MA4207 FINAL PRESENTATION

HOUSE TWENTY

Presented by

Abhijit Kundu

Abhisek Sarkar

Narayan Biswas

INTRODUCTION

The dataset is for 20 homes located near to the town of Loughborough in the East Midlands region of the UK. A building survey was carried out at each home, collecting data on building geometry, construction materials, occuapancy and energy services. Each home has a selection of the following sensors and devices installed: - CurrentCost mains clamps, to measure household mains electrical power load, Replacement gas meters, to measure household mains gas consumption.

TIMELINE September 2013 to February 2014: Building surveys were carried out and monitoring sensorswere placed in the buildings at or shortly after this time. June 2014 and October 2014: Smart Home deviceswere installed in the buildings

DATA STATISTICS
Number of homes: 20
Number of spaces (rooms): 389
Number of radiators: 252
Number of showers: 34
Number of appliances: 618
Number of light bulbs: 672
Number of fixed heaters: 19
Number of surfaces: 2237
Number of openings: 970
Number of sensors: 1,567
Number of variables recorded by sensors and devices: 2,457
Number of time series readings: 25,312,397

We use data of House 20 to make one dataset. There are two classes. The first class is household aggregate usage of electricity. The second class is aggregate electricity load of Tumble Dryer and Washing Machine.

Pattern Classification

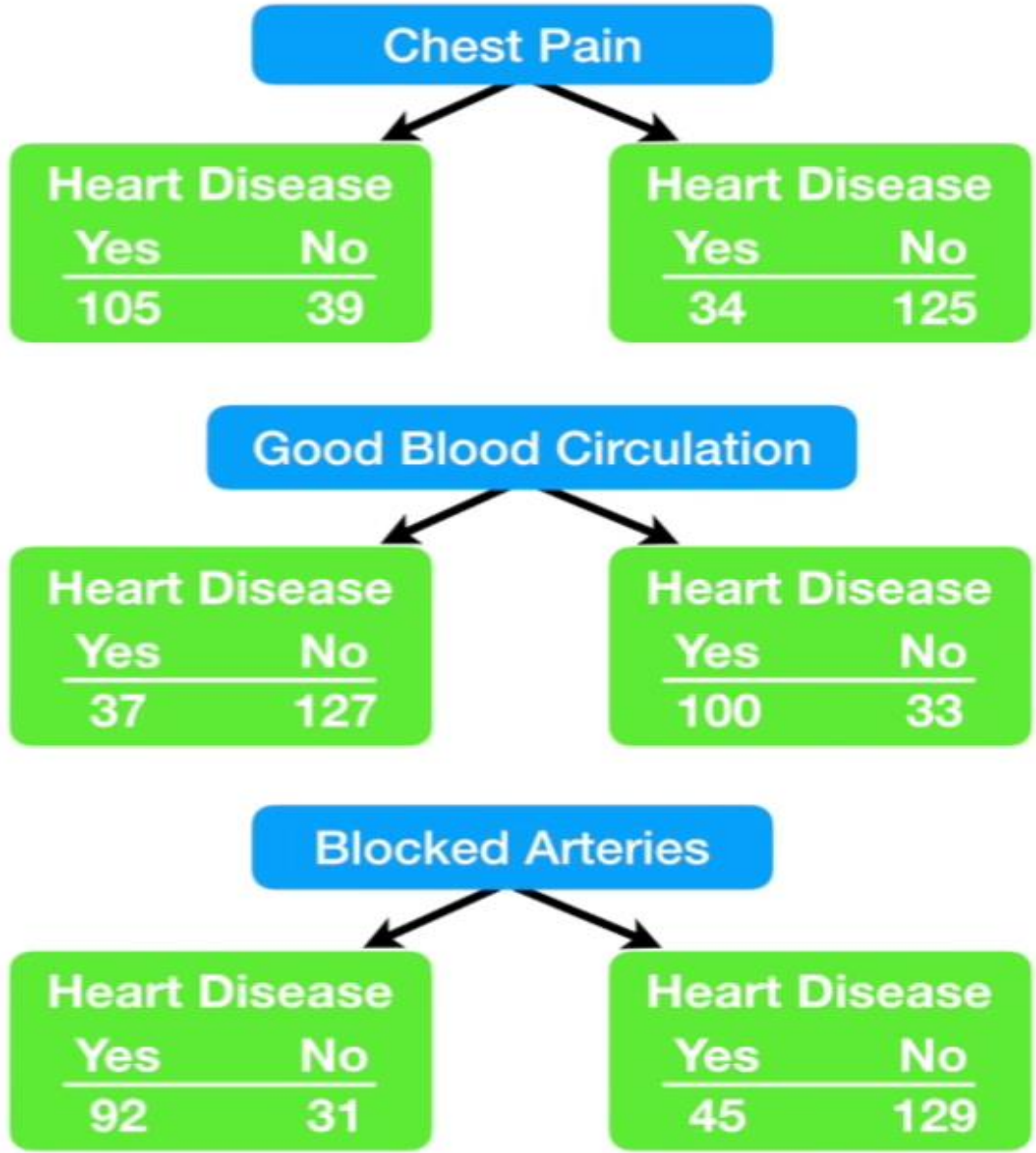
Decision Tree

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

Example

We try to find out Heart Dieases by Chest Pain, Good Blood Circulation and Blocked Arteries. We have some data of patients.

Chest Pain	Good Blood Circulation	Blocked Arteries	Heart Disease
No	No	No	No
Yes	Yes	Yes	Yes
Yes	Yes	No	No
Yes	No	???	Yes
etc...	etc...	etc...	etc...


Chest Pain
Heart Disease
Yes No
105 39
Heart Disease
Yes No
34 125
Good Blood Circulation
Heart Disease
Yes No
37 127
Heart Disease
Yes No
100 33
Blocked Arteries
Heart Disease
Yes No
92 31
Heart Disease
Yes No
45 129

We take 303 patients out of those 144 patients have chest pain and 159 patients don’t have any chest pain. Out of 144 patients those have chest pain 105 have heart disease and 39 don’t have heart disease. Similarly out of 159 patients those don’t have chest pain 34 patients have heart disease and 125 patients don’t have heart disease.
Similar things follows for Good Blood Circulation and Blocked Arteries.
So we see any of those features cannot perfectly predict Heart Disease. So there have some impurity.

There are bunch of ways we can measure impurity one of them is ”Gini”.

For the Chest Pain yes leaf, the Gini impurity is $= 1 - (\frac{105}{105+39})^2 - (\frac{39}{105+39})^2 = 0.395$

For the chest pain no leaf, the Gini impurity is $= 1 - (\frac{34}{34+125})^2 - (\frac{125}{34+125})^2 = 0.336$

