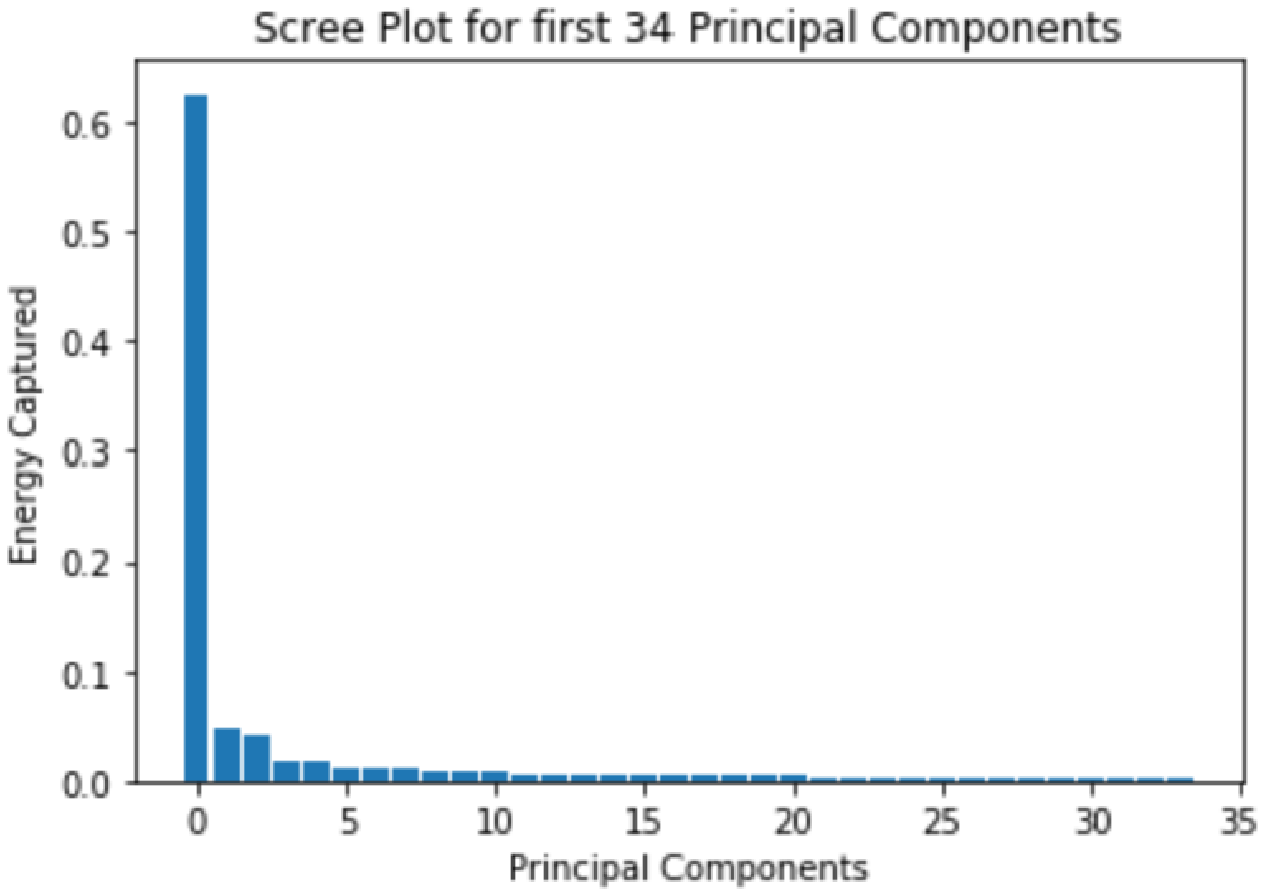
**Ebad ullah Qureshi**

**Runlin Hou**

# Clustering Problem

- Dataset: Human Activity Recognition Using Smartphones Dataset
- All the data were collected in an experiment the been carried out with a group of 30 volunteers. Each person performed six activities (WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, SITTING, STANDING, LAYING) and the data were collected according the timeline.
- Our purpose on this is to using clustering method to separate the samples into 6 clusters.

# Preprocessing & Feature Extraction




- For the preprocessing, we want all the samples to have a zero mean.
- And for the feature extraction, we want the reserved components carry more than 90% information.

# K-Means Clustering

- K-Means clustering can be divided into two kernel steps.

