

# Healthy Diet Recommendation and IPL Data Analysis

A Project as a Course requirement for  
**Master of Sciences in Data Science and Computing**

**S. Vinod Kumar**

19234



**SRI SATHYA SAI INSTITUTE OF HIGHER LEARNING**  
(Deemed to be University)

Department of Mathematics and Computer Science  
Muddenahalli Campus

April 2021



# SRI SATHYA SAI INSTITUTE OF HIGHER LEARNING

(Deemed to be University)

Dept. of **Mathematics & Computer Science**  
Muddenahalli Campus

## CERTIFICATE

This is to certify that this Project titled **Healthy Diet Recommendation and IPL Data Analysis** submitted by **S.Vinod Kumar, 19234**, Department Of Mathematics and Computer Science, Muddenahalli Campus is a bonafide record of the original work done under my/our supervision as a Course requirement for the Degree of MSc Data Science and Computing.

.....  
Shri V. Bhaskaran  
Project Supervisor

Countersigned by

.....  
Dr. (Mrs.) Rita Gupta  
Head of the Department

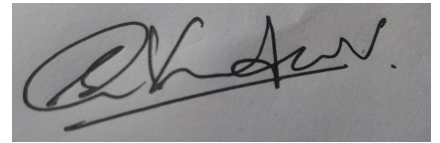
Place:

Date:

Sri Sathya Sai Grama, Dist. Chickballapur - 562 101, Karnataka, India  
Tel: +91 96637 65789 | hoddmacs@sssihl.edu.in | www.sssihl.edu.in

## DECLARATION

The Project titled **Healthy Diet Recommendation and IPL Data Analysis** was carried out by me under the supervision of **Shri V. Bhaskaran**, Department Of Mathematics and Computer Science, Muddenahalli Campus as a Course requirement for the Degree of MSc Data Science and Computing and has not formed the basis for the award of any degree, diploma or any other such title by this or any other University.



.....

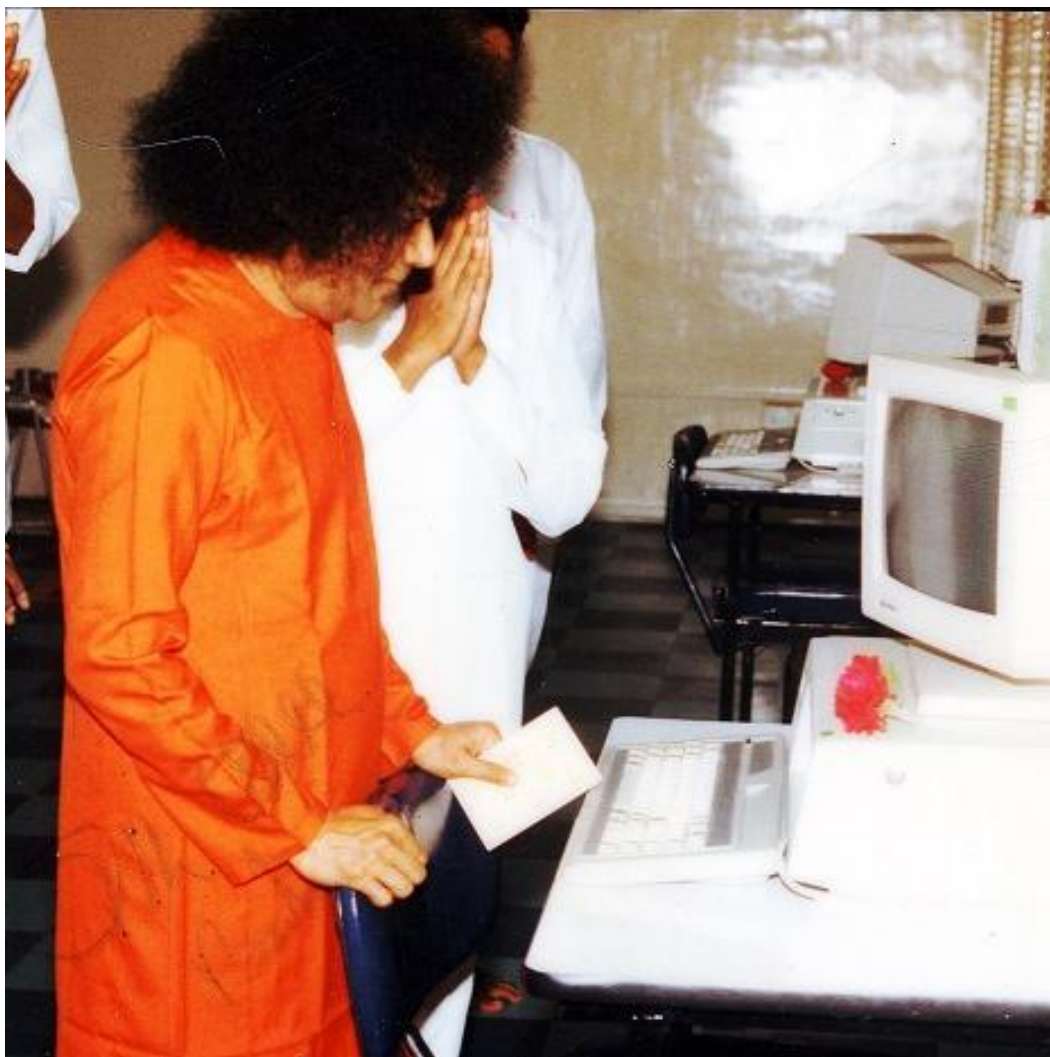
Place : Muddenahalli

S. Vinod Kumar

19234

Date : 20-04-2021

MSc Data Science and Computing  
Muddenahalli Campus



## ACKNOWLEDGEMENT

I am overwhelmed in all gratefulness and humbleness to acknowledge my deep and sincere gratitude to all those who helped me to shape this project.

It would have been impossible to achieve this project over a year without the backing of the elders and their guidance. I would be failing in my duty if I don't acknowledge their support throughout my continuous struggle, rise and fall to complete this project.

I bow down in unfathomable gratitude to **Bhagawan Sri Sathya Sai Baba**, for if not His will, I would not have been part of this project itself. I am also grateful to Him for showing the path whenever I was stuck in the crossroads of making any decision.

I thank my professor guide **Shri V. Bhaskaran Sir**, whose strong conviction, persistent motivation, and endless enthusiasm helped us to have that yearning within.

I also thank my external guides **Shri M Chandrasekhar, Vinay Babu sir, Sandeep sir** and Dell team for their constant support, anticipation, and belief which helped me at every step of this project.

I would also like to thank the University Administration for providing me with the resources required in the course of this project. I thank all my classmates for their cheerfulness and for providing perceptions and vigor for this project.

Last but not least, I would like to thank my family for their unbroken assurance of me.

## TABLE OF CONTENTS

1. ABSTRACT.	8
2. INTRODUCTION.	9
3. TOOLS AND TECHNOLOGIES USED.	10
4. METHODS AND ALGORITHMS USED.	11
a. DECISION TREE.	11
b. RANDOM FOREST.	11
c. LOGISTIC REGRESSION.	12
d. SUPPORT VECTOR MACHINES (SVM).	13
e. CONFUSION MATRIX.	13
f. CLASSIFICATION REPORT.	14
5. DATA COLLECTION.	15
6. PRE-PROCESSING.	17
7. FEATURE ENGINEERING FOR PLAYERS.	18
8. MODELING.	21
a. MODEL 1	21
i. CODE	22
ii. CLASSIFICATION REPORT.	23
iii. DESCRIPTION.	24
iv. CODE FOR COMBINING CLASSES.	25
b. MODEL 2	26
i. CLASSIFICATION REPORT.	26
ii. DESCRIPTION.	28
iii. FEATURE ENGINEERING (last 6 balls).	28
c. MODEL 3	29
i. CLASSIFICATION REPORT	29
ii. DESCRIPTION.	31
9. CONCLUSION.	31
10. FUTURE SCOPE.	31
11. HEALTHY DIET RECOMMENDATION.	32

## TABLE OF FIGURES

1. DECISION TREE.	11
2. RANDOM FOREST.	12
3. LOGISTIC REGRESSION.	12
4. SUPPORT VECTOR MACHINES.	13
5. CONFUSION MATRIX.	13
6. CLASSIFICATION REPORT.	14
7. BATSMAN DATA.	20
8. MODEL 1 - CONFUSION MATRIX OF RANDOM FOREST.	25
9. MODEL 2 - CONFUSION MATRIX OF RANDOM FOREST.	27
10. MODEL 3 - CONFUSION MATRIX OF GAUSSIAN NAIVE BAYES.	30

## **ABSTRACT**

IPL (Indian Premier League) is one of the most famous forms of Domestic cricket. Players from all over the world come to participate in IPL. It also provides a chance for young cricketers to play with veterans from different countries. This form of playing creates a bond between players and also boosts the spirit of the sport.