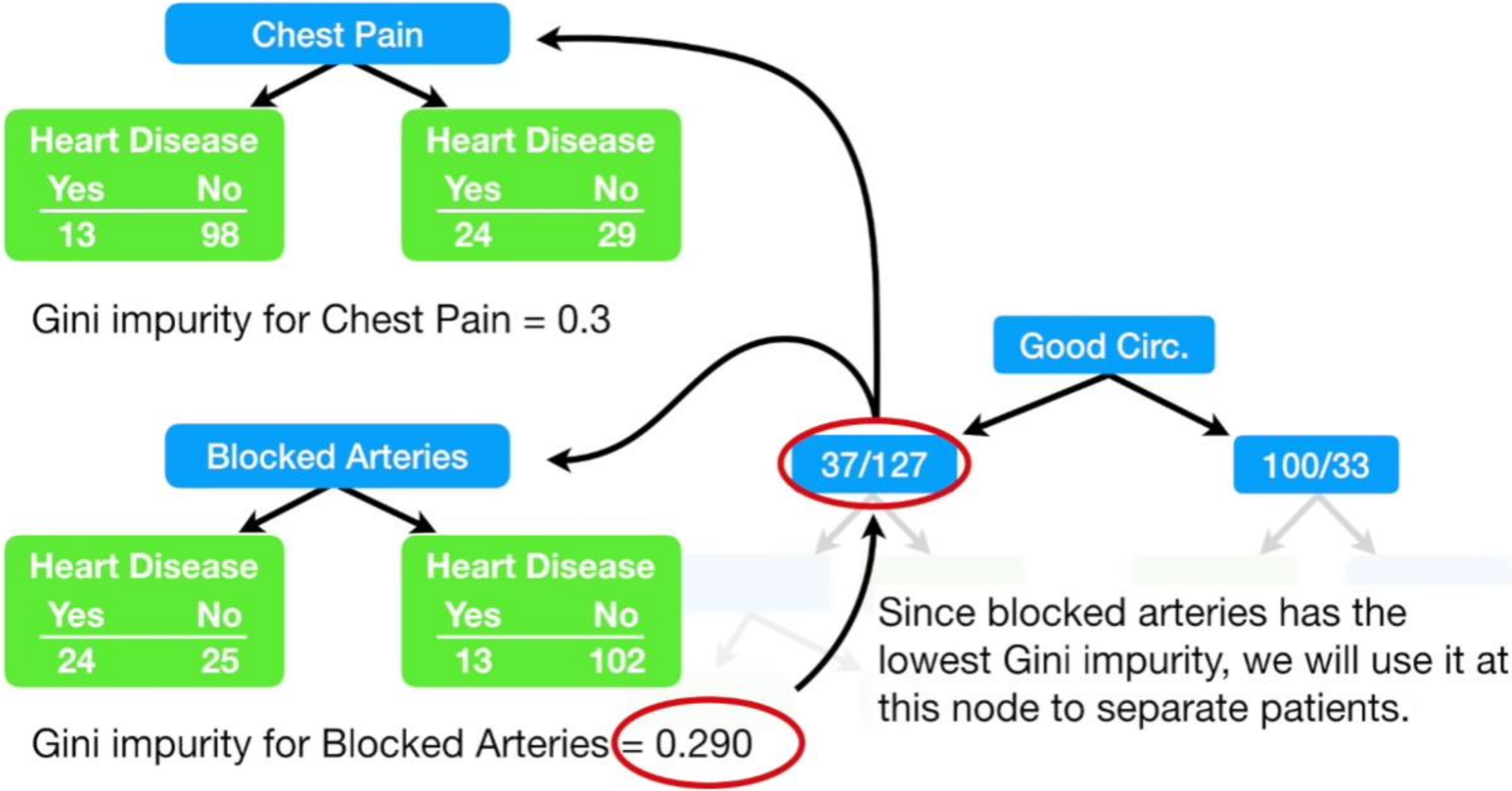
Total Gini Impurity for Chest Pain $= \frac{144}{144+159} \times 0.395 + \frac{159}{144+159} \times 0.336 = 0.364$

Similarly Total Gini Impurity for Good Blood Circulation $= 0.360$

Total Gini Impurity for Blocked Arteries $= 0.381$.

As Good Blood Circulation has the lowest impurity so it separates patients with and without heart disease the best. So we will use it at the root of the tree.

Now taking Good Blood Circulation as the root of the tree we try to calculate which feature has the lowest Gini Impurity.

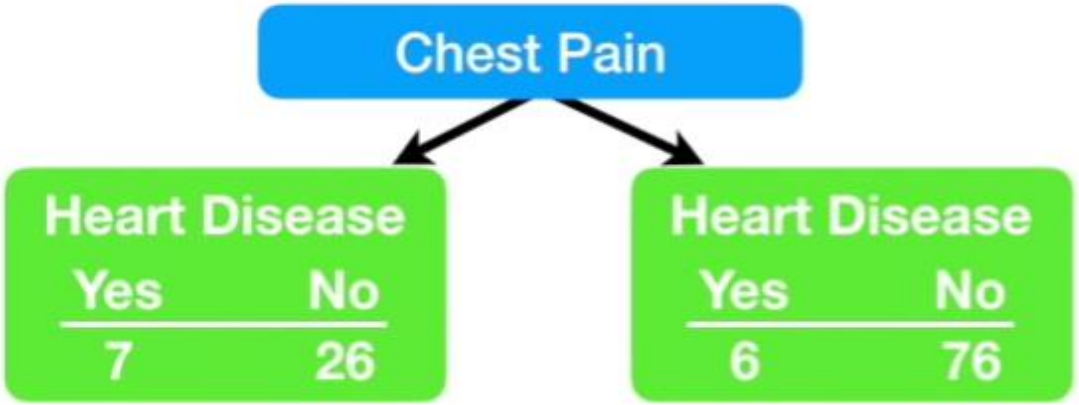

Chest Pain
Heart Disease
Yes No
13 98
Heart Disease
Yes No
24 29
Gini impurity for Chest Pain = 0.3
Blocked Arteries
Heart Disease
Yes No
24 25
Heart Disease
Yes No
13 102
Gini impurity for Blocked Arteries = 0.290
Good Circ.
37/127
100/33
Since blocked arteries has the lowest Gini impurity, we will use it at this node to separate patients.

So we get that the Gini Impurity for Blocked Arteries is lower than the Gini Impurity of Chest Pain. So wechoose Blocked Atreries for separation the patient.

So firstly we separate using Good Blood Ciculation if yes then we go with Blocked Arteries. So all we have left is Chest Pain.

Now in the case of Blocked Arteries yes we get the Gini Impurity for Chest pain = 0.32 and without using Chest Pain Gini Impurity = 0.34. So finally we use chest pain to separate it.

Now we see the case when Good Circulation is yes Blocked Arterities is no.



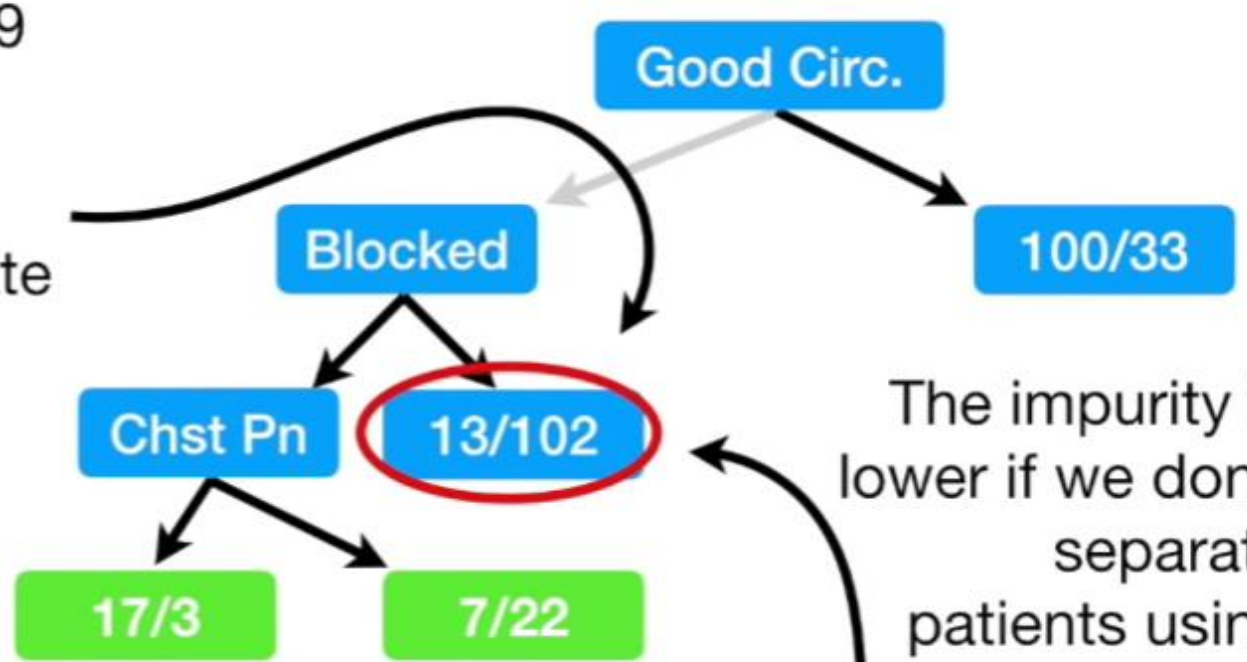
Gini impurity for Chest Pain = 0.29

The Gini impurity for this node, before using chest pain to separate patients is…