1) We will pick k cluster centroids
2) Then we keep doing the following works:
   (1) Assign each observation to the cluster whose means has the least squared Euclidean distance
   (2) Calculate the new means of each clusters

- K-Means  often doesn’t work so well when clusters are not round shape, since it use some kind of distance function and distance is measured from cluster center.

# GMM Clustering

- Gaussian Mixture Models(GMM) can be mathematically defined as a mixture of K gaussian distribution that means it's a weighted average of K gaussian distribution.
- So the distribution can be written like this:

$$p(x) = \sum_{k=1}^{k} \pi_k p(x|\mu_k, \sigma_k)$$

- Then after getting component distribution upon variance and mean, we can decide the probability of any samples belongs to every cluster.

# Spectral Clustering

- Spectral Clustering can be divided into three important steps.

(1) Create an affinity matrix that describe the graph based on the samples, the method we used is KNN

(2) Cut the graph into different pieces based on weights

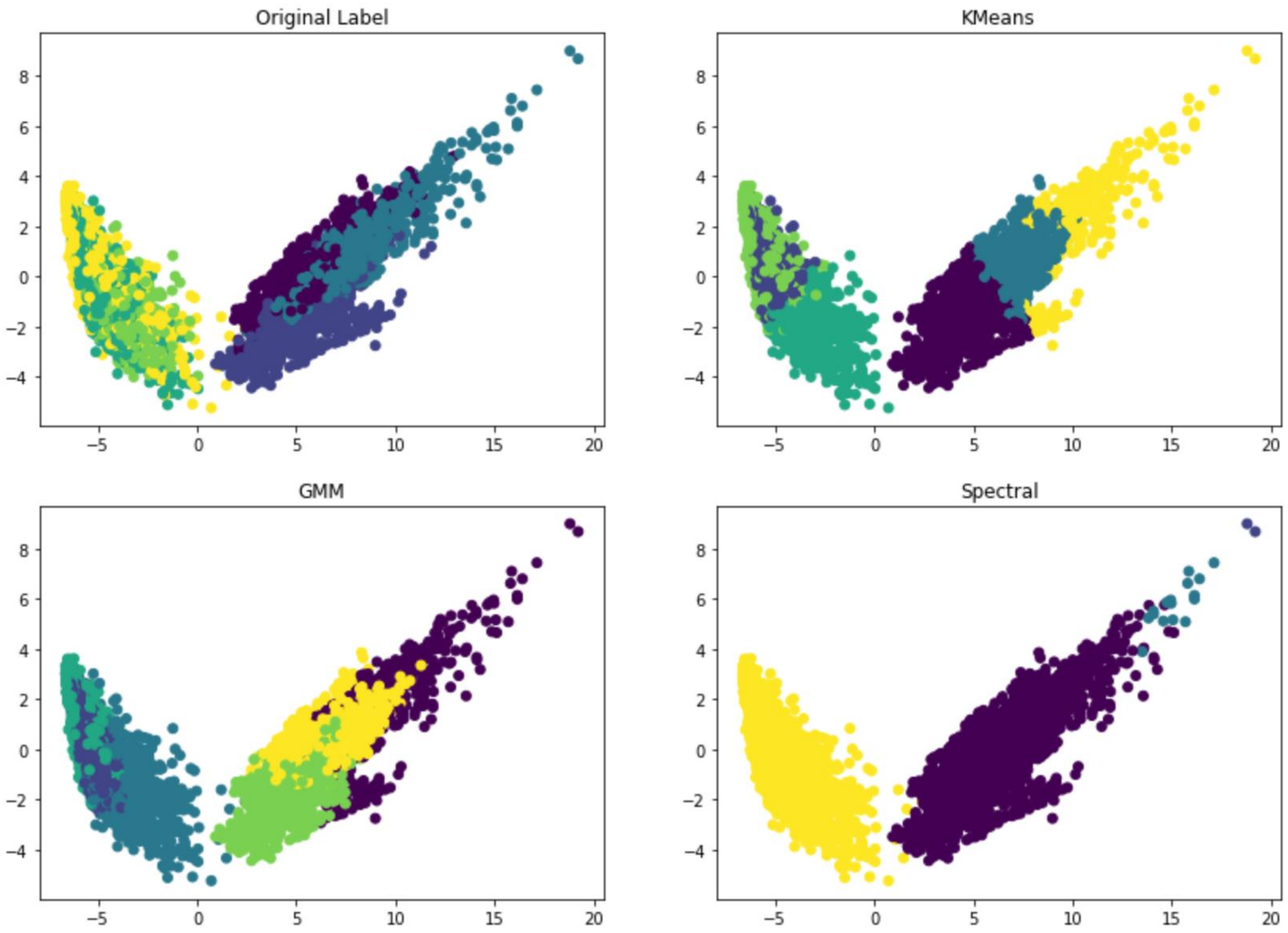
(3) Use K-Means to do the clustering.