The aim of this project is that given a batsman and a bowler how much can the batsman score in that particular ball. The possible outcomes are [ 1, 2, 3, 4, 6] runs and a wicket. There are many algorithms that try to predict the win percentage of a team, runs scored by a batsman, etc. In this case of Multiclass classification, Machine Learning algorithms like Decision Tree, Random Forest, SVM( Support Vector Machine) are used in order to achieve the goal.



## INTRODUCTION

Cricket is one of the most common and unique sports which is played across the world in three different formats; they are Test matches, One-Day Internationals, and 20-20. 20-20 Cricket is the shortest format of the game where there will be 20 overs for each innings. The first and last five overs in 20-20 are powerplay over and Depth overs respectively. The batting powerplay and depth overs play a significant role. If both teams score the same number of runs then there will be a super over, where the match will be conducted for a single over with two wickets in hand.

Game's data can be found on the internet, which could be used by coaches and team management which would help players in improving their skills. Many insights have been brought after applying many statistical techniques on the past data of players in each sport. However, there has been quite some research done in the area of sports and even many machine learning algorithms have been deployed for predicting the match outcome with high accuracy.

Machine learning is used to solve many real-world problems in a variety of fields. It's applied to soccer/football, baseball, cricket, Tennis, Badminton to improve the competition between professional players. For example, a neural network model that is trained on a baseball dataset can be used on a cricket dataset by changing the hyperparameters.

### **IPL AND MACHINE LEARNING:**

Indian Premier League(IPL) is the World's richest professional 20-20 cricket league. There have been 13 seasons completed in the IPL tournament. A large amount of data has been collected over the last 13 seasons and currently, there are models to predict a team's win and individual batsman score and prediction of a wicket. This project aims to predict a ball's outcome which leads to multiclass classification and algorithms like SVM, Random forest, Decision tree, etc can be used.

## **TOOLS AND TECHNOLOGIES USED**

### **PROGRAMMING LANGUAGE :**

- Python 3.8

### **PROGRAMMING ENVIRONMENT :**

- Pycharm

### **PYTHON PACKAGES :**

- Pandas
- Numpy
- Matplotlib
- Seaborn
- SciKit learn

### **SYSTEM CONFIGURATION:**

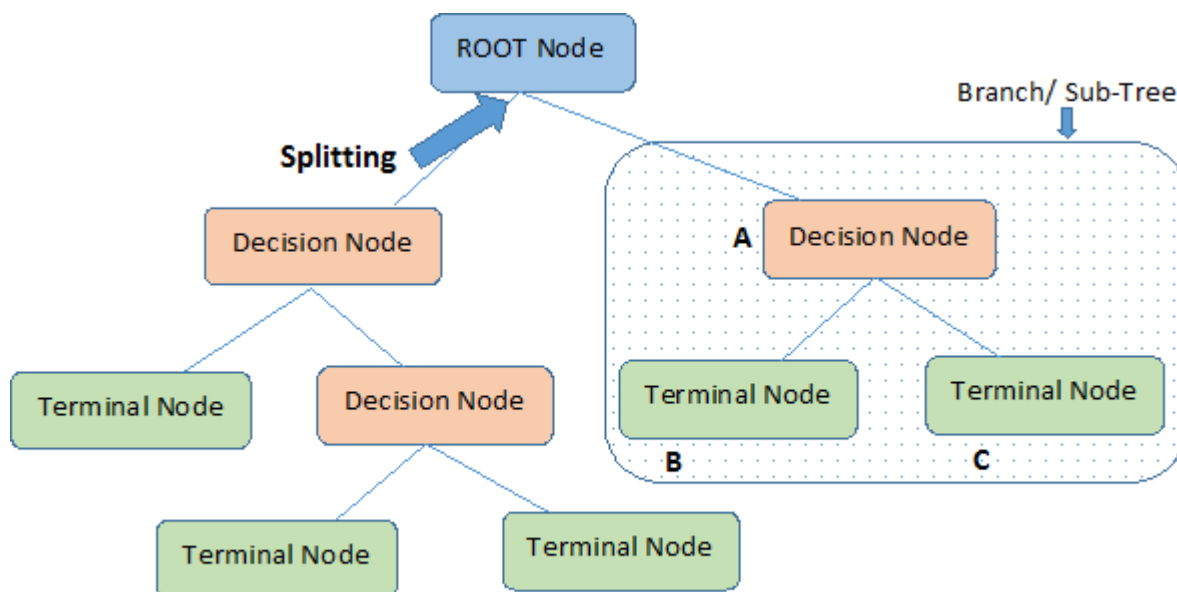
- Memory - 8GB
- Processor - Intel® Core™ i3 CPU M 380 @ 2.53GHz × 4
- OS - Ubuntu 20.04.2 LTS 64 bit

## METHODS AND ALGORITHMS USED

During the course of this project, a variety of Machine Learning, Deep Learning algorithms, and Statistical methods have been used and a brief description of them is given in this chapter.

### DECISION TREE :

Decision trees can be used for both classification and regression tasks. It works with both continuous and categorical input and output values. It works with very little data and Standardization or Normalization is performed default by the algorithm.



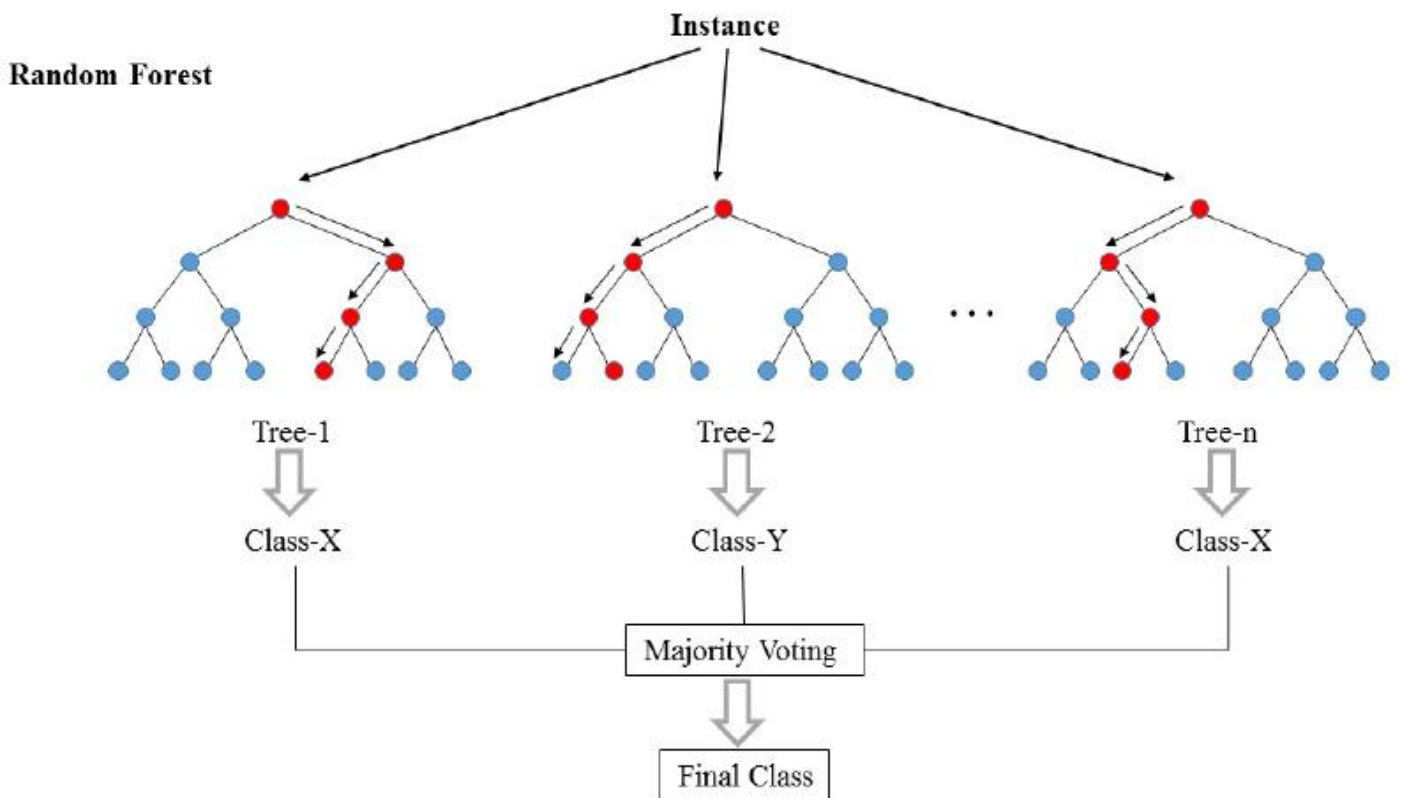
**Note:-** A is parent node of B and C.