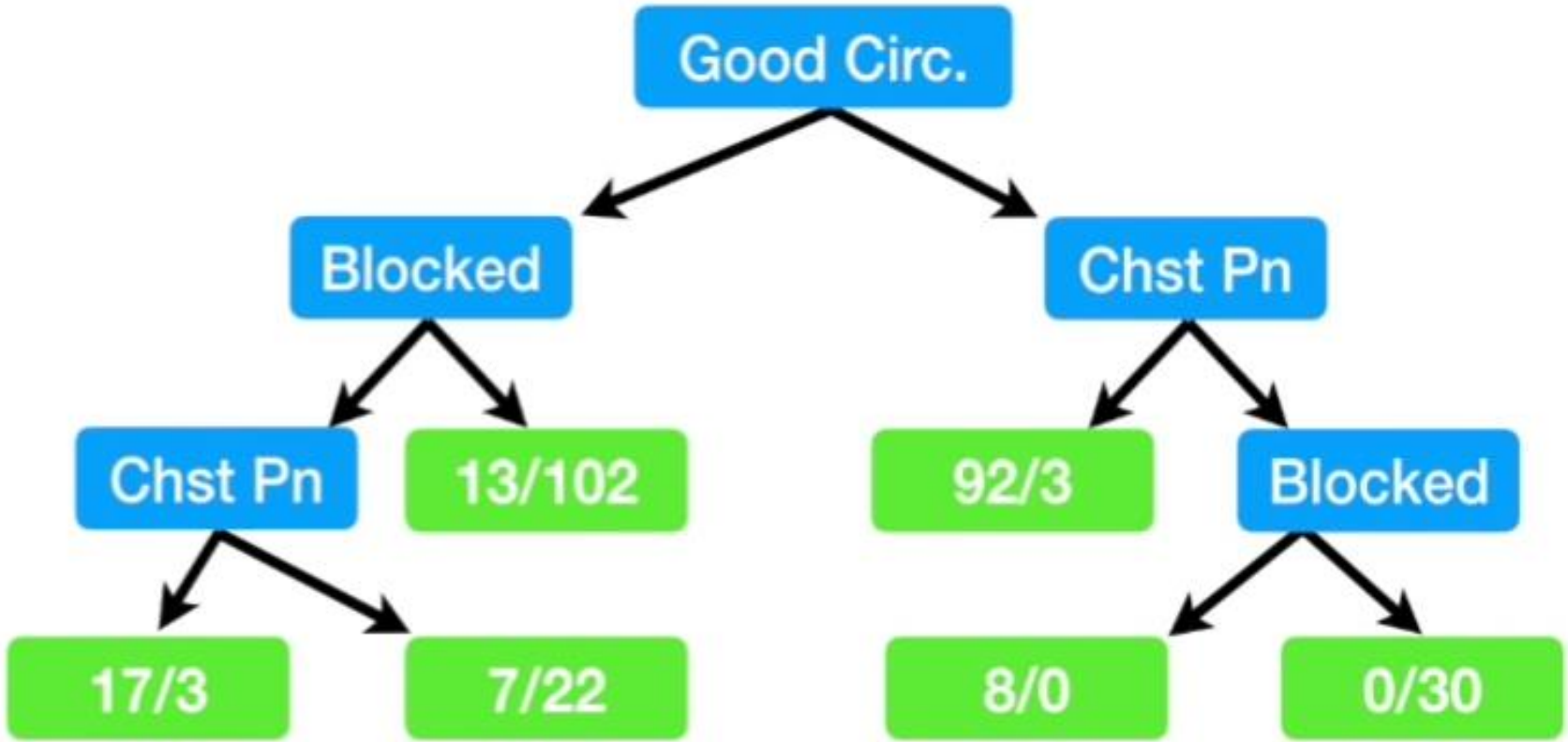
= 1 - (the probability of "yes")^2
- (the probability of "no")^2

$$= 1 - \left(\frac{13}{13 + 102}\right)^2 - \left(\frac{102}{13 + 102}\right)^2$$

= 0.2



The impurity is lower if we don't separate patients using Chest Pain.


Good Circ.
Blocked
Chst Pn
Chst Pn
13/102
92/3
Blocked
17/3
7/22
8/0
0/30

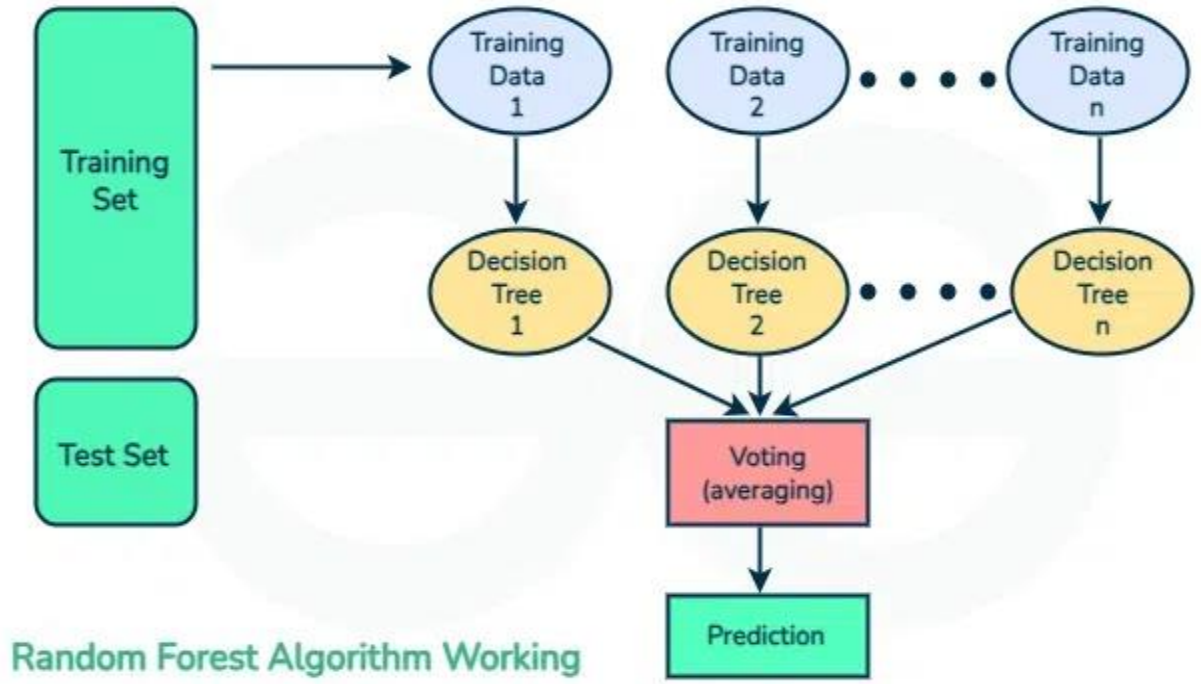
Using the above method we get the accuracy = 57%.
So the accuracy is bit low.

So we have to focus on some other method also for getting a better accuracy for the model.

We use Random Forest, KNN method, Logistic Regression also.

In this case for Random Forest we get the accuracy = 71%
In this case for KNN we get the accuracy = 52%
In this case for Logistic regression accuracy = 75%