# Performance

| K-Means Clustering | GMM Clustering | Spectral Clustering |
|---|---|---|
| 49.5% | 53.8% | 36.0% |

- Since we have the label of each samples, we choose purity to do the evaluation of our three methods.
- **Purity:** Purity is a measure of the extent to which clusters contain a single class.
- **Calculation:** given some set of clusters M and some set of classes D, both partitioning N data points
  - $\frac{\mathbf{1}}{\boldsymbol{N}} \sum_{\boldsymbol{m} \in \boldsymbol{M}} \max_{d \in D} |m \cap d|$

# Visualization



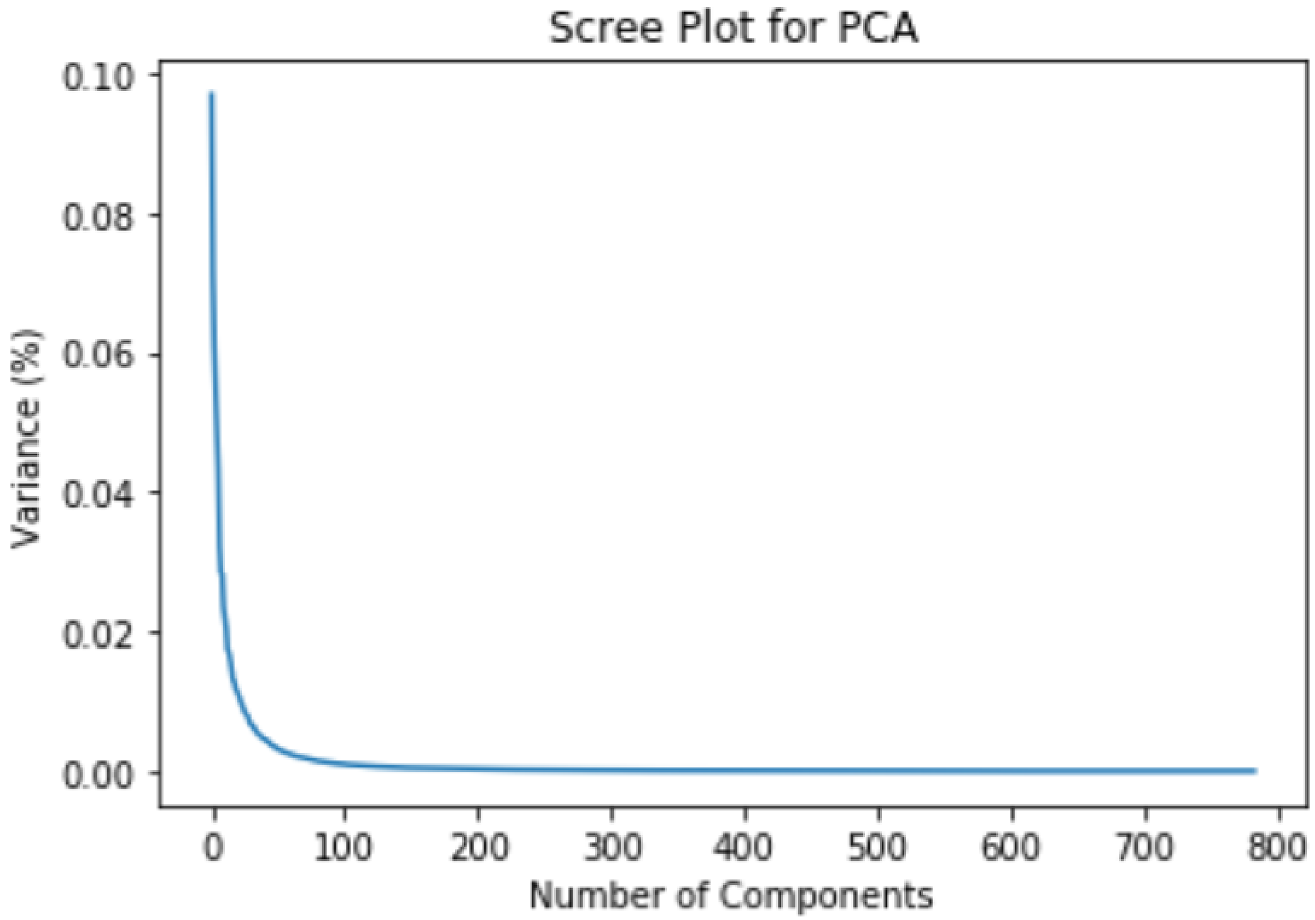

