MA4207 Final Project

House Twenty

Submitted by

Abhijit Kundu (20MS177)
Abhisek Sarkar (20MS091)
Narayan Biswas (23RS060)

IISER KOLKATA

Github Repository Link

Contents

1 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 (data available from Strathclyde University, see below). - Replacement gas meters, to measure household mains gas consumption. - Hobo pendant or Hobo U12 sensors to measure room air temperature, relative humidity and light level. - iButton temperature sensors to measure radiator surface temperature. - CurrentCost individual appliance monitors, to measure plug electrical power loads (data available from Strathclyde University, see below). - RWE Smart Home devices including programmable thermostatic radiator valves, interior and exterior motion detectors, door and window opening sensors and smoke alarms. - British Gas Hive programmable thermostats. In addition climate data was collected at the Loughborough University campus weather station.
— TIMELINE September 2013 to February 2014: Building surveys were carried out and monitoring sensors were placed in the buildings at or shortly after this time. June 2014 and October 2014: Smart Home devices were installed in the buildings.
April 2015: Data collection finished.

—

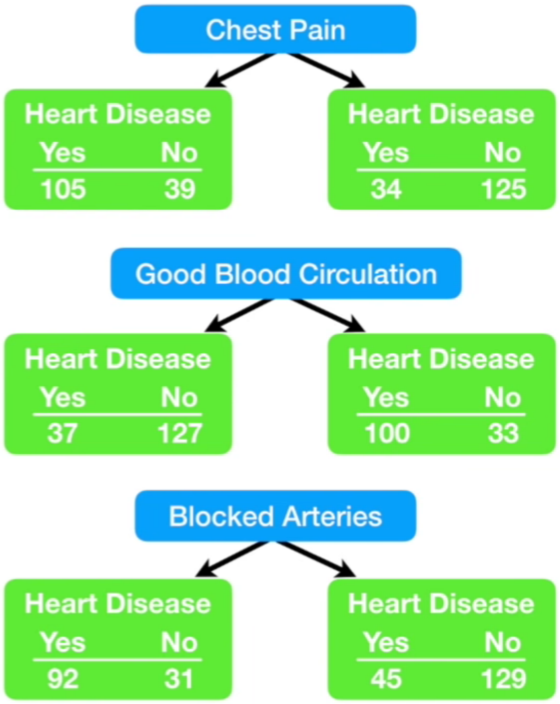
DATA STATISTICS
Number of homes: 20
Number of spaces (rooms): 389
Number of radiators: 252
Number of showers: 34
Number of appliances: 618
Number of light bulbs: 672
Number of fixed heaters: 19
Number of surfaces: 2237
Number of openings: 970
Number of sensors: 1,567
Number of variables recorded by sensors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

2 Pattern Classification Technique

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



So we have the following data:

(a) 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 $= 1 - (\frac{34}{34+125})^2 - (\frac{125}{34+125})^2 = 0.336$

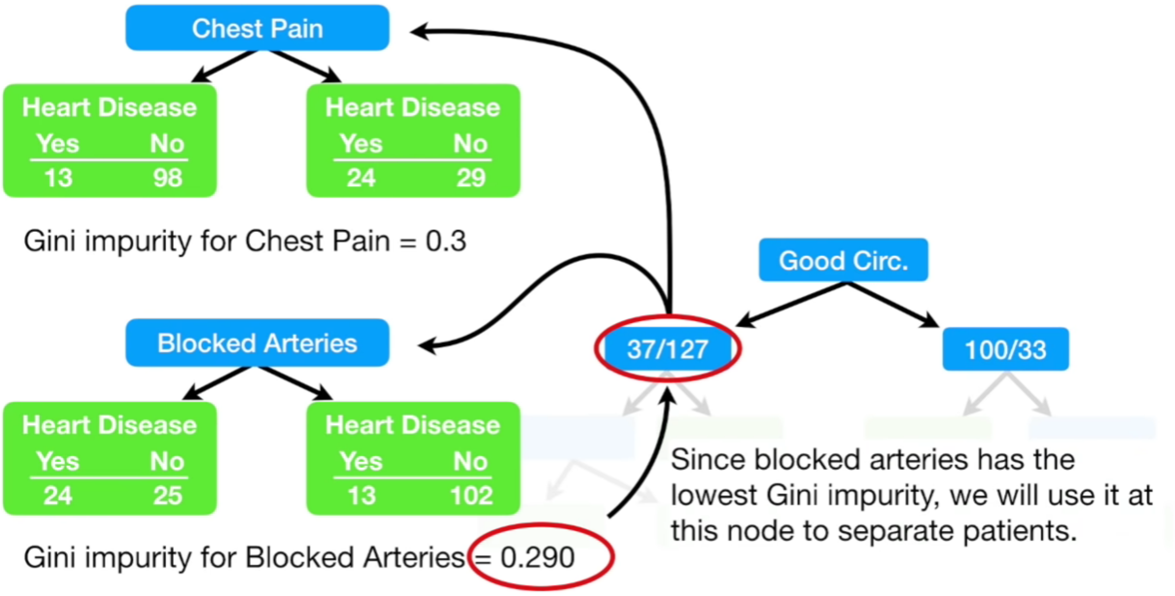
Total Gini Impurity for Chest Pain $= \frac{144}{144+159} \times 0.395 + \frac{159}{144+159} \times 0.336 = 0.364$

