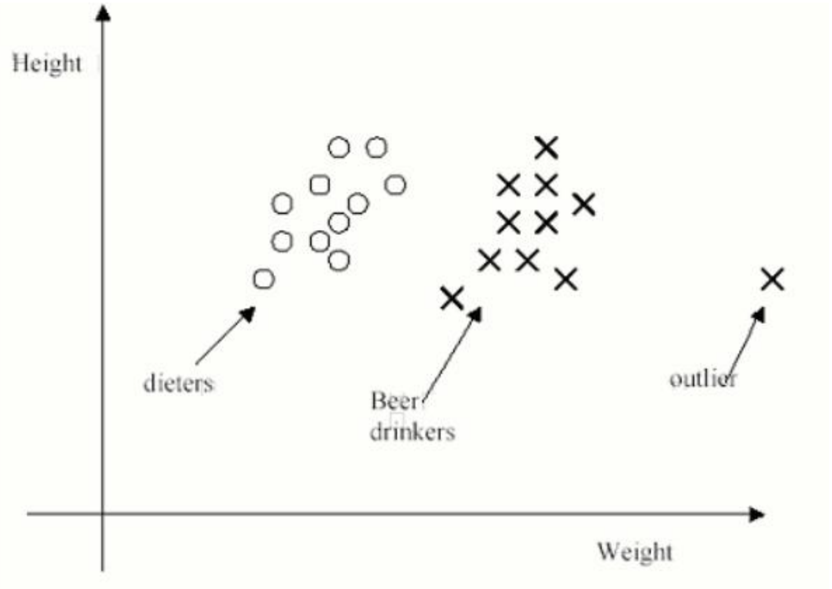
Project Development Phase
Sprint – 1

Date	15 October 2022
Team ID	PNT2022TMID24826IBM
Project Name	Project - A novel Method for Handwritten Digit Recognition System

Pattern Recognition

Pattern recognition system consists of two-stage process. The first stage is feature extraction and the second stage is classification. Feature extraction is the measurement on a population of entities that will be classified. This assists the classification stage by looking for features that allows fairly easy to distinguish between the different classes. Several different features have to be used for classification. The set of features that are used makes up a feature vector, which represents each member of the population. Then, Pattern recognition system classifies each member of the population on the basis of information contained in the feature vector. The following is an example of feature vectors that have been plotted on a graph.



Thos shows two clusters of points. Each of them corresponds to one of the classes. These classes are fully separated and can be easily distinguished.

- **Bayesian decision theory.**

The Bayesian decision theory is a system that minimizes the classification error. This theory plays a role of a prioi. This is when there is priority information about something that we would like to classify. For example, suppose we do not know much about the fruits in the conveyer belt. The only information we know is that 80% of the fruit in the conveyer belt are apples, and the rest of them are oranges. If this is the only information we have, then we can classify that a random fruit from the Conveyer belt is apple. In this case, the prior information is the probability of either an apple or an orange is in the conveyer belt. If we only have so little information, then we would have the following rule: **Decide"apple'if P (apple) > P (orange), otherwise decide"orange'**

Here, P (apple) is the probability of being an apple in the conveyer belt. This means that $P(apple) = 0.8$ (80%). This is probably strange, because if the above rule is used, then we are classifying a random fruit as an apple. But if we use this rule, we will be right 80% of the time.

This is a simple example and can be used to understand the basic idea of pattern recognition. In real life, there will be a lot more information given about things that we are trying to classify. For example, we know that the color of the apples is red. Therefore if we can observe a red fruit, we should be able to classify it as an apple. We can have the probability distribution for the color of apples and oranges.

Let wapp represent the state of nature where the fruit is an apple, let wora represent the state of nature where the fruit is an orange and let x be a continuous random variable that represents the color of a fruit. Then we can have the expression $p(x|wapp)$ representing the density function for x given that the state of nature is an apple.