- Since the total dimensionality we have is 34, which we can not picture them all.
- So we pick the first two components to have a local view of the whole hyperplane.

# Digit Classification – Overview

- Given an image of some handwritten digit as input, the system outputs the digit on the image
- The classifier is trained on a dataset consisting of 5000 samples
- Each sample is a 28 x 28 grayscale image of some handwritten digit

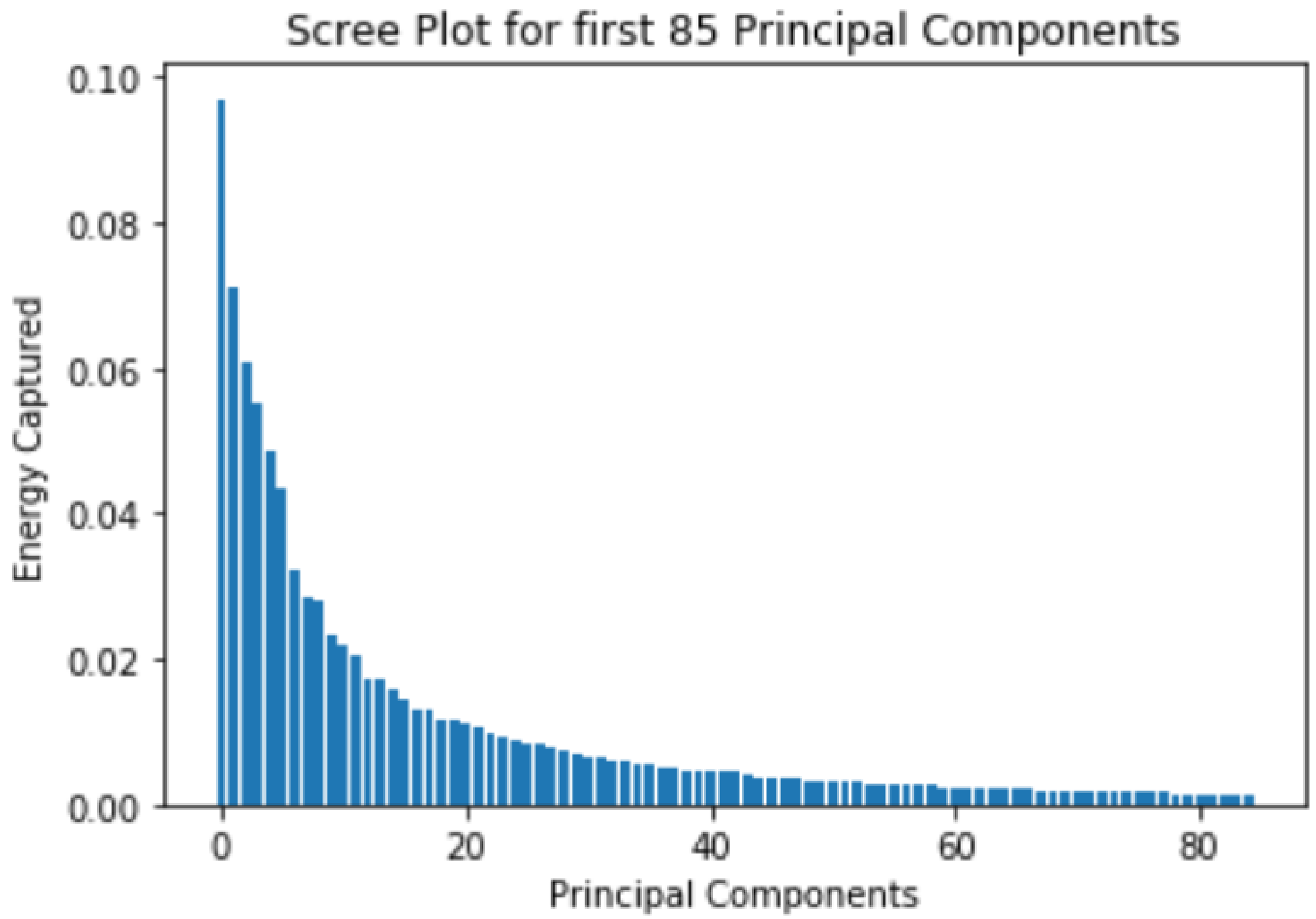
- Data $\in \mathbb{R}^{5000 \times 785}$
- 1 Label + 784 Features