### RANDOM FOREST:

Random Forests is the best-known ensemble method. It uses a set of decision trees each of which has a fixed depth and random features. The algorithm constructs a decision tree of similar length for every tree using some randomly chosen features and the final prediction is an aggregation of predictions from all the trees. This method works well and has proven to give better performances.

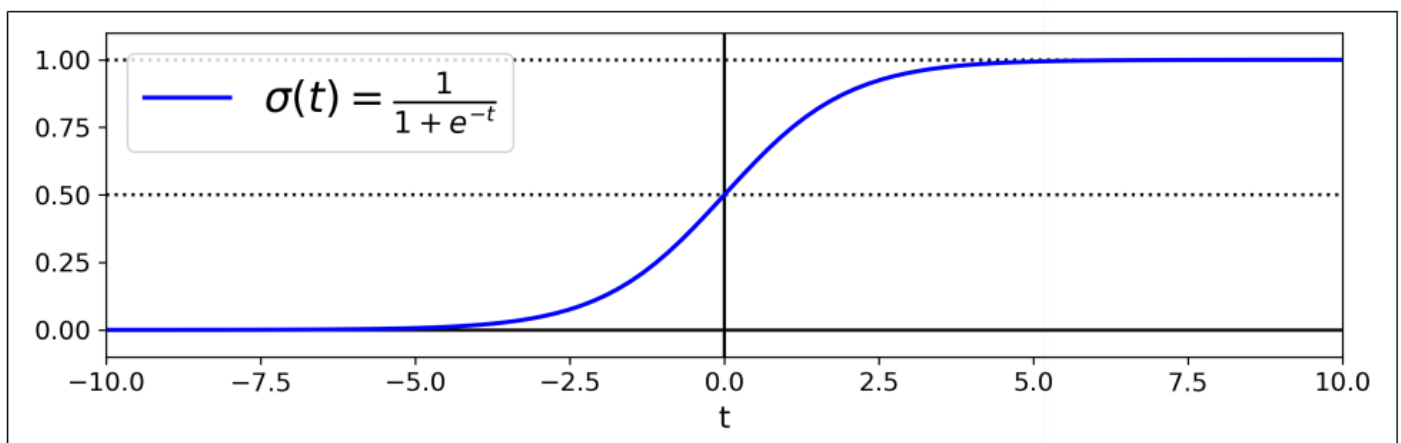
### HYPERPARAMETERS:

- max\_depth, min\_sample\_split, max\_leaf\_nodes, min\_samples\_leaf, n\_estimators, max\_sample (bootstrap sample), max\_feature.



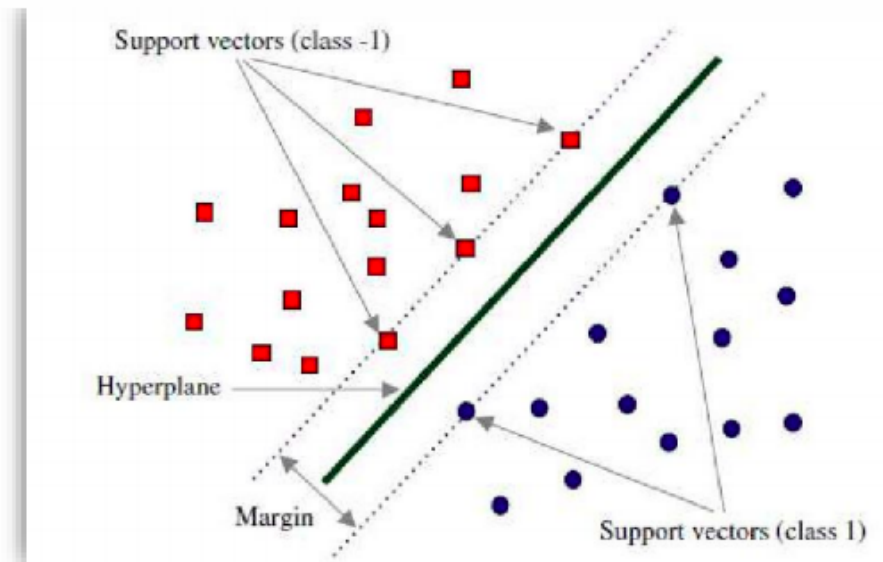
## LOGISTIC REGRESSION :

Logistic Regression is a binary classification algorithm that can be used if the outcome is either 1 or 0, True or False. Threshold by default is 50%, If the estimated probability is greater than the threshold value the model predicts as 1 else 0.



## SUPPORT VECTOR MACHINES

A Support vector machine is a supervised learning model that is used for binary classification. It uses some data points called support vectors to construct a Hyperplane that separates the two classes of data points. Though there may be many hyperplanes that may be separating the two classes, the best plane that has maximum separation, or margin, between the two classes is chosen.



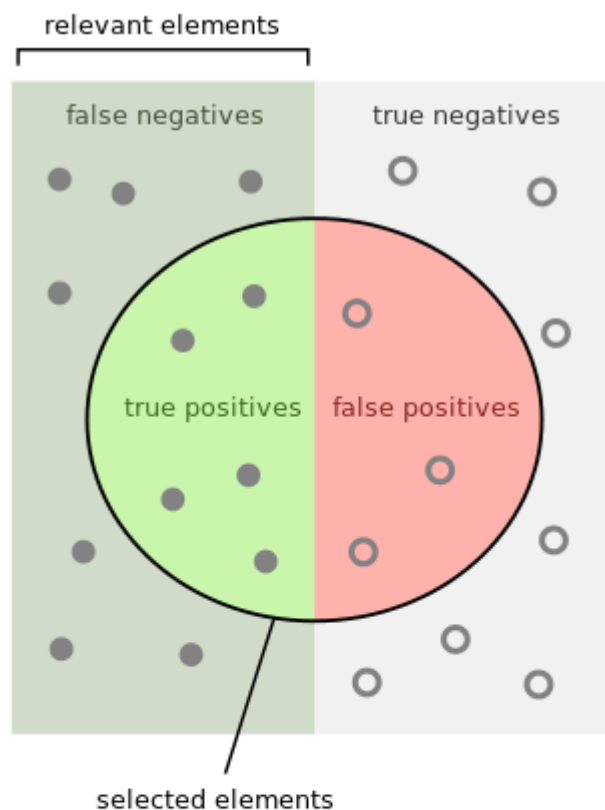
## CONFUSION MATRIX :

A confusion matrix is used to analyze the performance of a classification model. It provides true classifications and miss classifications for each class variable. Classification Report provides insights from the confusion matrix by calculating few metrics such as Precision, recall, and f1\_score.

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

## CLASSIFICATION REPORT:

- This contains 4 key metrics : Precision, Recall, F1-Score, Support.
  - **Precision:** This gives the percentage of True Positive predictions.
  - **Recall:** True positive / Total no of data points in that class.
  - **F1-Score:**  $(\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$ .
  - **Support:** Total no of data points in that particular class.
- Classification reports come in handy when the problem is multiclass classification and when the data is imbalanced.



How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

## DATA COLLECTION

The Datasets for the problem statement have been taken from Kaggle.