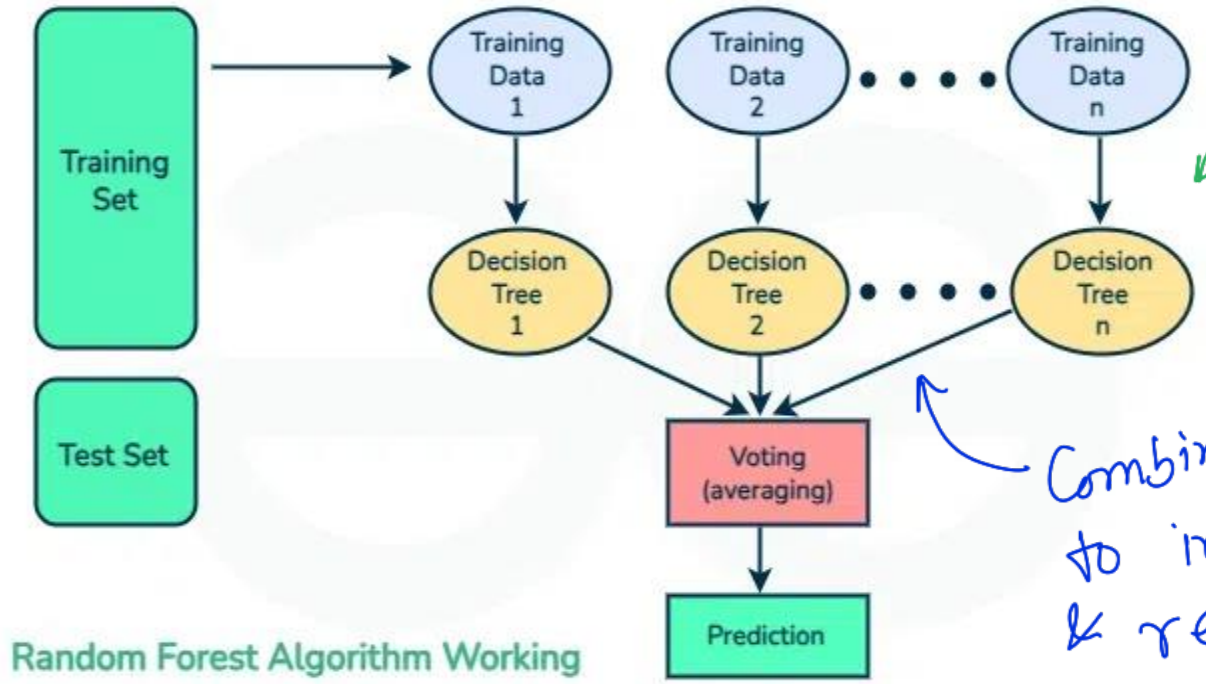
Random Forest Algorithm (71% accuracy)



Random Forest Algorithm Working

Random Forest Algorithm

ensemble learning algorithm



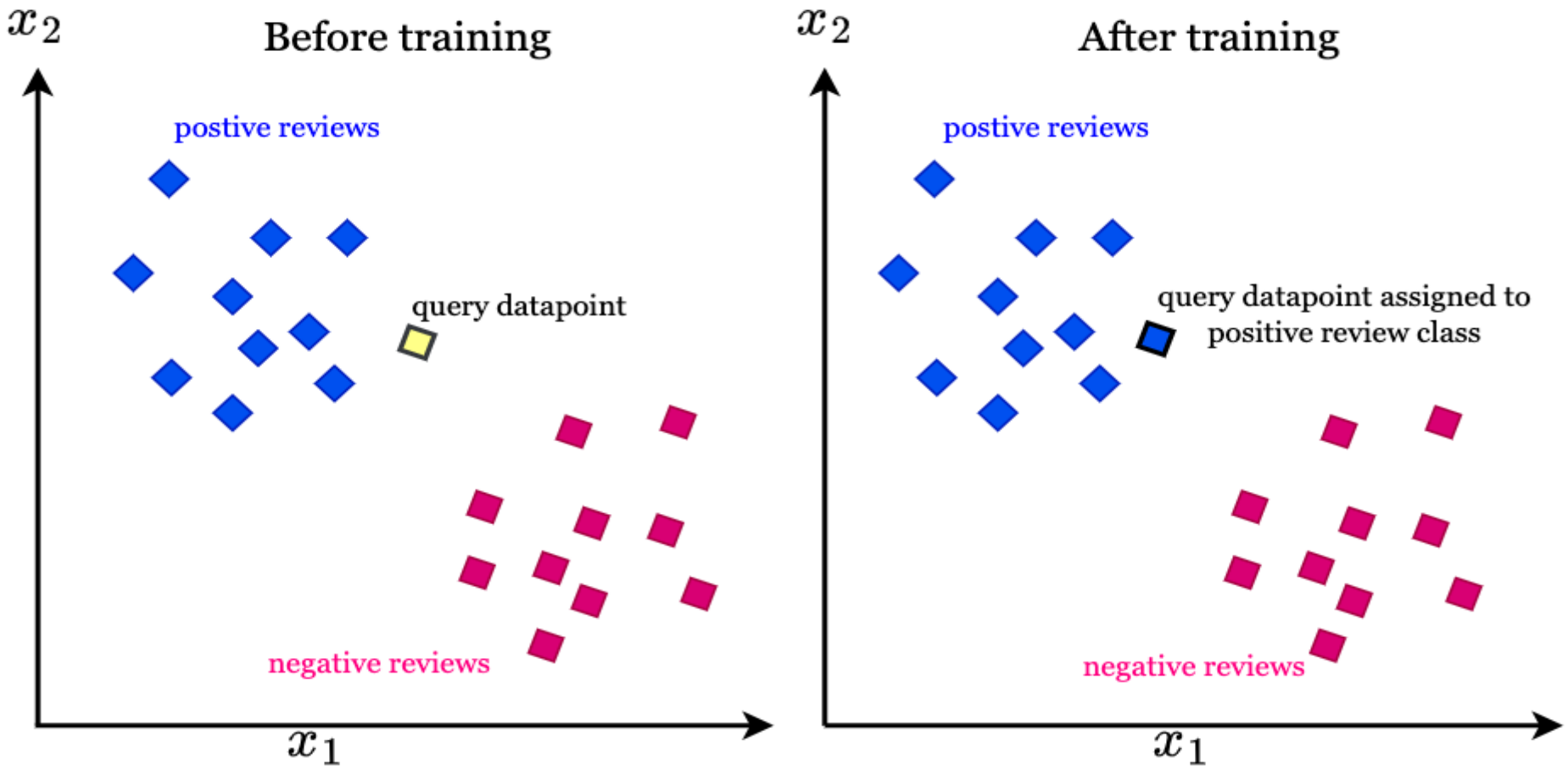
Random Forest Algorithm Working

builds multiple decision trees

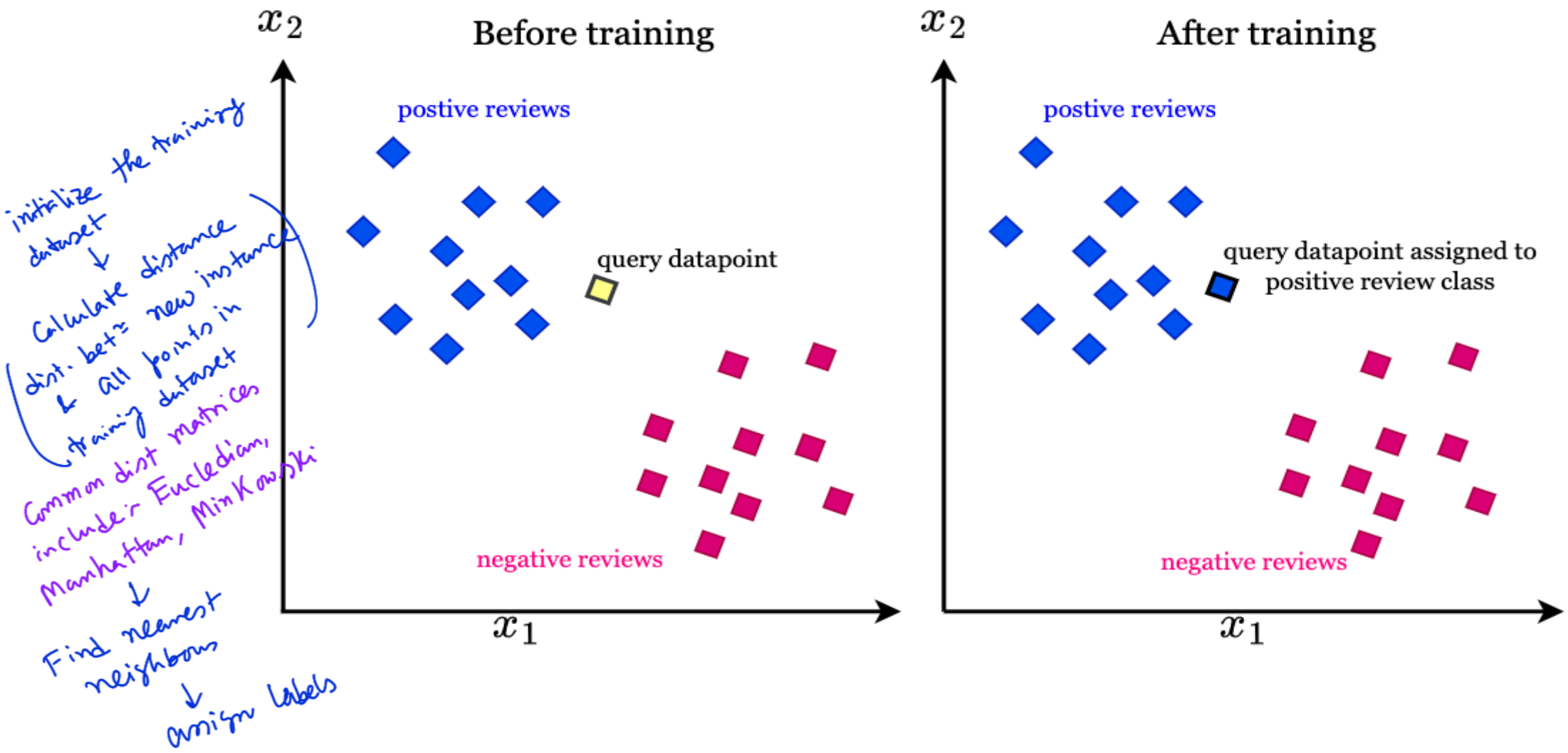
Combines their prediction to improve accuracy & reduce overfitting