# Dimensionality Reduction



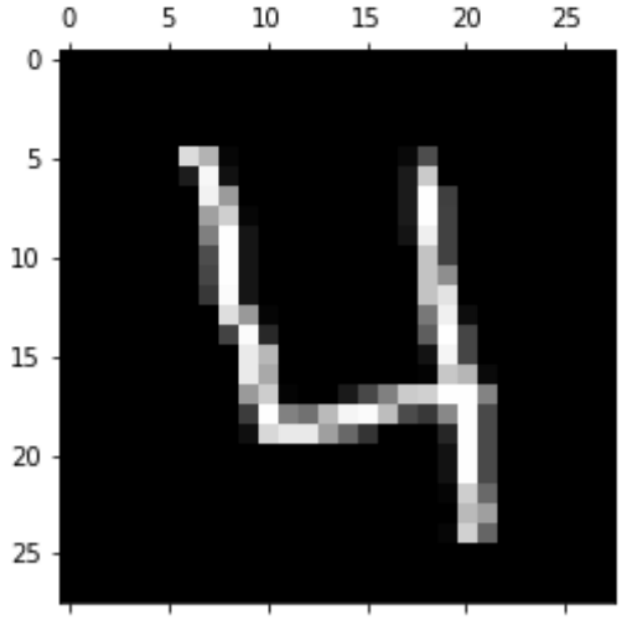

- Features in the training set are standardized to have zero mean
- Principal components are determined for the dataset
- Most of the information is contained in less than the first 100 Principal Components

# Principal Components

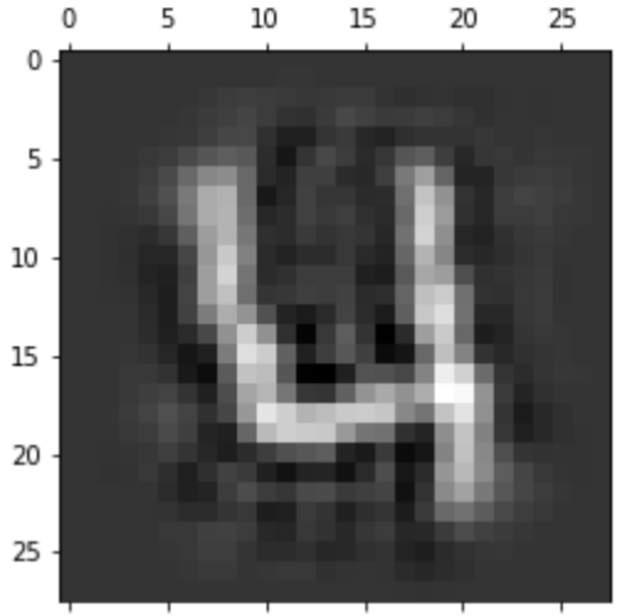



- It turns out that the first 85 Principal Components capture 90% information about the dataset

# Image Reconstructed from Principal Components




Lable for this image: 4




label for this image: 4

# *K*-Nearest Neighbors

- An image is classified by computing its Euclidean distance in 85-dimensional space and classifying it as the digit(s) it is closest to