1. IPL ball by ball 2008-2020.csv
2. Matches.csv
3. Players.csv

### ATTRIBUTES OF DATA:

- IPL ball by ball 2008-2020:
  - ID - Unique Identifier for each match.
  - Inning - {1,2} representing first or second innings.
  - Over - {0-19} representing overs.
  - Ball - {1-6} ball in that over.
  - Batsman- Name of the batsman who is on strike for that ball.
  - Non\_striker - Name of the batsman on the non-striker end.
  - Bowler - Name of the bowler for the current over.
  - Batsman\_runs - No of Runs batsman scored in that particular ball.
  - Extra\_runs - Extras given by bowler in that particular ball.
  - Total\_runs - Batsman\_runs + Total\_runs.
  - Is\_wicket - {0,1} - 1 - Fall of wicket in that ball, 0 - No wicket in that ball
  - Dismissal\_kind - Way the Batsman got out.
  - Player\_dismissed - Name of the batsman who got dismissed.
  - Fielder - Name of the player who dismissed that batsman.
  - Extras\_type - Type of extra that the bowler has given (Wide, No-ball, etc..)
  - Batting\_team - Name of the Batting team for this inning.
  - Bowling\_team - Name of the bowling team for this inning.

- Matches.csv

- Id - Match ID.
- City - City where the match is being held.
- Date - Date on which the match had happened.
- Season - season number of IPL
- Player\_of\_match - Player of the match.
- Venue - a place of the stadium.
- Neutral\_venue - 0 / 1.
- Team1 - Team 1 name.
- Team2 - Team 2 name.
- Toss\_winner - Name of the team that had won the toss.
- Toss\_decision - Batting/Bowling decision by the toss-winning team.
- Winner - winner of the match.
- Result - Won by wickets / Runs.
- Result\_margin - Difference between 1st inning and 2nd inning.
- Eliminator - Eliminator matches for a season.
- Method - Duckworth-Lewis-Stern or DLS method (D/L).
- Umpire1 - Name of 1st Umpire.
- Umpire2 - Name of 2nd Umpire.

- Players.csv

- Player\_Name - Name of the Player.
- DOB - Date of birth.
- Batting\_Hand - Right/Left.
- Bowling\_Style - style of Bowling.
- Country - Nation of the Batsman.



## PRE-PROCESSING

- Removed duplicate team names in the IPL ball by ball dataset.

```
BBB.replace('Bangalore', 'Bengaluru', inplace=True)

BBB.batting_team.replace({'Rising Pune Supergiants': 'Rising Pune Supergiant'}, regex=True, inplace=True)

BBB.batting_team.replace({'Delhi Daredevils': 'Delhi Capitals'}, regex=True, inplace=True)

BBB.bowling_team.replace({'Delhi Daredevils': 'Delhi Capitals'}, regex=True, inplace=True)

BBB.bowling_team.replace({'Rising Pune Supergiants': 'Rising Pune Supergiant'}, regex=True, inplace=True)
```

- In Bowlers data, the Bowling style (Right arm googly) is divided into two columns ARM and SYLE.T
- Arranging IPL data by over w.r.t balls.

```
TEMP = TEMP.sort_values(['over', 'ball'], ascending=[True, True])
```

## FEATURE ENGINEERING FOR PLAYERS

- **BATSMAN:**

- No Of Matches Played - No of Matches played by the batsman.
- No Of Not Outs
- Total Runs - Total Runs of the Batsman in IPL
- No Of Balls Faced - Total No Of Balls Faced by the Batsman.
- Highest Score - The highest score by the player in IPL.
- Average Score - No of runs scored / No of Outs
- Strike Rate -  $(\text{No of runs scored} / \text{No of Balls faced}) * 100$
- 100's - Total No Of Hundreds
- 50's - Total No Of the Fifties.
- 4's - Total No Of Fours.
- 6's - Total No Of Sixes.

- **BOWLER:**

- Mat - Matches played by the bowler.
- Inns - No Of Innings the Bowler has bowled.
- Overs - No of overs bowled by the bowler.
- Runs - No of runs given by the bowler.
- Wickets - No of wickets taken.
- Average - Average of the Bowler (Runs/Wickets)
- Economy - Economy Rate of the bowler.
- SR - A strike rate of the bowler
- 4Wkts - No of 4 wickets taken by the bowler.
- 5Wkts - No of 5 wickets taken by the bowler.

## CODE FOR BATSMAN FEATURES.

```
import pandas as pd
import numpy as np

BBB = pd.read_csv("datasets/IPL Ball-by-Ball 2008-2020.csv")
MATCHES = pd.read_csv("datasets/MATCHES.csv")

# BATSMAN ATTRIBUTES...
def compute(x):
    l = list()
    # No of Matches Played
    NoOfMatchesPlayed = len(list(x["id"].unique()))
    l.append(NoOfMatchesPlayed)
    # No of Not Outs
    NoOf0uts = x["is_wicket"].sum()
    l.append(NoOfMatchesPlayed - NoOf0uts)
    # Total Runs
    l.append(x["batsman_runs"].sum())
    # No Of Balls Faced
    l.append(x["batsman_runs"].count())
    # Highest Score
    l.append(x.groupby(["id"])["batsman_runs"].sum().max())
    # Average Score
    if NoOf0uts == 0:
        l.append(x["batsman_runs"].sum())
    else:
        l.append(round(x["batsman_runs"].sum() / NoOf0uts, 2))
    # Strike Rate
    l.append(round((x['batsman_runs'].sum() / x['ball'].count()) * 100, 2))
    # 100's
    hundred = 0
```

```

fifty = 0

runs = x.groupby(["id"])["batsman_runs"].sum()

for e in runs:
    if e > 99:
        hundred = hundred + 1
    elif e > 49:
        fifty = fifty + 1
l.append(hundred)

# 50's

l.append(fifty)

# 6's

six = 0
four = 0

for b in x["batsman_runs"]:
    if b == 6:
        six = six + 1
    elif b == 4:
        four = four + 1
l.append(six)

# 4's

l.append(four)

return (pd.Series(l, index=["Matches Played", "Not Outs", "Runs", "Balls
    Faced", "Highest Score", "Average", "Strike Rate", "100's", "50's", "6's", "4's"]))

```

batsman	Matches Played	Not Outs	Runs	Balls Faced	Highest Score	Average	Strike Rate	100's	50's	6's	4's
A Ashish Reddy	23.0	8.0	280.0	196.0	36.0	18.67	142.86	0.0	0.0	15.0	16.0
A Chandila	2.0	1.0	4.0	7.0	4.0	4.0	57.14	0.0	0.0	0.0	0.0
A Chopra	6.0	1.0	53.0	75.0	24.0	10.6	70.67	0.0	0.0	0.0	7.0
A Choudhary	3.0	1.0	25.0	20.0	15.0	12.5	125.0	0.0	0.0	1.0	1.0
A Dananjaya	1.0	1.0	4.0	5.0	4.0	4.0	80.0	0.0	0.0	0.0	0.0

## MODEL - 1

### MODEL USED:

Random Forest.

### ATTRIBUTES PASSED TO THE MODEL: (X Values)

#### Batsman:

- Average Score.
- Strike Rate.
- Not Outs.
- 6's
- 4's

#### Bowler:

- Wickets
- 3 Wickets
- 4 Wickets
- Economy
- Wicket Taking Ability/SR
- Average

#### Match:

- Over
- Ball

### TARGET VARIABLE:

- Batsman\_runs.