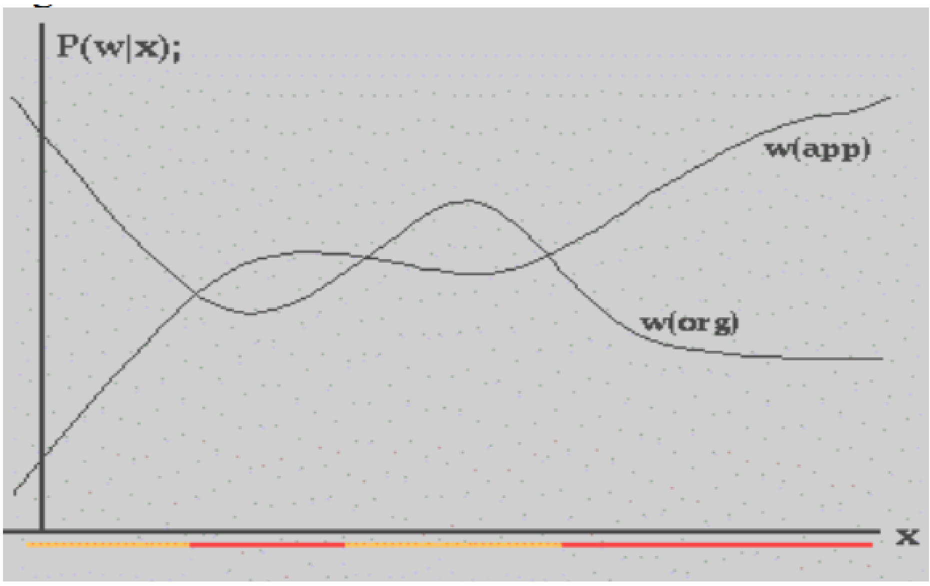
In a typical problem, we would be able to calculate the conditional densities $p(x|wj)$ for j so it will be either an apple or an orange. We would also know the prior probabilities $P(wapp)$ and $P(wora)$. These represent the total number of apples versus oranges in the conveyer belt. Here we are looking for a formula that will tell us about the probability of a fruit being an apple or an orange just by observing a certain color x. If we have the probability, then for the given color that we observed, we can classify the fruit by comparing it to the probability that an orange had such a color versus the probability that an apple had such a color. If we were more certain that an apple had such a color, then the fruit would be classified as an apple.

So, we can use Baye's formula, which states the following: **$P(wj|x) = p(x|wj) P(wj)/p(x)$**

What the formula means is that using a priori information, we can calculate the a posteriori probability of the state of nature being in state w hat we have given that the feature value x has been measured. So, if we observe a certain x for a random fruit in the conveyer belt, then by calculating $P(wapp/x)$ and $P(worg /x)$. we would decide that the fruit is apple if the first value is greater than the second one and if $P(worg/x)$ is org greater, then we would decide that the fruit is orange. So, the Bayesian decision rule can be stated as: **Decide worg if $P(worg|x) > P(wapp|x)$, otherwise, decide wapp ,**

Since $p(x)$ occurs on both sides of the comparison, the rule can also be equivalent to the following rule: **Decide worg if $p(x|worg)P(worg) > p(x|wapp)P(wapp)$, otherwise decide wapp**