- This is done for different values of ***K***

- Evaluated using 10-Fold cross validation Method

# First fold results in 10-Fold cross validation

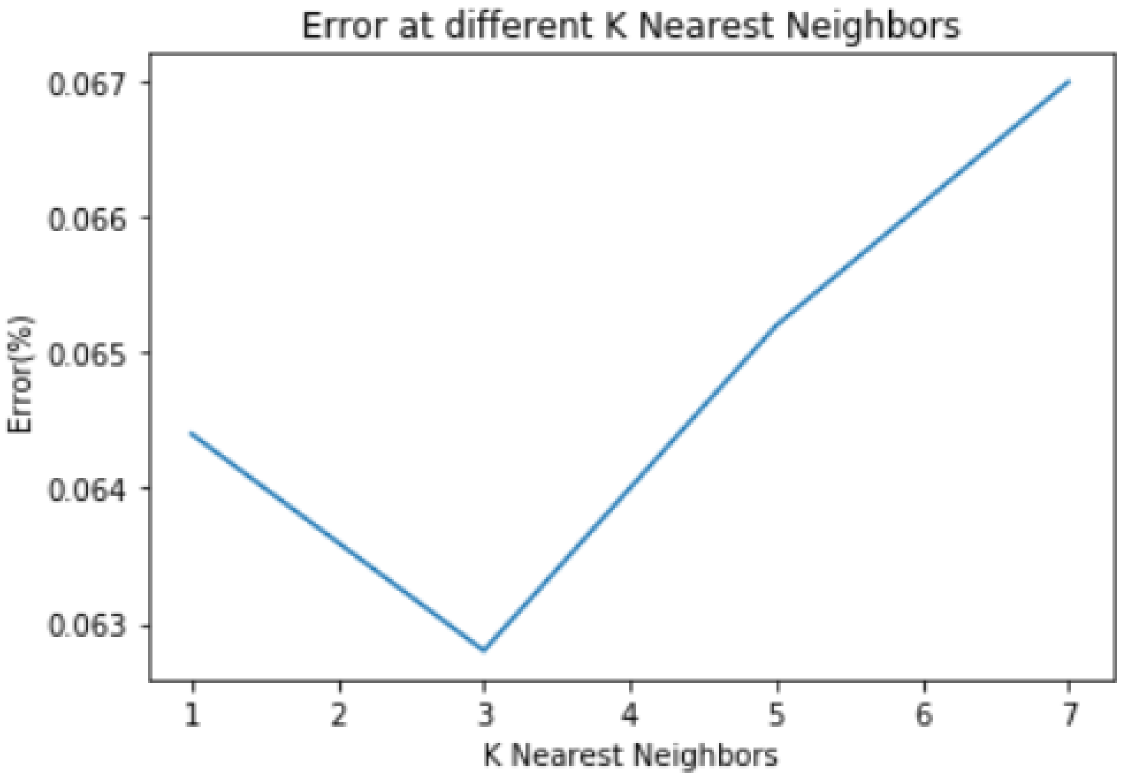
- Test Data in Fold 1 $\in \mathbb{R}^{500\times 85}$ + 500 Labels
- Training Data in Fold 1 $\in \mathbb{R}^{4500\times 85}$ + 4500 Labels

- Error observed in Fold 1 at different values of ***K***

$$
\begin{aligned}
\boldsymbol{K} &= 1 \;\Rightarrow\; \boldsymbol{\varepsilon} = 5.4\ \% \\
\boldsymbol{K} &= 3 \;\Rightarrow\; \boldsymbol{\varepsilon} = 5.4\ \% \\
\boldsymbol{K} &= 5 \;\Rightarrow\; \boldsymbol{\varepsilon} = 5.2\ \% \\
\boldsymbol{K} &= 7 \;\Rightarrow\; \boldsymbol{\varepsilon} = 6.0\ \%
\end{aligned}
$$

# Evaluation of K-Nearest Neighbor classification

- The Error rate is lowest when K is chosen to be 3
- Drawback of K Nearest Neighbors is that it requires too much computational power and is time consuming to generate an output Label



# Naïve Bayes' Classifier

$\mathbf{X} \in \mathbb{R}^{5000\times 85} \quad \mathbf{Y} \in \{0, 1, \; 2, \dots, 9\}$

In Naïve Bayes Approach to classification, all entries in X are assumed to be independent of each other