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.

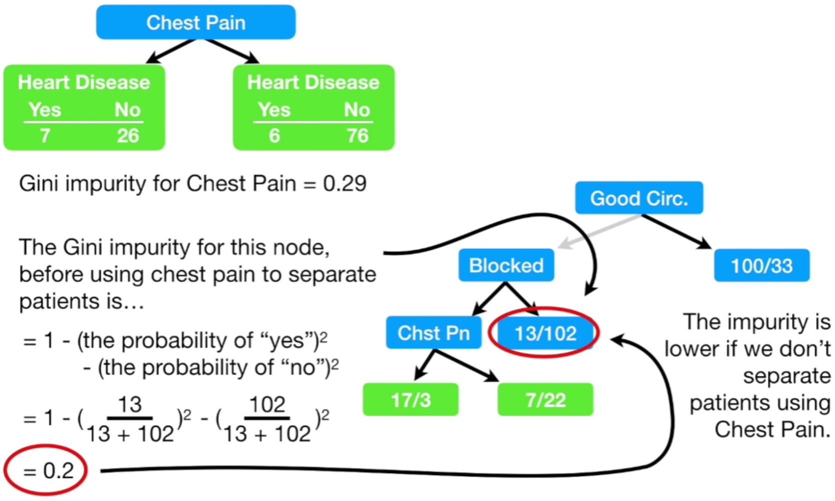

Chest Pain
Heart Disease
Yes No
13 98
Heart Disease
Yes No
24 29
Gini impurity for Chest Pain = 0.3
Good Circ.
37/127
100/33
Blocked Arteries
Heart Disease
Yes No
24 25
Heart Disease
Yes No
13 102
Gini impurity for Blocked Arteries = 0.290
Since blocked arteries has the lowest Gini impurity, we will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

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

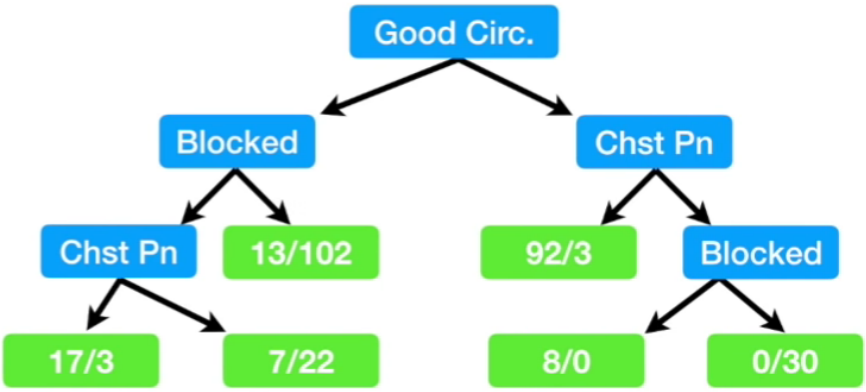
So firstly we separate using Good Blood Ciculation if yes then we go with Blocked Arteries. So all we have left is Chest Pain.

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.