## CODE

```
def stacking(name, model, train, y, test, ytest, n_fold):
    folds = StratifiedKFold(n_splits=n_fold, random_state=1, shuffle=True)
    test_pred = np.empty((test.shape[0], 1), float)
    train_pred = np.empty((0, 1), float)
    for train_indices, val_indices in folds.split(train, y.values):
        x_train, x_val = train.iloc[train_indices], train.iloc[val_indices]
        y_train, y_val = y.iloc[train_indices], y.iloc[val_indices]

        model.fit(X=x_train, y=y_train)
        train_pred = np.append(train_pred, model.predict(x_val))
        test_pred = np.append(test_pred, model.predict(test))
    y_pred = model.predict(test)
    print("Confusion Matrix \n", confusion_matrix(ytest, y_pred))
    print("Classification Report \n", classification_report(ytest,
                                                            y_pred, zero_division=1))

    print('Accuracy of ' + name + ' classifier on test set: {:.4f}'.format(
        metrics.accuracy_score(ytest, y_pred)))
    labels = ["R", "B", "W"]
    cm_analysis(ytest, y_pred, labels, name)
    return test_pred.reshape(-1, 1), train_pred

from sklearn.tree import DecisionTreeClassifier
models = [('Random Forest', RandomForestClassifier(n_estimators=100, random_state=7)),
          ('SVM', SVC(gamma='auto', random_state=7)), ('KNN', KNeighborsClassifier()),
          ('Decision Tree Classifier', DecisionTreeClassifier(random_state=7)),
          ('Gaussian NB', GaussianNB())
          ]

for name, model1 in models:
    test_pred1, train_pred1 = stacking(name=name, model=model1, n_fold=10, train=X_Train,
    test=X_Test, y=Y_Train, ytest=Y_Test)
```

## CLASSIFICATION REPORT

RANDOM FOREST	PRECISION	RECALL	F1-SCORE	SUPPORT
-1	0.00	0.00	0.00	14
0	0.39	0.55	0.45	100
1	0.38	0.42	0.40	106
2	0.12	0.05	0.07	19
3	1.00	0.00	0.00	1
4	0.13	0.06	0.08	35
6	0.00	0.00	0.00	13

WEIGHTED AVERAGE F1-SCORE: 0.32

ACCURACY OF RANDOM FOREST: 0.35

SVM	PRECISION	RECALL	F1-SCORE	SUPPORT
-1	1.00	0.00	0.00	14
0	0.46	0.62	0.53	100
1	0.44	0.64	0.52	106
2	1.00	0.00	0.00	19
3	1.00	0.00	0.00	1
4	1.00	0.00	0.00	35
6	1.00	0.00	0.00	13

WEIGHTED AVERAGE F1-SCORE: 0.38

ACCURACY OF SVM: 0.45

KNN	PRECISION	RECALL	F1-SCORE	SUPPORT
-1	0.00	0.00	0.00	14
0	0.37	0.57	0.45	100
1	0.40	0.42	0.41	106
2	0.12	0.05	0.07	19
3	1.00	0.00	0.00	1
4	0.08	0.03	0.04	35
6	0.00	0.00	0.00	13

WEIGHTED AVERAGE F1-SCORE: 0.32

ACCURACY OF KNN: 0.35

DECISION TREE	PRECISION	RECALL	F1-SCORE	SUPPORT
-1	0.00	0.00	0.00	14
0	0.39	0.63	0.48	100
1	0.38	0.39	0.38	106
2	0.00	0.00	0.00	19
3	1.00	0.00	0.00	1
4	0.14	0.03	0.05	35
6	0.00	0.00	0.00	13

WEIGHTED AVERAGE F1-SCORE: 0.32

ACCURACY OF DECISION TREE: 0.36

NAIVE BAYES	PRECISION	RECALL	F1-SCORE	SUPPORT
-1	1.00	0.00	0.00	14
0	0.46	0.70	0.56	100
1	0.43	0.56	0.49	106
2	1.00	0.00	0.00	19
3	1.00	0.00	0.00	1
4	1.00	0.00	0.00	35
6	1.00	0.00	0.00	13

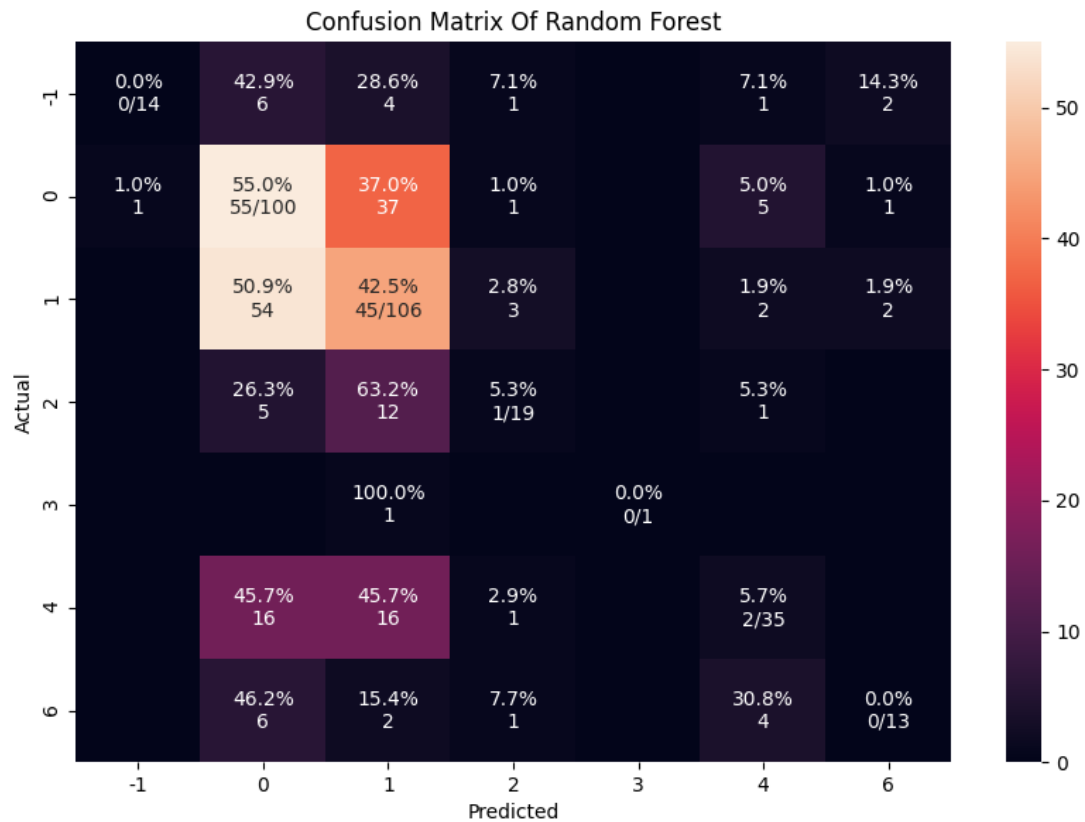
WEIGHTED AVERAGE F1-SCORE: 0.37

ACCURACY OF NAIVE BAYES: 0.44

#### DESCRIPTION:

- Five Different classification models have been executed on the Dataset. They are Random Forest, SVM, KNN, Decision Tree, Naive Bayes.
- In the case of classification rather than accuracy as a measure of metric F1 score should be considered because there might not be equal distribution of classes among the test data.
- Therefore higher the F1 Score, the better the classification model.
- Since the misclassifications are distributed across the classes, let's combine classes into groups. [0,1,2,3] to Runs, [4,6] to boundaries and [-1] to wickets.
- Now from 7 classes classification, the model has reduced to 3 classes classification.





### CODE FOR COMBINING CLASSES:

```

Target = list()

for i in range(len(DATA)):
    if DATA.iloc[i,5] == 1:
        Target.append("W")
    elif DATA.iloc[i,4] >= 4:
        Target.append("B")
    else:
        Target.append("R")

Target = pd.Series(Target)

```

## MODEL - 2

### CLASSIFICATION REPORT

<b>RANDOM FOREST</b>	<b>PRECISION</b>	<b>RECALL</b>	<b>F1-SCORE</b>	<b>SUPPORT</b>
<b>B</b>	0.44	0.08	0.14	48
<b>R</b>	0.80	0.99	0.88	226
<b>W</b>	1.00	0.00	0.00	14

WEIGHTED AVERAGE F1 SCORE: 0.72

ACCURACY OF RANDOM FOREST: 0.78

<b>SVM</b>	<b>PRECISION</b>	<b>RECALL</b>	<b>F1-SCORE</b>	<b>SUPPORT</b>
<b>B</b>	1.00	0.00	0.00	48
<b>R</b>	0.78	1.00	0.88	226
<b>W</b>	1.00	0.00	0.00	14

WEIGHTED AVERAGE F1 SCORE: 0.72

ACCURACY OF SVM: 0.78

<b>KNN</b>	<b>PRECISION</b>	<b>RECALL</b>	<b>F1-SCORE</b>	<b>SUPPORT</b>
<b>B</b>	0.33	0.12	0.18	48
<b>R</b>	0.80	0.96	0.87	226
<b>W</b>	1.00	0.00	0.00	14