Class is assigned to a test sample as follows

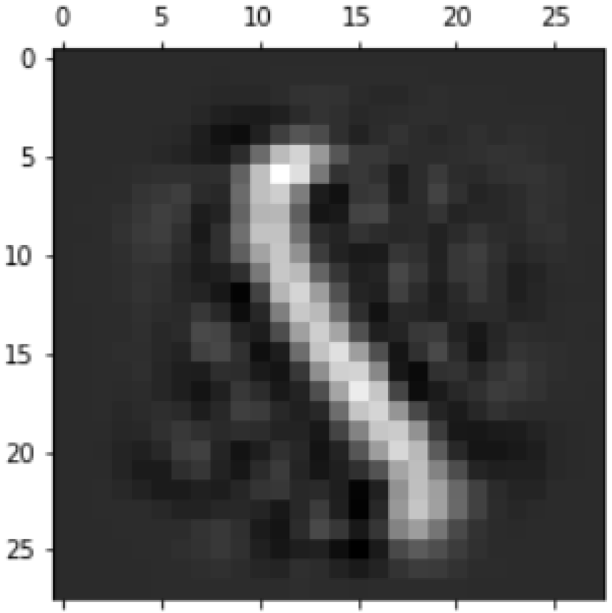
$$\hat{l} = \arg\max\big(P_{x|y}(\mathbf{X}|y = l)P(\mathbf{Y} = l)\big)$$

Probability Distribution of each class is assumed to be Gaussian

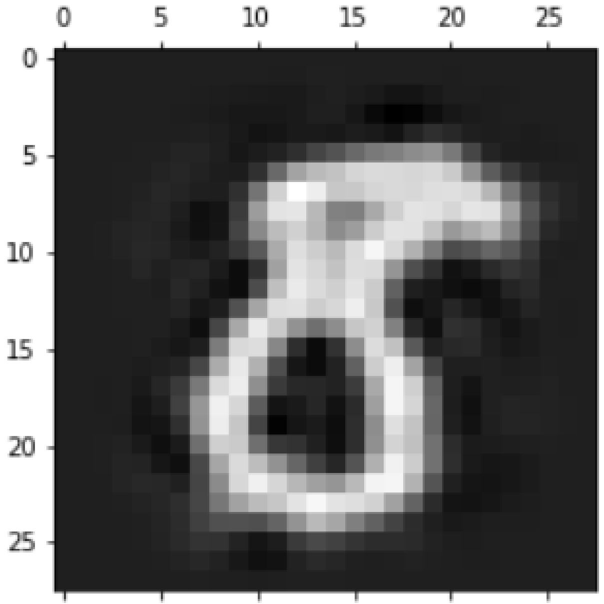
# Naïve Bayes' Approach

- Partition Data set in 70:30 ratio randomly to get train and test datasets respectively
- Split Training data w.r.t class labels **L**
- Compute Prior Probabilities of each class **P(Y=L)**
- Compute mean vector and variances for each class
- Covariance matrix of **X** is the diagonal matrix containing variances of each separate feature
- Define the Likelihood function (Gaussian) for each class
- A Test image is classified as class **L** such that **L** has maximum aposterior probability

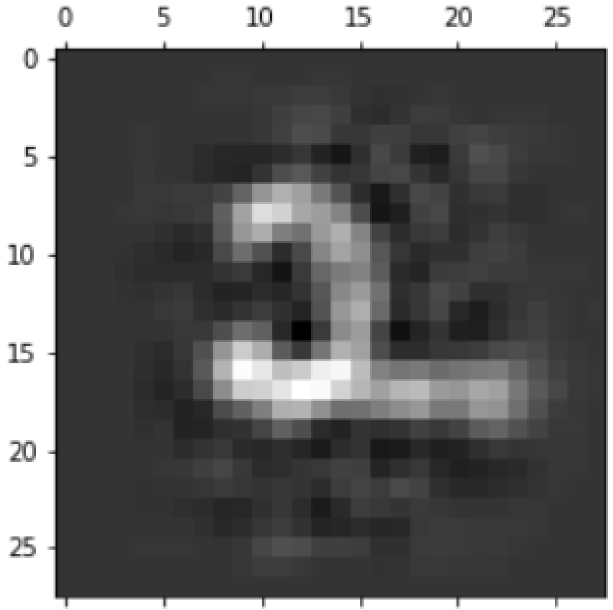
Actual Label: [1]
Output Label: [5]
<Figure size 432x288 with 0 Axes>