Chest Pain
Heart Disease Yes No 7 26
Heart Disease Yes No 6 76
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 - (13 / 13 + 102)2 - (102 / 13 + 102)2
= 0.2
Good Circ.
Blocked
100/33
Chst Pn
13/102
17/3
7/22
The impurity is lower if we don't separate patients using Chest Pain.

In the similar way we finally get this decision tree.



So we use the same method for classification of our dataset.
using this method we get accuracy 57 %, which is bit less so we have to use some other method also.

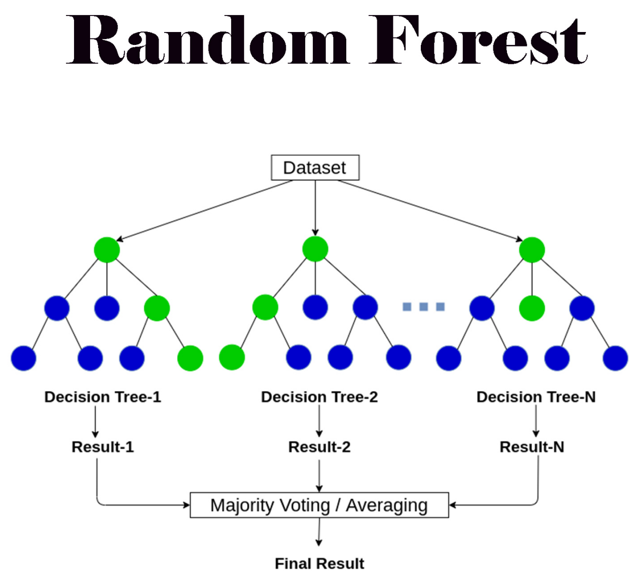
2.2 Random Forest:

2.2.1 Definition:

Random Forest is a machine learning model that uses an ensemble of decision trees to make its predictions.

2.2.2 Reasons to use Random Forest:

- **Reduces overfitting:** It can help reduce overfitting, which occurs when a model starts to memorize the data instead of trying to generalize for making predictions on future data. It helps you get around the limitations of your data, which might not be fully representative of all golfers or all the best features in your model.
- **Reduces bias:** It can also help reduce bias, which can occur when there is a certain degree of error introduced into the model, bias occurs when you're not evenly splitting your instant space during training. So instead of seeing all of the data points, you might see only half because of how you set your model up.



2.2.3 How Random Forest works?

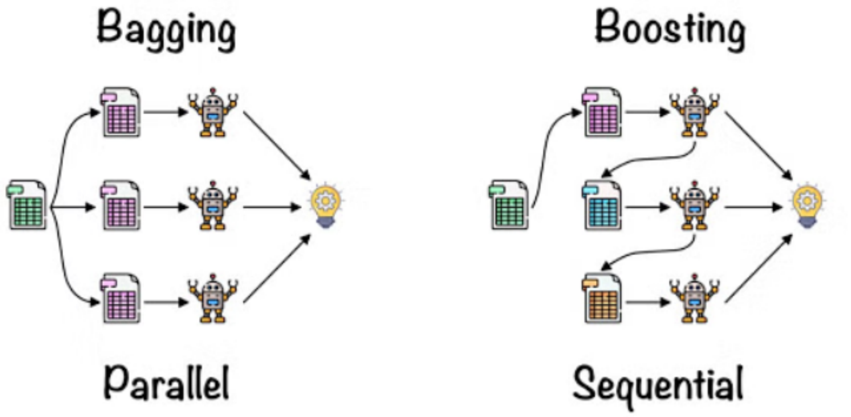
The Random Forest Algorithm's operation is explained in the phases that follow:

- Step 1: From a given training set or data, choose random samples.
- Step 2: For each training set of data, this algorithm will build a decision tree.

- Step 3: The decision tree will be averaged to determine the winner.
- Step 4: Choose the predicted result that received the most votes to be the final outcome.