WEIGHTED AVERAGE F1 SCORE: 0.72

ACCURACY OF KNN: 0.78

DECISION TREE	PRECISION	RECALL	F1-SCORE	SUPPORT
B	0.25	0.08	0.12	48
R	0.80	0.96	0.87	226
W	1.00	0.00	0.00	14

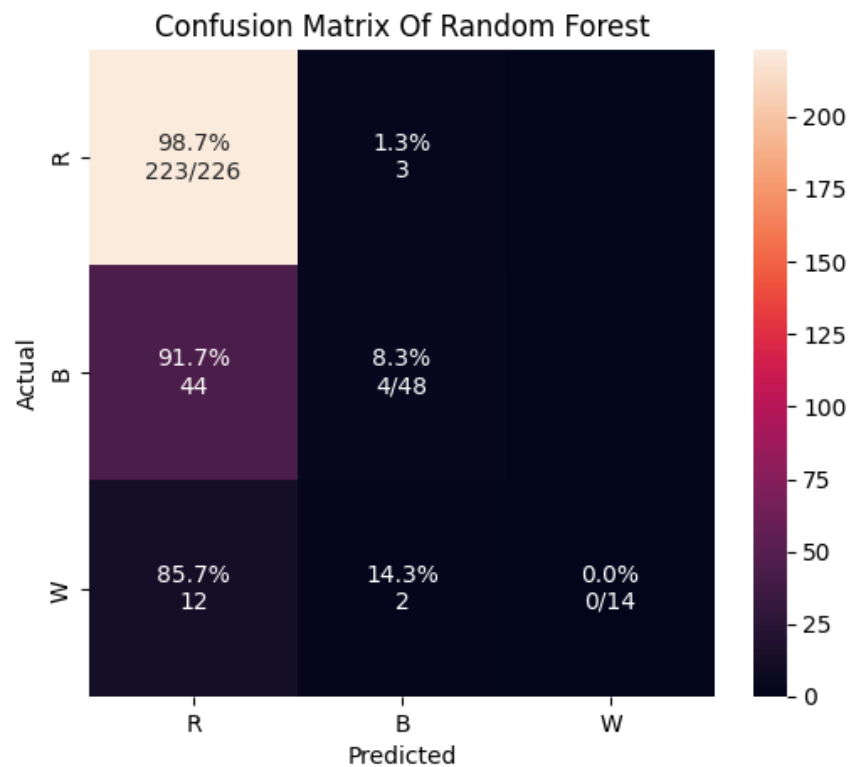
WEIGHTED AVERAGE F1 SCORE: 0.70

ACCURACY OF DECISION TREE: 0.76

NAIVE BAYES	PRECISION	RECALL	F1-SCORE	SUPPORT
B	1.00	0.00	0.00	48
R	0.78	1.00	0.88	226
W	1.00	0.00	0.00	14

WEIGHTED AVERAGE F1 SCORE: 0.69

ACCURACY OF NAIVE BAYES: 0.78



## **DESCRIPTION:**

- From this model, it can be observed that the F1 score and the accuracy has increased at the same time only the Runs class has been classified properly.
- The majority of boundaries are also classified as runs.
- The classification of wickets is 0.

## **CHANGES TO BE PERFORMED IN THE MODEL :**

- New features need to be added in order to classify these three classes.
- One of the main observations is that, Since the stats of individual players are constant values and the same values get repeated in all the rows.
- Therefore dynamic values should be passed, which would in turn create differentiation between each observation and might lead to better classification.

## **FEATURE ENGINEERING (Last 6 balls)**

- Runs In Last 6 balls - Total No of Runs scored by the team in the last 6 balls.
- Fall Of Wickets In Last 6 balls - No of wickets fell in the last six balls.
- No\_Of\_Dots\_In\_Last\_Over - No of Dots in the last six balls
- No\_Of\_Singles\_In\_Last\_Over - No of singles in the last six balls
- No\_Of\_Doubles\_In\_Last\_Over - No of doubles in the last six balls
- No\_Of\_Fours\_In\_Last\_Over - No of fours in the last six balls
- No\_Of\_Sixes\_In\_Last\_Over - No of sixes in the last six balls
- Team\_Score - Team score from the beginning to the previous ball.
- Strike\_Rate - Batsman strike rate in this match.
- Economy\_Rate - Bowlers economy rate in this match.
- P/NP - Weather the current over falls under powerplay or non-powerplay.
- Percentage P/NP - In the last 6 balls how many were powerplay balls.
- Pressure - ( Target - Team score )

### MODEL 3

#### CLASSIFICATION REPORT

LOGISTIC REGRESSION	PRECISION	RECALL	F1-SCORE	SUPPORT
B	0.46	0.80	0.59	15
R	0.93	0.73	0.81	51

WEIGHTED AVERAGE F1 SCORE: 0.76

ACCURACY OF LOGISTIC REGRESSION: 0.74

RANDOM FOREST	PRECISION	RECALL	F1-SCORE	SUPPORT
B	0.43	0.20	0.27	15
R	0.80	0.92	0.85	51

WEIGHTED AVERAGE F1 SCORE: 0.72

ACCURACY OF RANDOM FOREST: 0.75

SVM	PRECISION	RECALL	F1-SCORE	SUPPORT
B	1.00	0.07	0.12	15
R	0.78	1.00	0.88	51

WEIGHTED AVERAGE F1 SCORE: 0.71

ACCURACY OF SVM: 0.78

KNN	PRECISION	RECALL	F1-SCORE	SUPPORT
B	0.50	0.20	0.29	15
R	0.80	0.94	0.86	51

WEIGHTED AVERAGE F1 SCORE: 0.73

ACCURACY OF KNN: 0.77

DECISION TREE	PRECISION	RECALL	F1-SCORE	SUPPORT
B	0.42	0.67	0.51	15
R	0.88	0.73	0.80	51

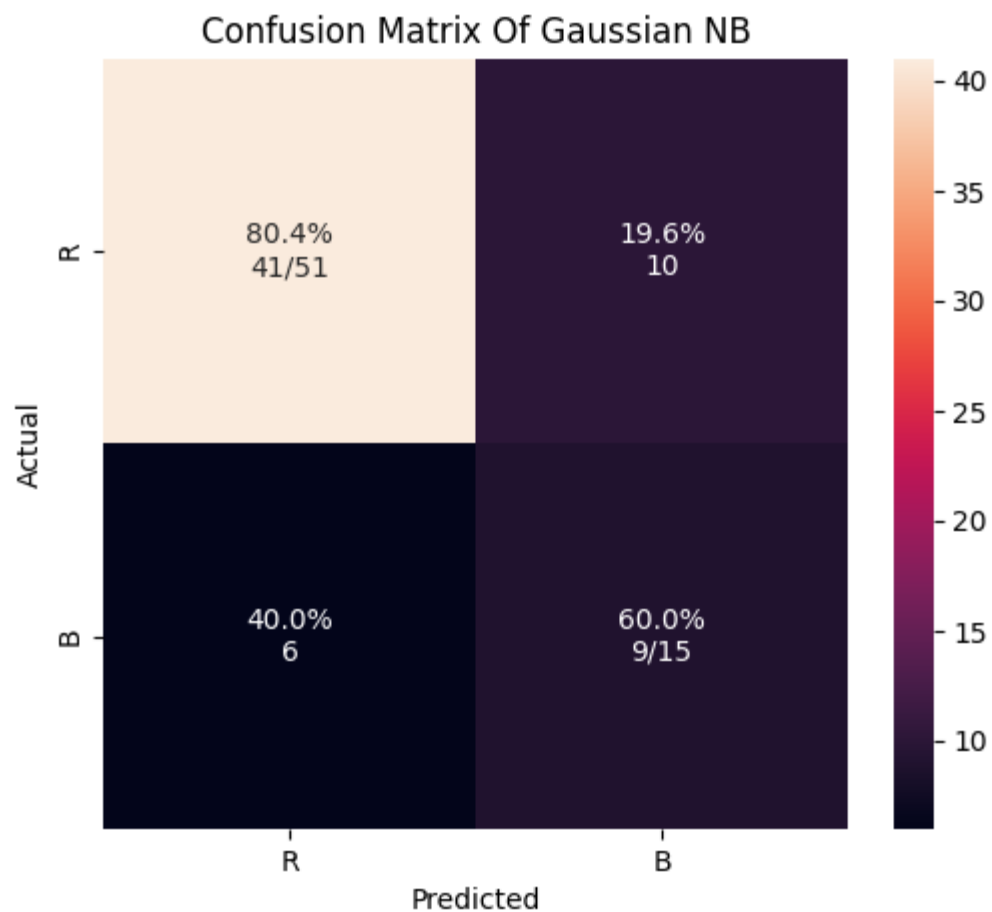
WEIGHTED AVERAGE F1 SCORE: 0.73

ACCURACY OF DECISION TREE: 0.71

NAIVE BAYES	PRECISION	RECALL	F1-SCORE	SUPPORT
B	0.47	0.60	0.53	15
R	0.87	0.80	0.84	51

WEIGHTED AVERAGE F1 SCORE: 0.77

ACCURACY OF NAIVE BAYES: 0.75



## **DESCRIPTION:**

- Finally the model has broken to a binary classification problem, where it predicts whether the ball's outcome would be a boundary or not.
- The class wicket class has been removed because the proportion of wickets w.r.t other classes is very less and it's challenging for the model to differentiate between these classes.
- It can be observed from the above mentioned classification reports that Gaussian Naive Bayes performs best on the test data.