Can handel Complex datasets
maintain robustness against overfitting
Provide insights into feature importance

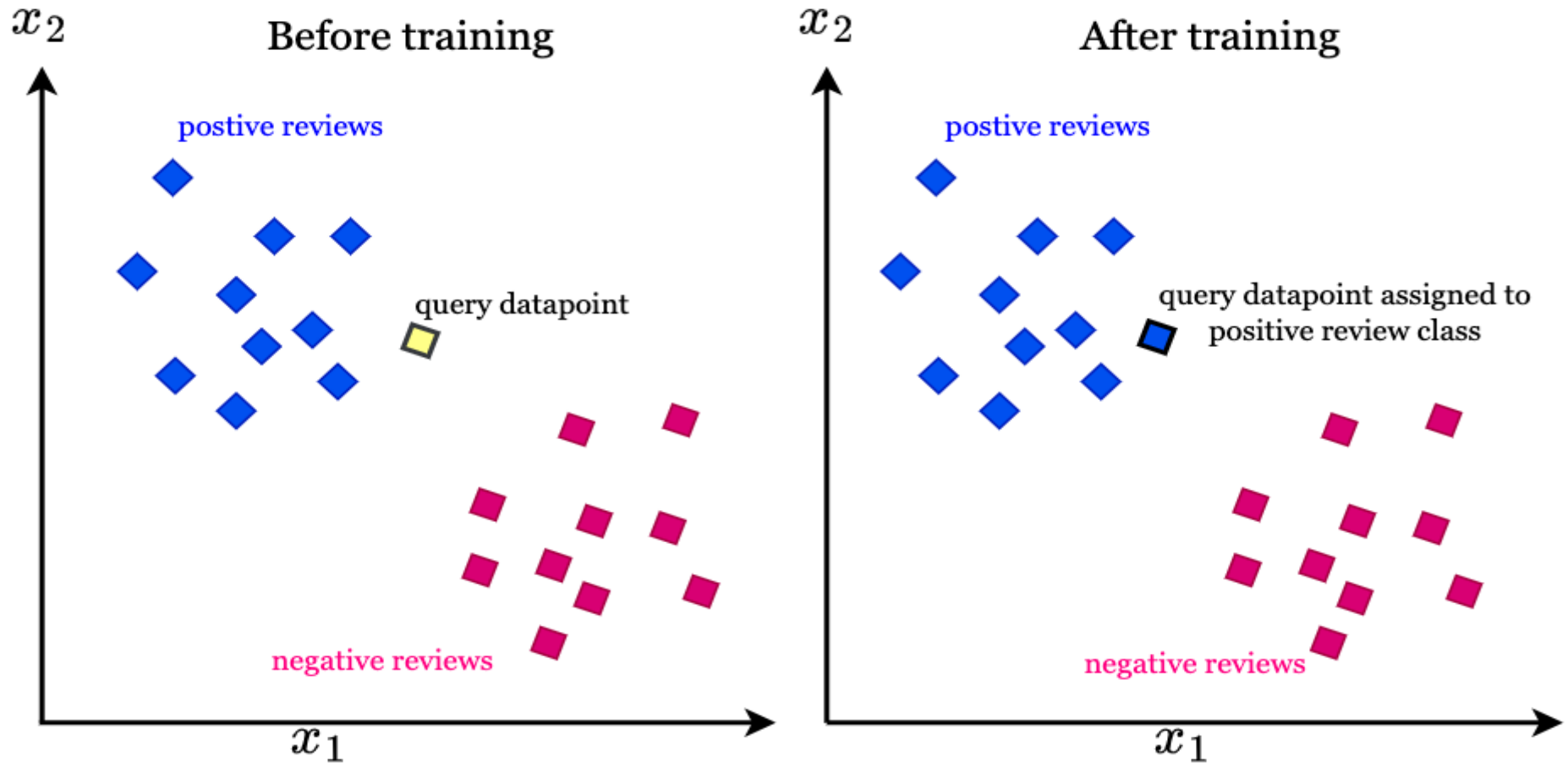
KNN Algorithm (52% accuracy)



KNN Algorithm (k-Nearest Neighbors)



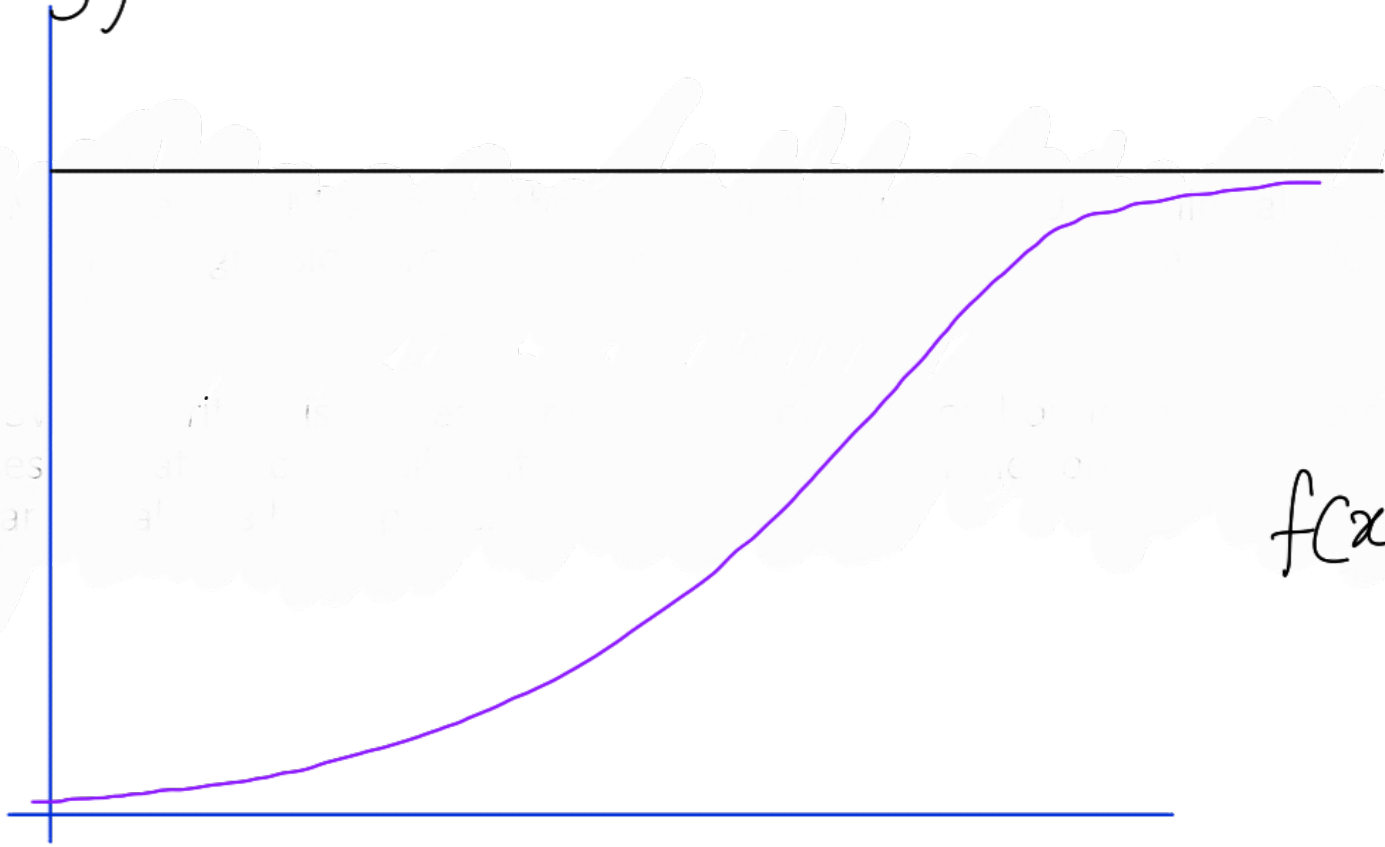
KNN Algorithm (k-Nearest Neighbors)



why use it? → Simple, non parametric, effective for small datasets & non linear relationship

Logistic Regression :- Linar model for binary Classification tasks.