We refer to this amalgamation of various models as an ensemble. Ensemble employs two techniques:

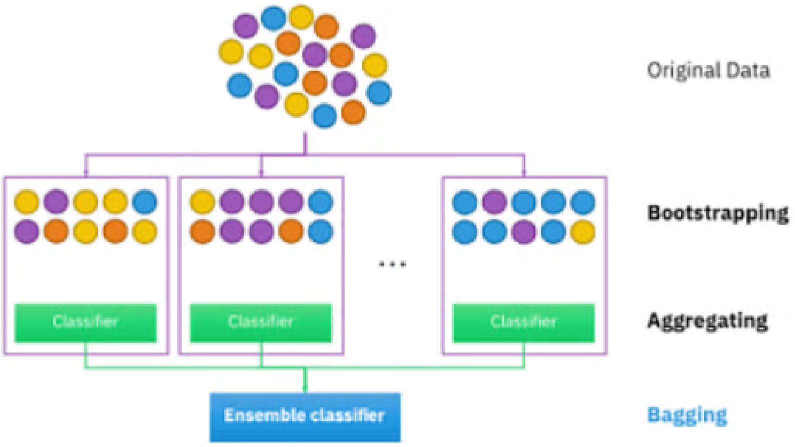
- **Bagging:** Bagging is the process of separating out a distinct training subset using replacement from sample training data. The result is determined by a majority vote.
- **Boosting:** Boosting is the process of combining weak learners with strong learners by building successive models until the greatest accuracy model is achieved. For instance, ADA and XG boosts.



It is clear that Random Forest employs the Bagging code based on the previously stated idea. Let's now examine this idea in more detail. Another name for bagging is the Bootstrap Aggregation that Random Forest uses. Starting with any initial random data, the process starts. It is then arranged and divided into what are referred to as Bootstrap Samples. We call this technique "bootstrapping."Additionally, each model is trained independently, producing aggregation—a term for the many outcomes. The output that is produced in the last stage is based on majority voting after all of the results have been aggregated. Known as Bagging, this stage makes use of an Ensemble Classifier.

2.2.4 Essential Features of Random Forest:

- **Random:** Every tree has a distinct quality, range, and characteristic in relation to other trees. Trees differ from one another.
- **Immune to the curse of dimensionality:** The curse of dimensionality does not apply to trees because they are conceptual concepts and do not need characteristics to be taken into account. As a result, there is less feature space.
- **Parallelization:** Since each tree is built independently from distinct data and features, we can produce random forests by utilizing the entire CPU.



- **Train-Test split:** Since the decision tree in a Random Forest never sees 30% of the input, we don't need to separate the data for training and testing.
- **Stability:** The outcome is determined by bagging, which uses the average or majority vote to determine the outcome.

2.2.5 Why use Random Forest algorithm?

The Random Forest Algorithm has several advantages, but reducing the likelihood of overfitting and the amount of training time needed is one of its primary benefits. It also provides a high degree of precision. By approximating missing data, the Random Forest algorithm generates extremely accurate predictions while operating quickly in big databases.

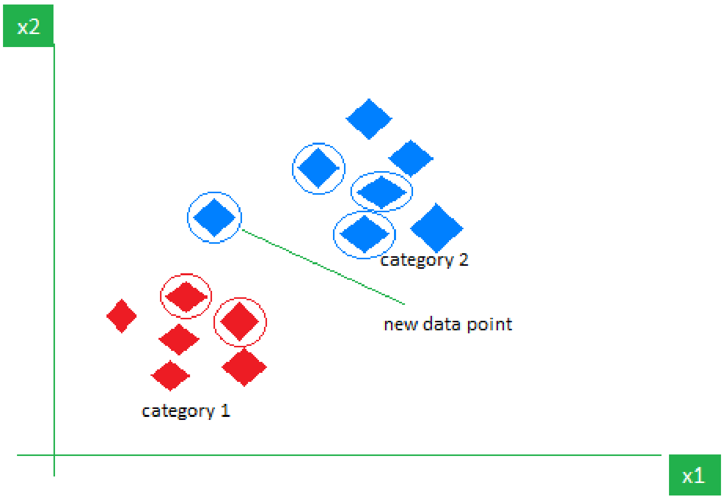
2.3 KNN:

The K-Nearest Neighbors (KNN) algorithm is a supervised machine learning approach for solving classification and regression issues. This algorithm was created in 1951 by Evelyn Fix and Joseph Hodges and was later enhanced by Thomas Cover.

2.3.1 How KNN works?

- **Data Preparation:** The data is prepared for the algorithm. This may involve scaling the data, handling missing values, and encoding categorical variables.
- **Distance Calculation:** The distance is calculated between the new data point and all the data points in the training set. There are various distance metrics used in KNN, such as Euclidean distance, Manhattan distance, and Hamming distance.
- **K Selection :** A value for K is chosen. K represents the number of nearest neighbors that will be used to classify the new data point. Finding Nearest Neighbors: The algorithm identifies the K nearest neighbors to the new data point based on the distance measure.