## **CONCLUSION:**

- In the beginning the models were trained on multiclass classification with 7 classes. Since the predictions were distributed among near classes, combined classes into Runs, Boundaries and Wickets.
- Later Dynamic Features were added to the model.
- Since the prediction of wicket being zero, class wicket was removed and the problem broke down to binary classification with whether the outcome would be a Run or a Boundary.
- Gaussian Naive Bayes algorithm turned out to be performing well on the test data.

## **FUTURE SCOPE:**

- Since the wickets were not predicted, could create a separate model for predicting a wicket.
- Implementing the same with Neural Networks.

## **REFERENCES**

1. <https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/> - Ensemble Learning.
2. <https://medium.com/@kohlishivam5522/understanding-a-classification-report-for-your-machine-learning-model-88815e2ce397> - Classification Report.

## HEALTHY DIET RECOMMENDATION

### TABLE OF CONTENTS

1. INTRODUCTION.	33
2. DATA COLLECTION.	34
3. LITERATURE STUDY.	35
4. METHODS AND ALGORITHMS USED	36
a. SEABORN.	36
b. K-MEANS.	36
c. ELBOW METHOD.	37
5. ANALYSIS.	38
6. CLUSTERING.	39
7. DAILY VALUES.	40
8. SOCIAL NETWORK GRAPH.	41
a. DISTANCE MEASURE.	41
b. REPRESENTATION OF DATASET AS A SOCIAL NETWORK GRAPH.	41
c. SOCIAL NETWORKING GRAPH WITH 5 ITEMS.	42
d. BI-PARTITE GRAPH.	43
9. CONCLUSION.	44
10. REFERENCES.	44

### TABLE OF FIGURES

1. SEABORN.	36
2. K-MEANS.	37
3. ELBOW METHOD.	37
4. NUTRIENT CONTENT IN BEEF & PORK.	38
5. NUTRIENT CONTENT IN SALADS.	38
6. ELBOW METHOD.	39
7. CLUSTER.	39
8. DV VALUES.	40
9. SOCIAL NETWORKING GRAPH.	42
10. BI-PARTITE GRAPH.	43



## INTRODUCTION

Right nutrition is the oxygen to healthy life. It is not about consuming less quantity but about consuming the right quantity that makes a difference. We consume a variety of items on a daily basis but do we know how healthy each item is.?

Nutrition and health go hand in hand. Since what we eat is in our hands, our health too is under our control. With several diseases like hypertension, diabetes, hormonal imbalances on the rise, it becomes extremely important to know “what to eat” and “how much to eat”.

Every time we consume something, if we could analyse its contribution to a healthy lifestyle, we could probably eradicate most of the nutritional disorders from the world. Afterall, we are as healthy as what we consume.

A balanced diet is one which has all the necessary nutrients in the right proportion; it is as simple as “Eat Healthy”, “Live Healthy”. This project aims at providing a health food recommendation from McDonalds menu. Like swami says

“Mind is born of the food you take. As is the food, so is the mind.

Therefore, Healthy Food = Healthy mind = Healthy you.

## DATA COLLECTION

- The Dataset contains dishes and its respective nutrient values of McDonalds US.

### ATTRIBUTES OF DATA:

- Category - There are a total of 9 categories:
  - Breakfast, Beef & Pork, Chicken & Fish, Salads, Snacks & Sides, Desserts, Beverages, Coffee & Tea, Smoothies & Shakes.
- Item - Name of the dish.
- Serving size - Dish serving weight. Measured in ounces.
- Calories - Amount of calories in the dish.
- Total Fat - Amount of fat in the food.
- Calories from fat - Amount of calories from fat.
- Total Fat (%DV) - Daily value of fat this dish provides.
- Saturated Fat - Amount of Saturated fat in the dish.
- Saturated Fat (%DV) - Daily value of Saturated fat this dish provides.
- Trans Fat - Amount of Trans Fat in the dish.
- Cholesterol - Amount of Cholesterol in the dish.
- Cholesterol (%DV) - Daily value of Cholesterol this dish provides.
- Sodium - Amount of Sodium in the dish.
- Sodium (%DV) - Daily value of Sodium this dish provides.
- Carbohydrates - Amount of Carbohydrates in the dish.
- Carbohydrates (%DV) - Daily value of Carbohydrates this dish provides.
- Dietary Fiber - Amount of Dietary Fiber in the dish.
- Dietary Fiber (%DV) - Daily value of Dietary Fiber this dish provides.
- Sugars - Amount of Sugars in the dish.
- Protein - Amount of Protein in the dish.
- Vitamin A (%DV) - Daily value of Vitamin A this dish provides.

- Vitamin C (%DV) - Daily value of Vitamin C this dish provides.
- Calcium (%DV) - Daily value of Calcium this dish provides.
- Iron (%DV) - Daily value of Iron this dish provides.

### **PRE-PROCESSING**

- Serving size - Converted from ounces to grams, femtoliter to ml.
- Sodium - Converted milligrams to grams.
- Vitamin A - Converted milligrams to grams.
- Vitamin C - Converted milligrams to grams.
- Calcium - Converted milligrams to grams.
- Cholesterol - Converted milligrams to grams
- Conversion of %DV to grams and milligrams:
  - Since Vitamins and minerals are in DV scale and other nutrient values are in grams and milligrams. The Daily consumption of these nutrient values were obtained from the internet and converted %DV to g and mg.

### **LITERATURE STUDY**

- Calories define the amount of energy that can be obtained from a food item.
- Saturated Fat, Trans Fat and Cholesterol are part of Total Fat.
- Sodium is an electrolyte that maintains the shape of cells, high or low consumption of sodium leads to high or low Blood pressure respectively.
- Sugars are part of carbohydrates.
- Dietary Fibre are from plant sources and cant be digested, but it's required for the absorption of food in the intestine.
- Proteins are required for muscle growth.
- Vitamin A required for eyes.
- Vitamin C helps in maintaining immunity.
- Iron helps in increasing hemoglobin.
- Calcium helps to maintain the strength of bones.

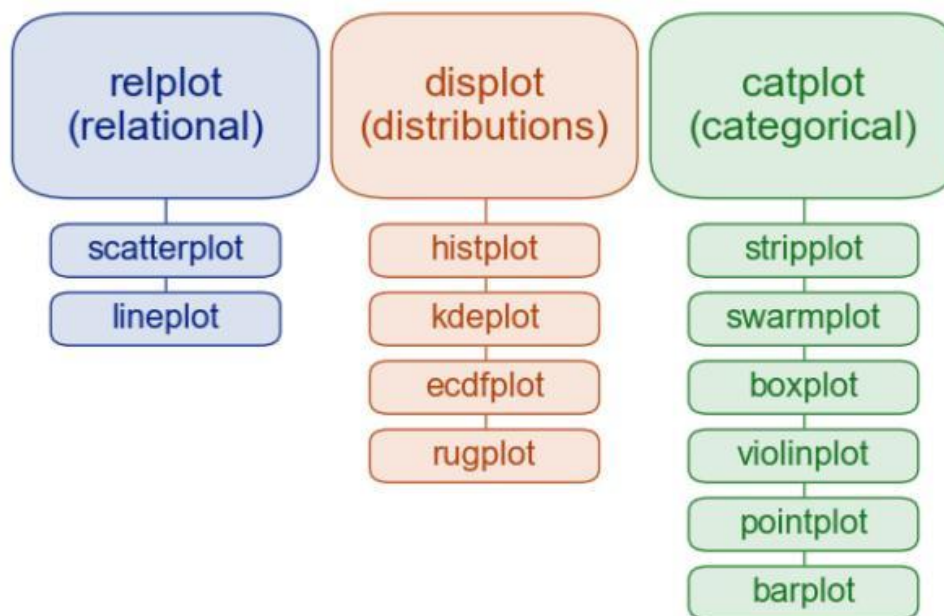
## METHODS AND ALGORITHMS USED

During the course of this project, a variety of Machine Learning algorithms, and python have been used and a brief description of them is given in this chapter.