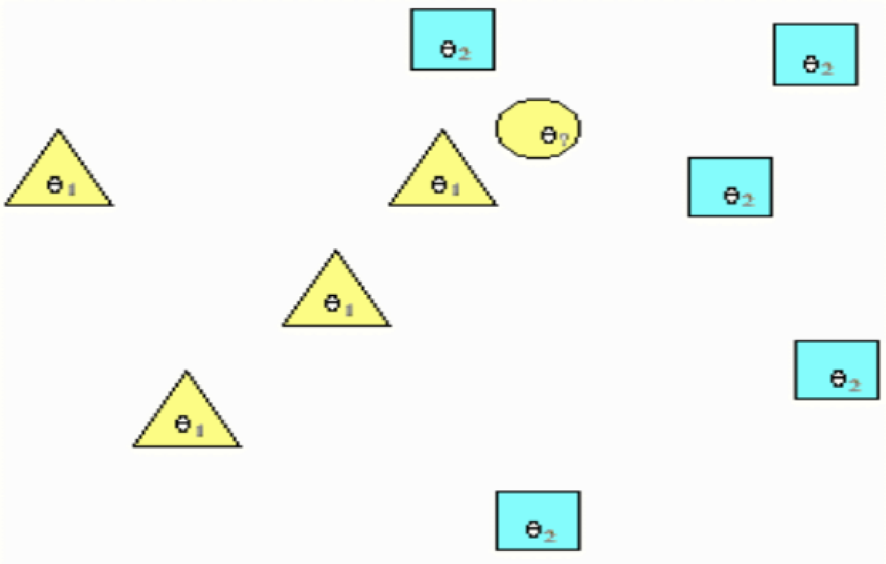
The following graph shows the a posteriori probabilities for the two-class decision problem. For every x, the posteriors has to sum to 1. The red region on the x axes represents the values for x for which would decide as ”apple‘. The orange region represents values for x for which would decide as ”orange‘.



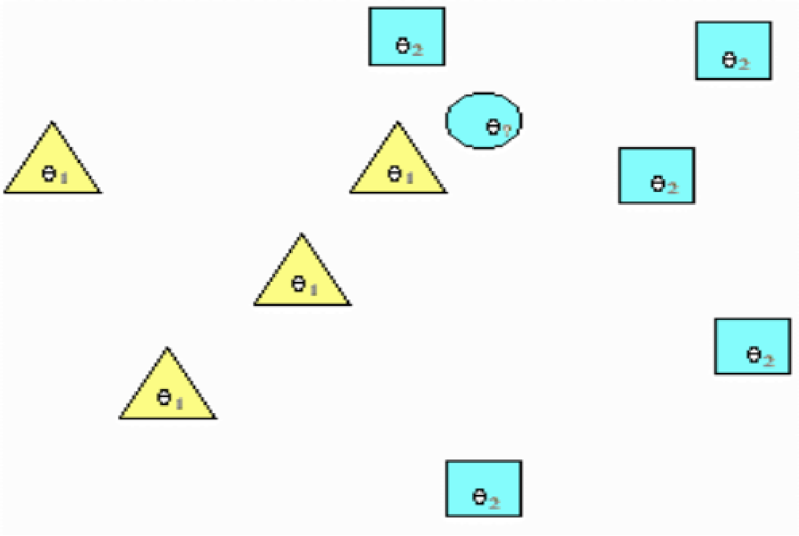
The probability that we would probably make an error is any minimum of the 2 curves at any point, because it represents the smaller probability that we did not pick. The formula for the error is the following: **P(error|x) = min[p(wapp |x), p(worg |x)].**

- ## Nearest Neighbor rule

The Nearest Neighbor (NN) [10]rule is used to classify handwritten characters. The distance measured between the two character images is needed in order to use this rule. Without a priori assumptions about the distributions from which the training examples are drawn, the NN [2]rule achieves very high performance. The rule involves a training set of both positive and negative cases. A new sample is classified by calculating the distance to the nearest training case. The sign of the point determines the classification of the sample. The following figure shows an example of NN rule



In this example there are two classes θ1, which are the yellow triangles and θ2 which are blue squares. The yellow circle represents the unknown sample X.We can see that the unknown samples nearest neighbor is from class θ1 therefore it is labeled as class θ1.When the amount of pre-classified points is large, it is good to use the majority vote of the nearest k neighbors instead of the single nearest neighbor. This method is called the k nearest neighbor (k-NN) rule. The k-NN[2] rule extends the idea by taking and assigning the k nearest points the sign of the majority. It is common to choose k small (with respect to the number of samples) so that the points are closer to x to give accurate estimate of the true class of x.If the k values are large, it will help reduce the effects of noisy points within training data set. Also, large k values will minimize the probability of misclassifying x. The following figure shows an example of k-NN rule with k value equal 3:



As before, there are two classes: θ1 which are the yellow triangles, and θ2 which are the blue squares. The blue circle represents the unknown sample x. We see that two of its nearest neighbors are from class θ2 , so it is labeled as class θ2.