- **Majority Vote:** The most frequent class label among the K nearest neighbors is assigned as the class label of the new data point.



2.3.2 Advantages:

- **Easy to implement:** Because of the algorithm's low complexity, it is simple to implement.

- **Easily Adapts:** The KNN algorithm operates by storing all of the data in memory storage. As a result, whenever a new example or data point is supplied, the algorithm automatically modifies itself to take into account the new example and contributes to the future forecasts.

- **Few Hyperparameters:** The value of k and the distance metric we want to select from our evaluation metric are the only parameters needed for the training of a KNN algorithm.

2.3.3 Disadvantages:

- **Does not scale:** According to what we've heard, the KNN method is likewise categorized as a lazy algorithm. This term's primary significance lies in the fact that it requires a significant amount of processing power and data storage. Because of this, this algorithm uses a lot of time and resources.

- **Curse of Dimensionality:** The peaking phenomena states that the KNN method is impacted by the curse of dimensionality, which means that when dimensionality is too large, the algorithm has difficulty correctly categorizing the data points.

- **Prone to Overfitting:** The algorithm is vulnerable to overfitting issues as a result of the dimensionality curse. Therefore, to address this issue, feature selection and dimensionality reduction approaches are typically used.

2.4 Logistic Regression:

Logistic regression is used for binary classification where we use sigmoid function, that takes input as independent variables and produces a probability value between 0 and 1.

2.4.1 Sigmoid function:

The sigmoid function is a mathematical function used to map the predicted values to probabilities. It maps any real value into another value within a range of 0 and 1. The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function. In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

2.4.2 How Logistic Regression works?

The logistic regression model uses a sigmoid function, which converts any real-valued collection of input independent variables into a value between 0 and 1, to convert the continuous value output of the linear regression function into categorical value output. The logistic function is the name given to this role.

Let the independent input features be:

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1m} \\ x_{21} & \cdots & x_{2m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{bmatrix}$$

where Y, the dependent variable, only has two possible values: 0 and 1.

$$Y = \begin{cases} 0, & \text{if Class 1} \\ 1, & \text{if Class 2} \end{cases}$$

Next, give the input variables X a multi-linear function application.

$$z = \sum_{i=1}^{n} w_i x_i + b$$

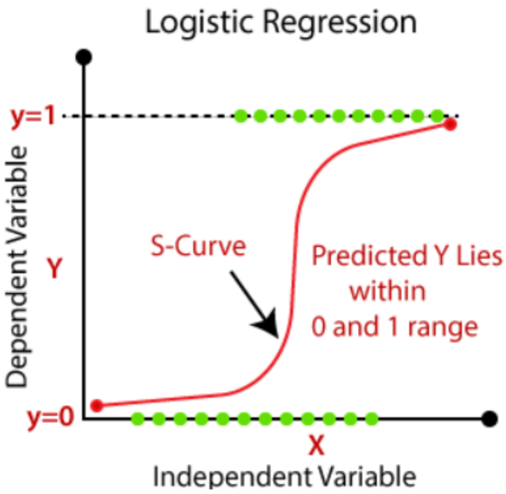
Here x_i is the ith observation of X, $wi = [w_1, w_2, w_3, \cdots, w_m]$ is the weights or Coefficient, and b is the bias term also known as intercept. simply this can be represented as the dot product of weight and bias.

$$z = w \cdot X + b$$

$$p(X; b, w) = \frac{e^{w \cdot X + b}}{1 + e^{w \cdot X + b}} = \frac{1}{1 + e^{-w \cdot X + b}}$$

where:

- $p(X; b, w)$ is the probability of a specific outcome
- X is the independent variable
- w is the weight vector
- b is the bias term



2.4.3 Types of Logistic Regression

Three forms of logistic regression can be distinguished based on the categories:

- **Binomial:** The dependent variables in a binomial logistic regression can only be of two types: either 0 or 1, Pass or Fail, etc.
- **Multinomial:** In multinomial logistic regression, the dependent variable, such as "cat," "dogs," or "sheep," may be one or more of three potential unordered kinds.
- **Ordinal:** Three or more ordered dependent variable types, such as "low," "medium," or "high," are conceivable in ordinal logistic regression.