(75% accuracy)



$$f(x) = \frac{1}{1 + e^{-(\beta_1 x + \beta_0)}}$$

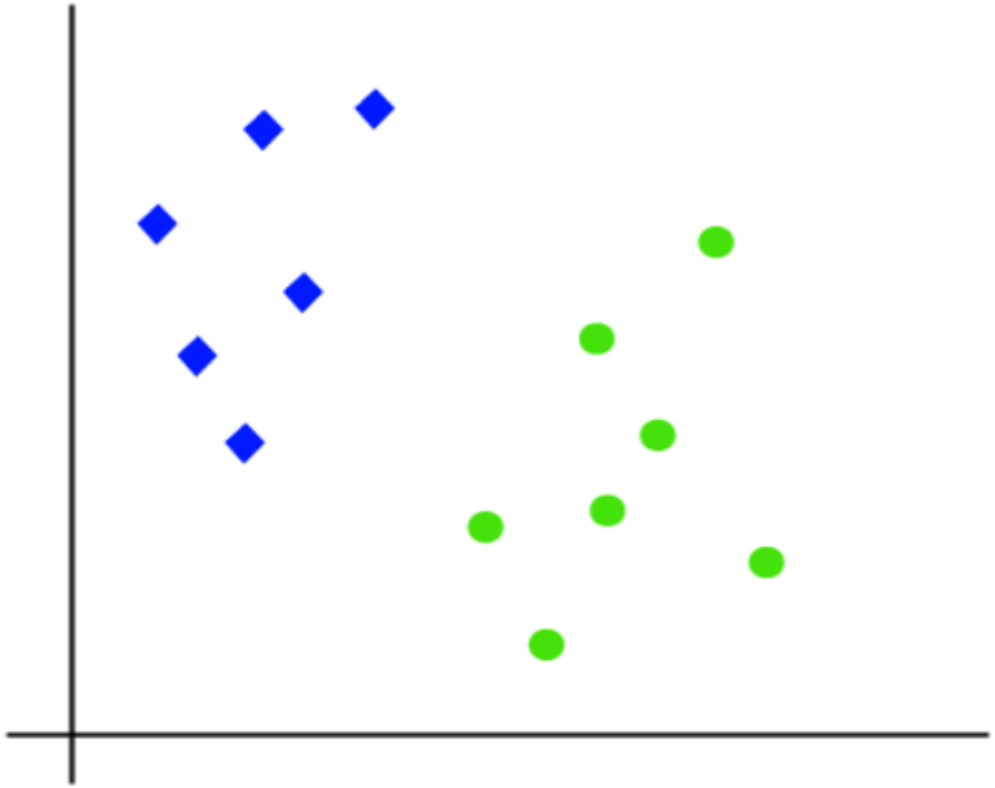
why use it :- Simple, Interpretable, efficient for linear relationship.

Linear SVM

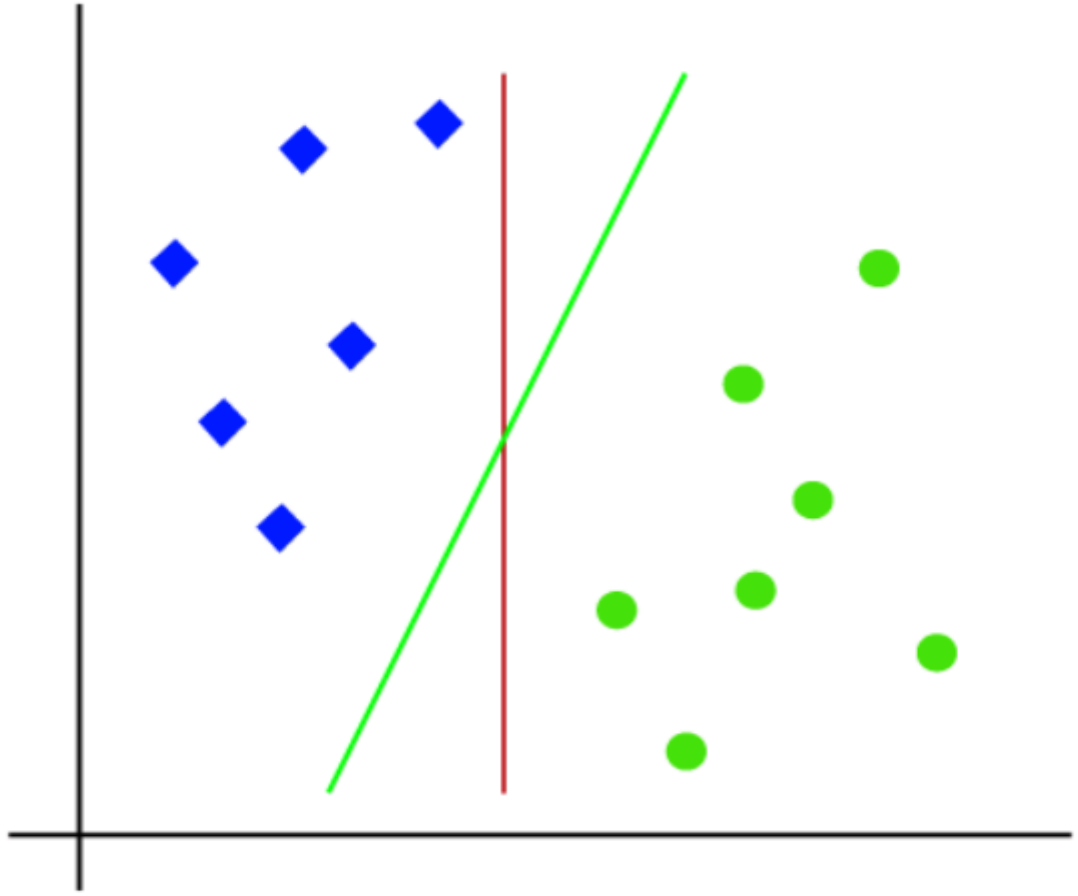
Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

The working of the SVM algorithm can be understood by using an example. Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the pair(x1, x2) of coordinates in either green or blue. Consider the below image:



So as it is 2-d space so by just using a straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes. Consider the below image:

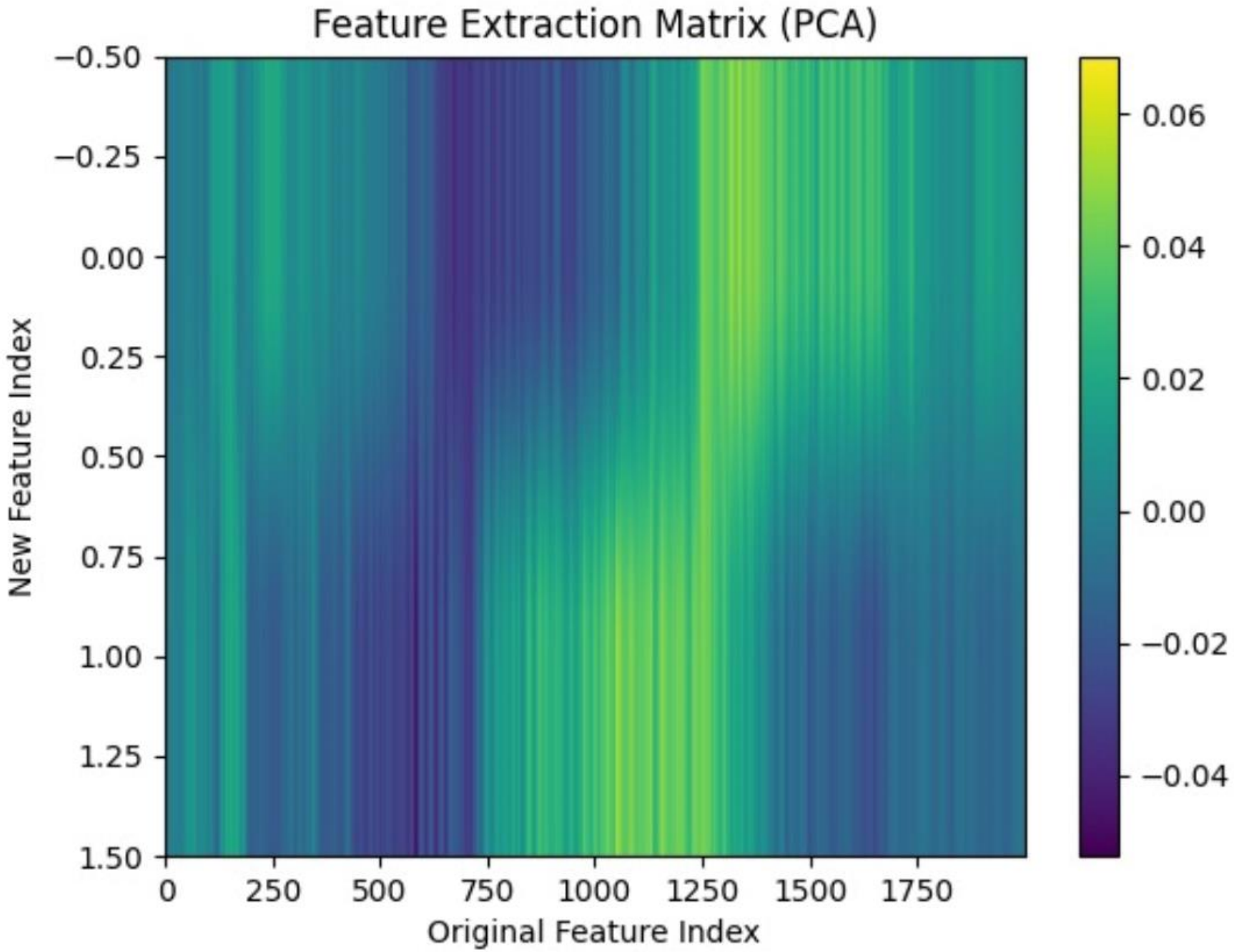


Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a hyperplane. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyperplane is called as margin. And the goal of SVM is to maximize this margin. The hyperplane with maximum margin is called the optimal hyperplane.

Using Linear SVM in our dataset we get the accuracy 72%.

Feature Extraction

Feature extraction is a part of the dimensionality reduction process, in which, an initial set of the raw data is divided and reduced to more manageable groups. So when you want to process it will be easier. The most important characteristic of these large data sets is that they have a large number of variables. These variables require a lot of computing resources to process. So Feature extraction helps to get the best feature from those big data sets by selecting and combining variables into features, thus, effectively reducing the amount of data. These features are easy to process, but still able to describe the actual data set with accuracy and originality.


Feature Extraction Matrix (PCA)
New Feature Index
Original Feature Index
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
1.25
1.50
0
250
500
750
1000
1250
1500
1750
0.06
0.04
0.02
0.00
−0.02
−0.04

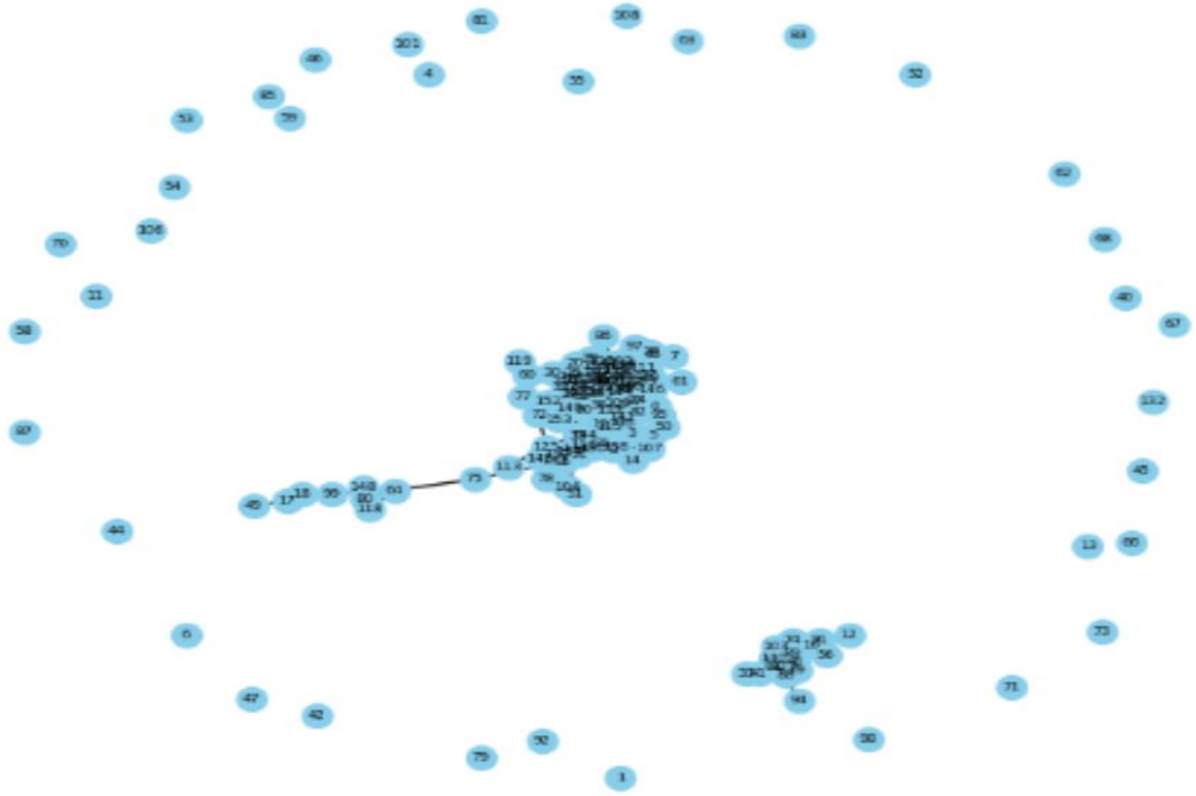
Network Analysis

Network analysis provides powerful tools and techniques for understanding the structure, dynamics, and behavior of complex systems represented as networks. It offers insights into the relationships between entities in diverse fields and helps solve real-world problems by uncovering hidden patterns and structures within networks.

The dataset has 2000 time point and there are two classes, the first class is ’Household Aggregate use of Electricity’ and the second class is ’Aggregate Electricity load of Tumble Dryer and Washing Machine’. The given dataset is divided into two part one is Train and another is Test. Train data has 40 observation and test data has 119 observation. After combining both dataset we have 159 observation with 2000 time points.