• Linear Classification Discrimination

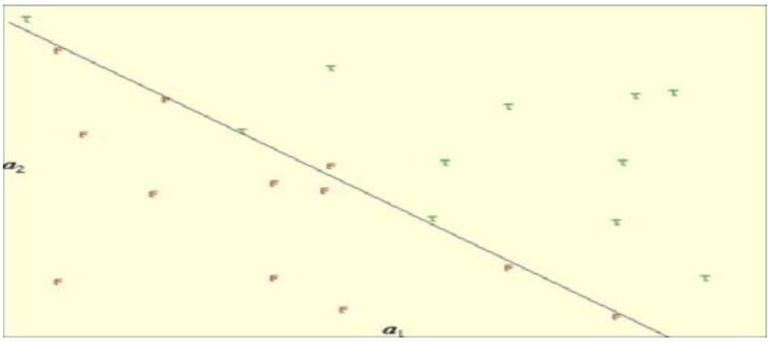
The goal of Linear Classification is to assign observations into the classes. This can be used to establish a classifier rule so that it can assign a new observation into a class. In another words, the rule deals with assigning a new point in a vector space to a class separated by a boundary. Linear classification provides a mathematical formula to predict a binary result. This result is a true or false (positive or negative) result or it can be any other pair of characters. In general, we will assume that our Results are Boolean variable. To do this prediction, we use a linear formula over the given input data. We refer this as inputs. The linear form is computed over the inputs and the result is compared against a basis constant. It is depending on the result of the comparison that we would be able to predict true or false. The following is the equation that can be stated as the discriminator:

$$\mathbf{a1\ x1 + a2\ x2 + ... + an\ xn > x0}$$

Here, a1, a2 are the variables that correspond to one observation. and x1,x2 together with x0 are the solution vector plus the basis constant. Corresponding to each of the input vector a, there is a variable b. This variable b is the dependent Boolean variable. We refer this as the output. We normally have many m data points (a row vector of inputs a and the corresponding output, b). For convenience, we place all the data in matrices and vectors. A denotes the matrix of the inputs and it is in the dimension of m by n. The m Boolean outputs will be stored in a vector B. Now, the problem of linear classification can be stated in terms of the matrices and vectors. For example: we want to find x and x0 such that:

$$\mathbf{(Ax > x0\ )\ equiv\ B}$$

This equivalence means that every row of the left-hand-side is a relation. The relation for the given data will be true or false. This has to match the corresponding B value. When we have more data points than columns, this is m > n, the problem does not have a solution. We can not find a vector x that will satisfy all the true values. Therefore we try to find a solution that can match vector B as good as possible. The best matching means that there is a minimum number of errors or there is a weighted minimum of errors. There will be a weight that is assigned to the true errors and another weight that is assigned to the false errors. The idea of obtaining good classification is to be able to use it to predict the output for new data points. Here, the discriminator is a prediction formula, A and B are the training data, and x and x are the parameters of the model fitted to the training 0 data. The following figure shows the main components of linear classification:



Here, true are marked with a green T and false are marked with a red F. The inputs are two independent variables, a1 and a2 . Here we use them to position the points in the plane. There are eleven Ts (positives) and twelve Fs (negatives). This linear classification in two dimensions is a straight line. The straight line drawn is a very good classification because the line separates most of the points correctly. We can see that all the Ts are on one side of the graph and all the Fs, but one, are on the other side of the graph. Therefore this classification has one "false positive" and no "false negatives". A false positive is a data point that is classified as positive but it is negative and vice versa for false negatives.