2.4.4 Key Points:

- Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value.
- It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.
- In Logistic regression, instead of fitting a regression line, we fit an “S” shaped logistic function, which predicts two maximum values (0 or 1).

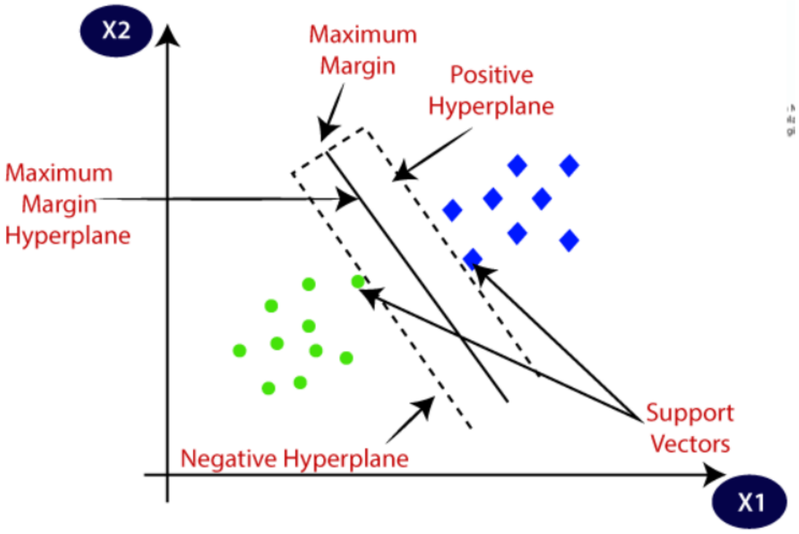
So using the above method in our dataset, we get that logistic regression is best for our dataset and we get 75 % accuracy.

2.5 linear SVM:

Strong machine learning algorithms like Support Vector Machine (SVM) are utilized for tasks including regression, outlier identification, and linear or nonlinear classification. Text classification, picture classification, handwriting recognition, spam detection, face detection, gene expression analysis, and anomaly detection are just a few of the many applications for SVMs. Because SVMs can handle high-dimensional data and nonlinear relationships, they are versatile and effective in a wide range of applications.

2.5.1 Support Vector Machine:

A supervised machine learning approach called Support Vector Machine (SVM) is used for regression as well as classification. Even yet, classification problems are the most appropriate use for regression problems. The SVM algorithm's primary goal is to locate the best hyperplane in an N-dimensional space that may be used to divide data points into various feature space classes. The hyperplane attempts to maintain the largest possible buffer between the nearest points of various classes. The number of features determines the hyperplane's dimension. The hyperplane is essentially a line if there are just two input features. The hyperplane transforms into a 2-D plane if there are three input features. If there are more than three features, it gets hard to imagine.



In the diagram where a decision boundary or hyperplane is used to classify two distinct categories:

So using linear SVM in our dataset we get 72 % accuracy.

3 Network Analysis

3.1 Introduction

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.

3.2 Nodes and Edges:

In a network, nodes represent entities, such as individuals, computers, proteins, or cities, while edges represent relationships or connections between pairs of nodes.

3.3 Types of Networks:

1. **Undirected Networks:** In undirected networks, edges have no directionality, meaning that the relationship between nodes is symmetric.
2. **Directed Networks:** In directed networks, edges have directionality, indicating a one-way relationship from one node to another.
3. **Weighted Networks:** In weighted networks, edges have weights or values associated with them, representing the strength or intensity of the relationship between nodes.
4. **Signed Networks:** In signed networks, edges can be positive or negative, representing positive or negative relationships between nodes.

3.4 Network Properties:

1. **Degree:** The degree of a node is the number of edges connected to it. In directed networks, nodes have both in-degree (number of incoming edges) and out-degree (number of outgoing edges).
2. **Centrality:** Centrality measures identify the most important nodes in a network based on different criteria, such as degree centrality, betweenness centrality, and closeness centrality.
3. **Clustering Coefficient:** The clustering coefficient measures the extent to which nodes in a network tend to cluster together, indicating the presence of communities or clusters.
4. **Path Length:** The shortest path length between two nodes is the minimum number of edges that must be traversed to go from one node to another.