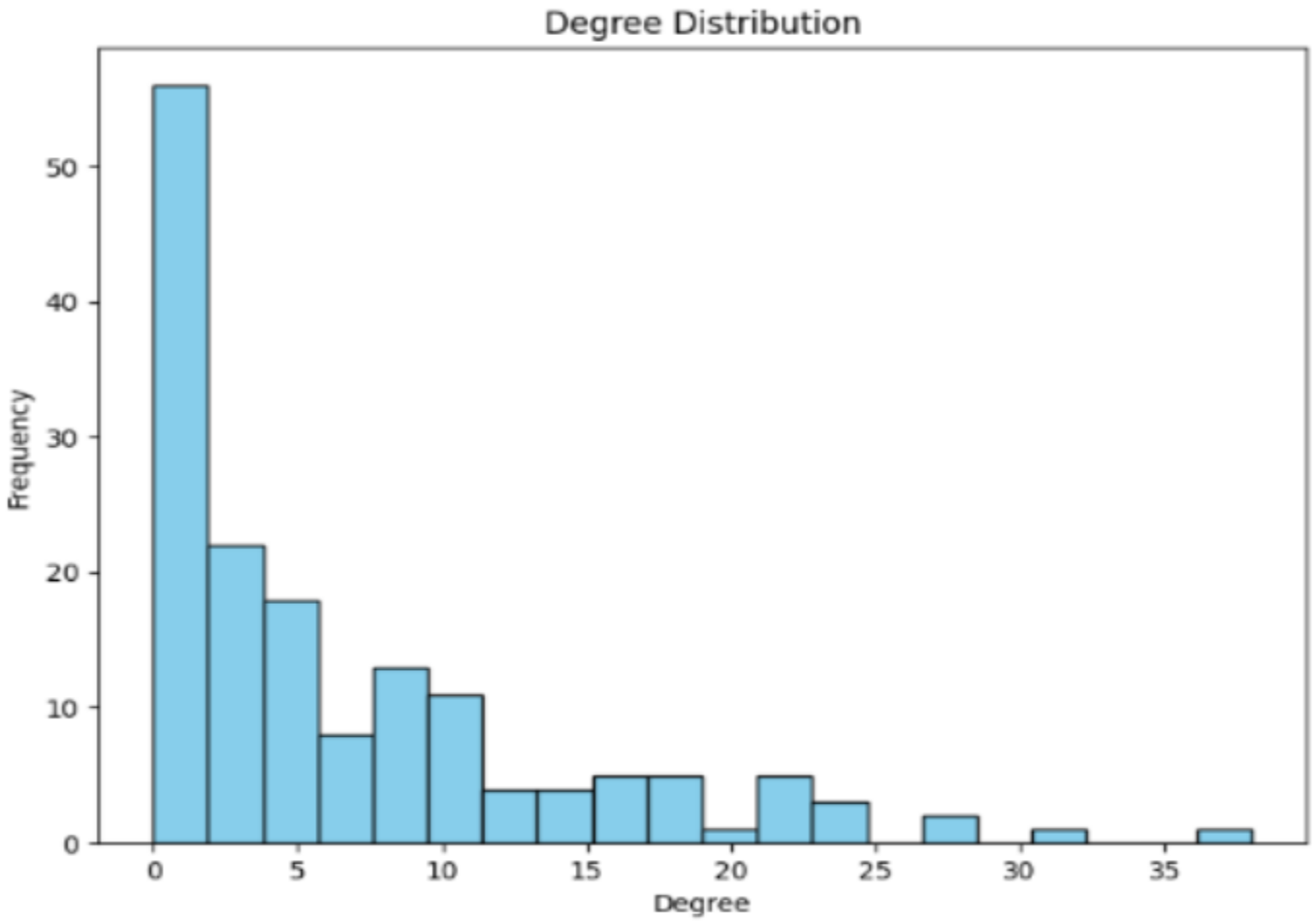
Drawing the Network

To draw the network we use observations as nodes and for similarity measure we use cosine similarity measure, with threshold=0.7 for edges. After drawing the graph looks like.



Analysis the Network

1. Degree Distribution: Visualizing the network we see that there are two clusters. The degree distribution of this network is given below:



2. Centrality measures: Top 10 central nodes:
[136, 147, 150, 22, 27, 36, 137, 122, 129, 139].

3. Clustering Coefficient: Average Clustering Coefficient: 0.386.

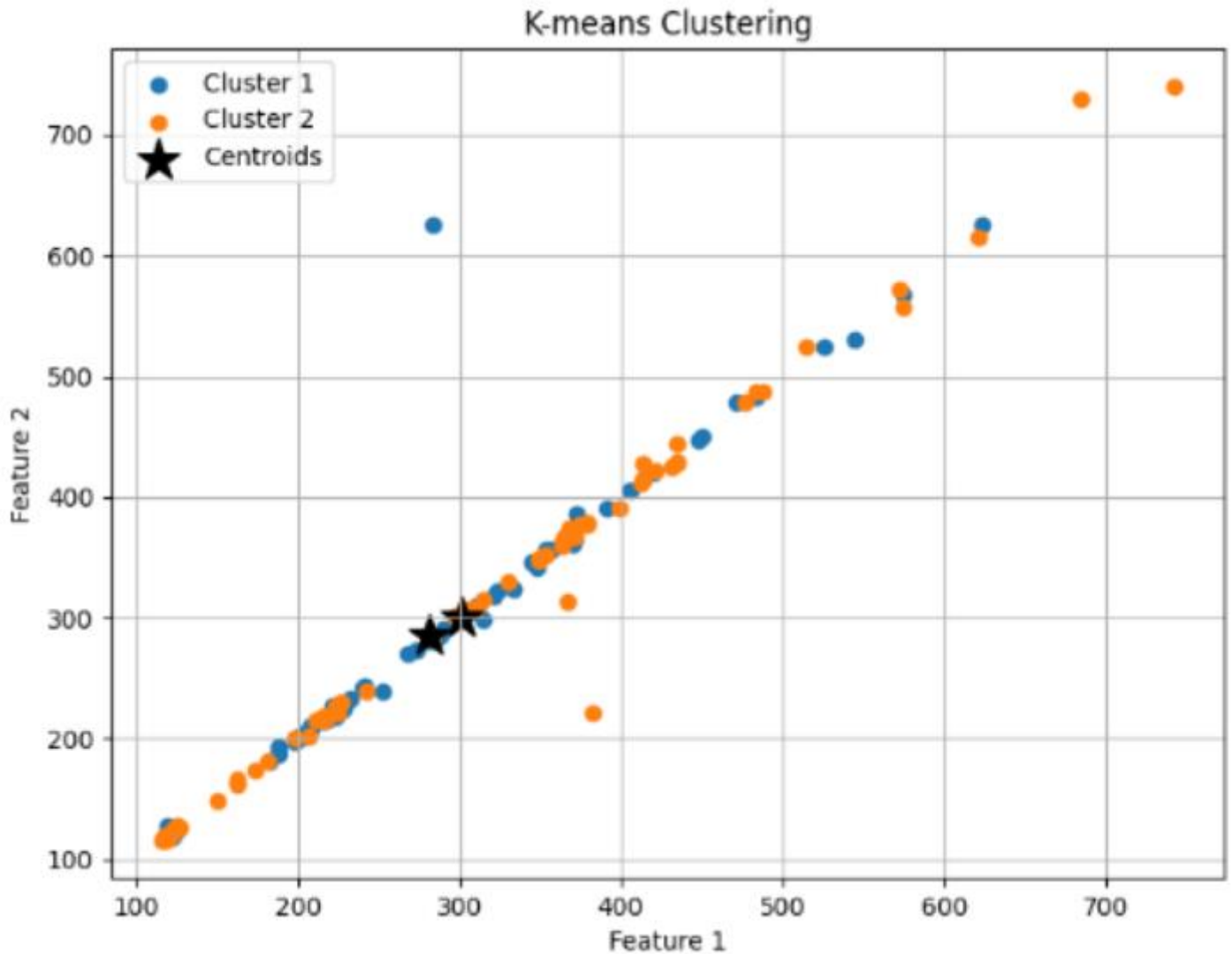
4. Number of Connected Components: Number of Connected Components is 38.

Clustering the Data

1. K-Mean Clustering: K-means clustering is a popular unsupervised machine learning algorithm used for partitioning data into clusters. It's a simple and efficient algorithm that iteratively assigns data points to clusters based on their proximity to cluster centroids and updates the centroids to minimize the within-cluster variance. Here's how k-means clustering works:

- Initialization: Choose the number of clusters (k) and randomly initialize the cluster centroids. These centroids serve as the initial cluster centers.
- Assignment Step: Assign each data point to the nearest cluster centroid based on a distance metric, typically Euclidean distance. Each data point is assigned to the cluster with the closest centroid.
- Update Step: Update the cluster centroids by computing the mean of all data points assigned to each cluster. This moves the centroids to the center of their respective clusters.
- Convergence: Repeat the assignment and update steps iteratively until convergence criteria are met. Convergence criteria can include a maximum number of iterations, minimal change in cluster assignments, or reaching a predefined threshold for within-cluster variance.
- Finalization: Once convergence is achieved, the algorithm outputs the final cluster assignments and centroids.

Here we take k=2 as there are two classes. After clustering we plot the graph for visualization.


K-means Clustering
Cluster 1
Cluster 2
Centroids
Feature 2
Feature 1
100
200
300
400
500
600
700

Accuracy= 0.45

The results shows that accuracy is very low. Therefore there is mismatch between the clusters and actual class labels of the data.