Actual Label: [8]
Output Label: [5]
<Figure size 432x288 with 0 Axes>




Actual Label: [2]
Output Label: [9]
<Figure size 432x288 with 0 Axes>



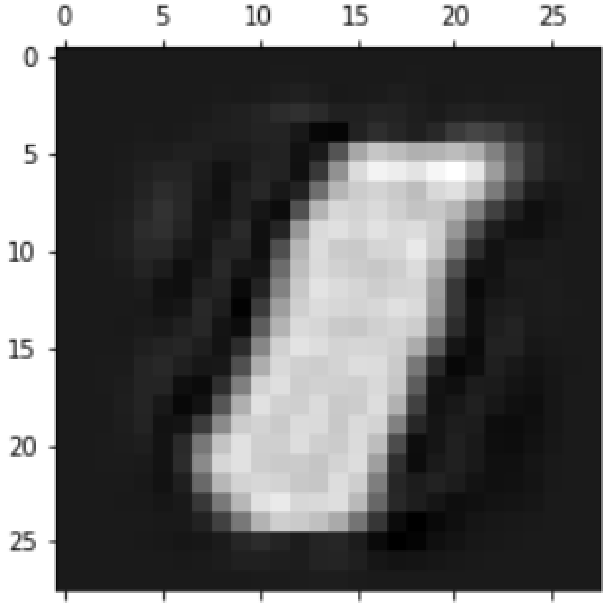

# Evaluation of Naïve Bayes' classifier

- Error 13.13 %

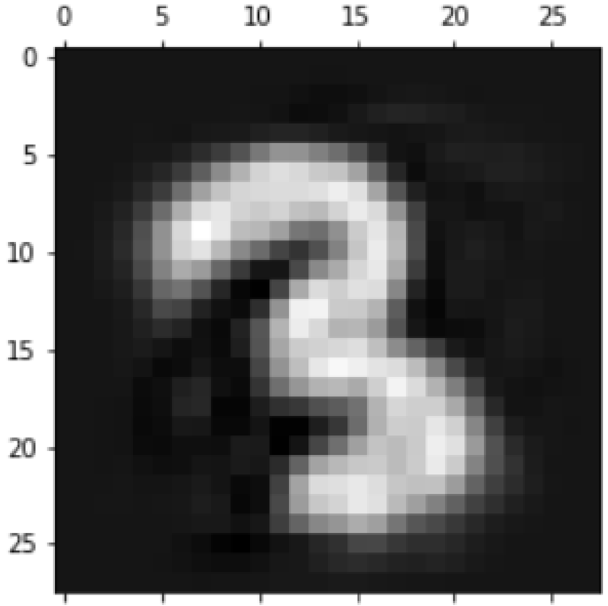
# Quadratic Discriminant Analysis

- Like Naïve Bayes’ classifier, Quadratic Discriminant Analysis also uses Probability distribution to classify digits
- It doesn’t assume that all features are independent of each other
- The Off-diagonal elements in the Covariance Matrix in case of QDA are not zeros

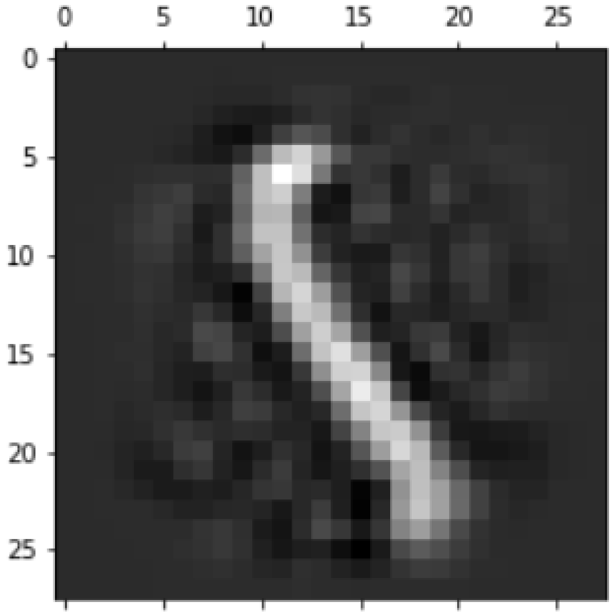
```
Actual Label: [1]
Output Label: [8]
<Figure size 432x288 with 0 Axes>
```




```
Actual Label: [3]
Output Label: [8]
<Figure size 432x288 with 0 Axes>
```




```
Actual Label: [1]
Output Label: [8]
<Figure size 432x288 with 0 Axes>
```




## Evaluation of Quadratic Discriminant Analysis

- **Error 6.33 % (Best Performance)**

Thank You