3.5 Dataset

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.

3.6 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

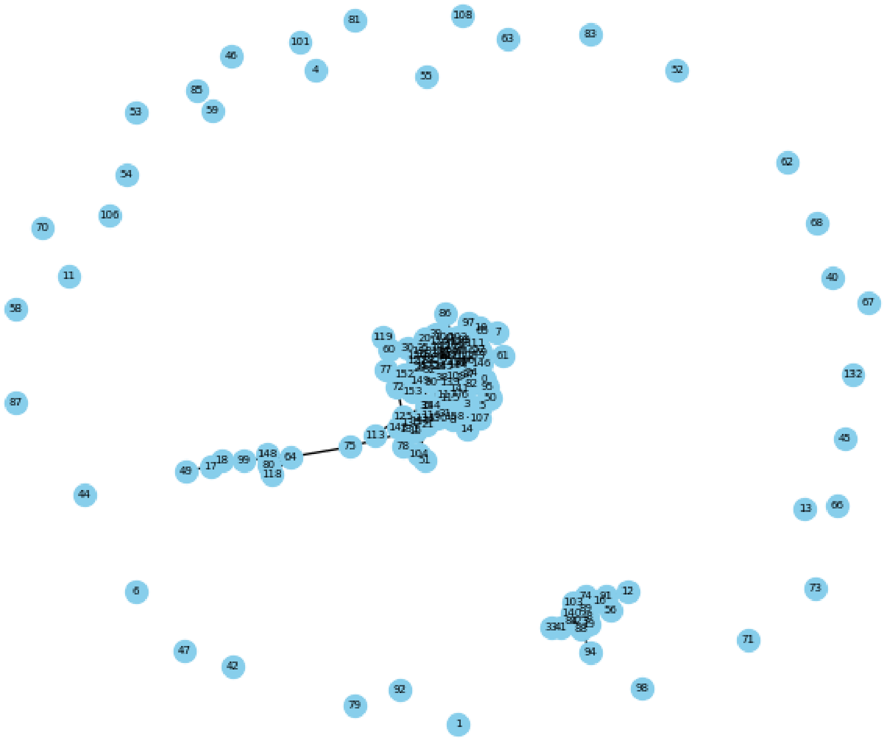


Figure 1: Network

3.7 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:

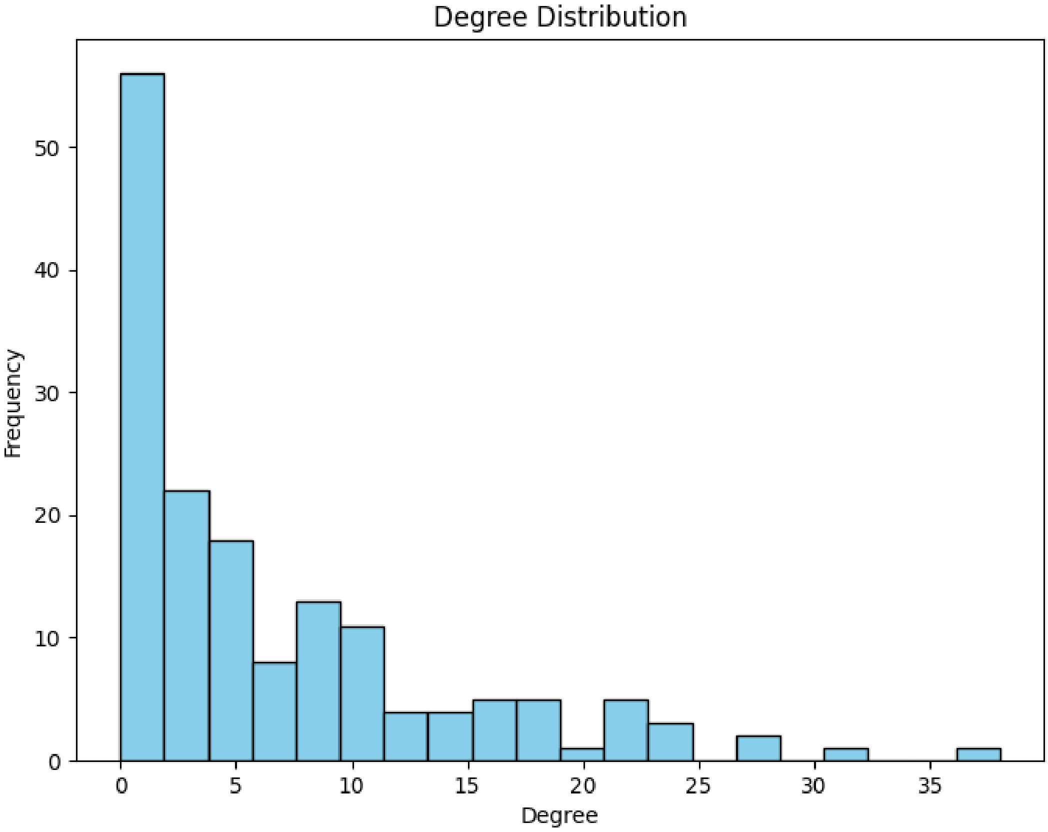


Figure 2: Degree Distribution

this shows that degree of any node is lesser then 40, that is there is no nodes which is connected more then 40 nodes.

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.

3.8 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.

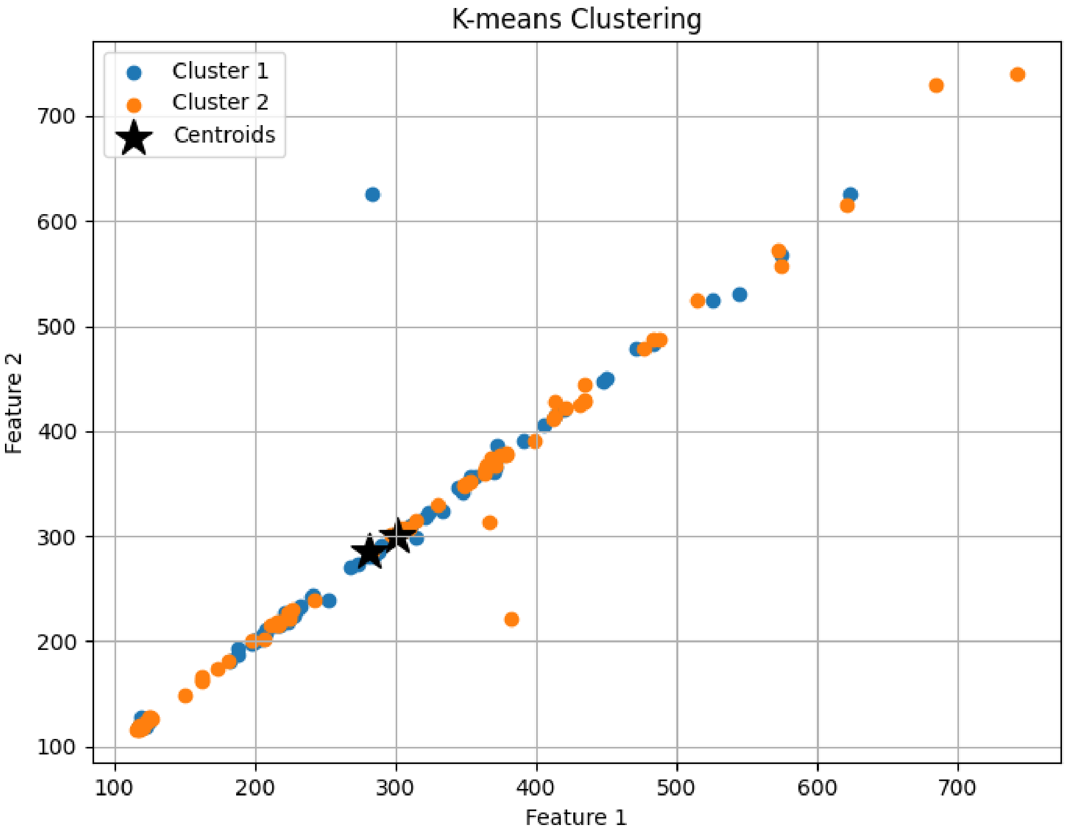


Figure 3: K-Mean Clustering

2. **Evaluation:**

   - Adjusted Rand Index: 0.123.
   - Adjusted Mutual Information: 0.094.
   - Silhouette Score: 0.087.
   - Accuracy: 0.195.

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