### SEABORN:

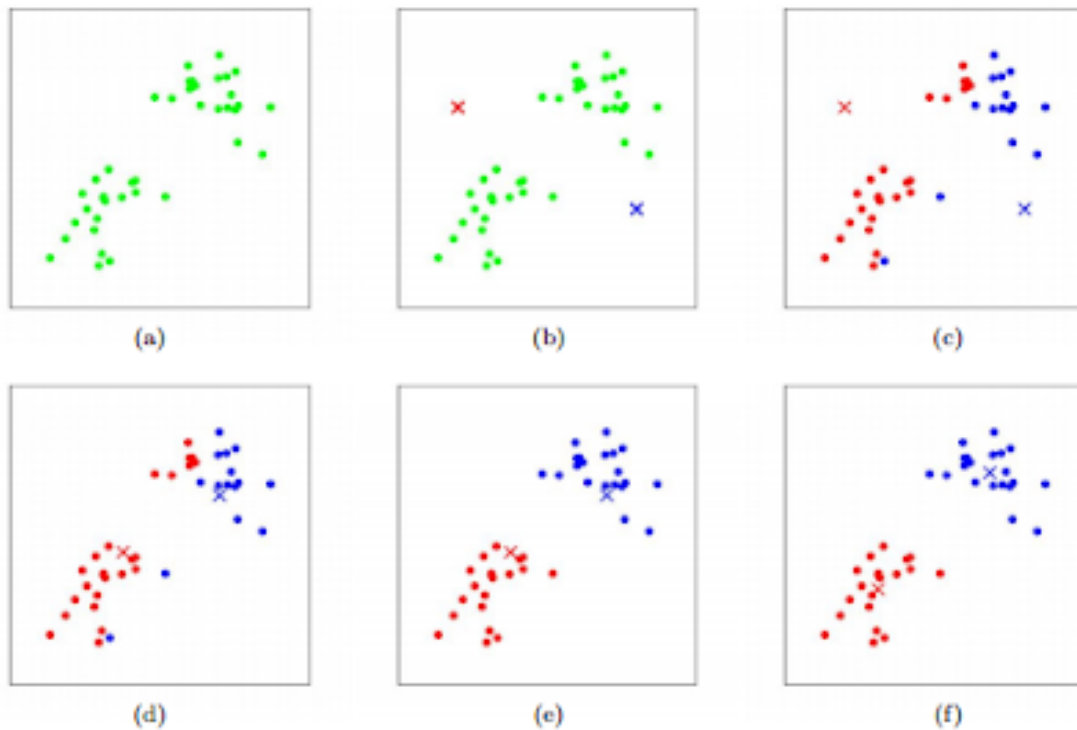
- It is a visualization library in python, which acts as a wrapper on matplotlib and provides high level interactive statistical plots etc.
- This library combines all plotting functions into three categories:
  - Relplot - relational
  - Displot - distributions
  - Catplot - categorical

## FIGURE LEVEL FUNCTIONS AND AXES LEVEL FUNCTIONS



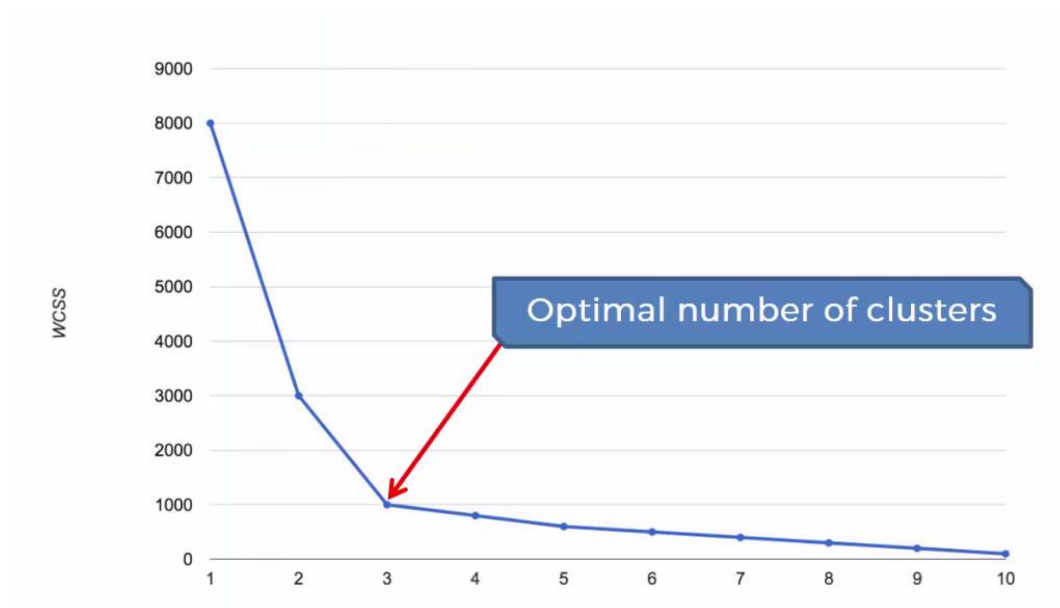
### K-MEANS CLUSTERING:

- It is a machine learning algorithm that comes under unsupervised learning which is used when the data is unlabeled.
- The algorithm works iteratively trying to allot data points to groups by calculating group means and allotting the points to groups which would reduce the within group variance.



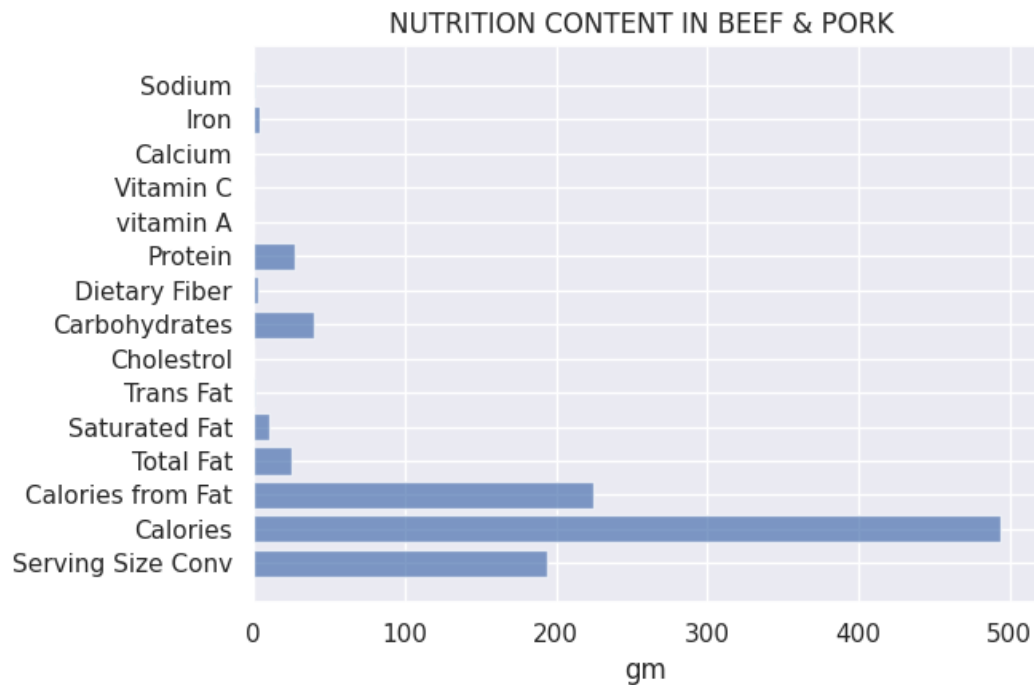
### ELBOW METHOD:

- This method is used to find the optimal number of clusters in a given dataset.
- The algorithm is an iterative one which plots within group variance against no of clusters in the dataset.
- The elbow of the curved is considered as the optimal k value.



## ANALYSIS

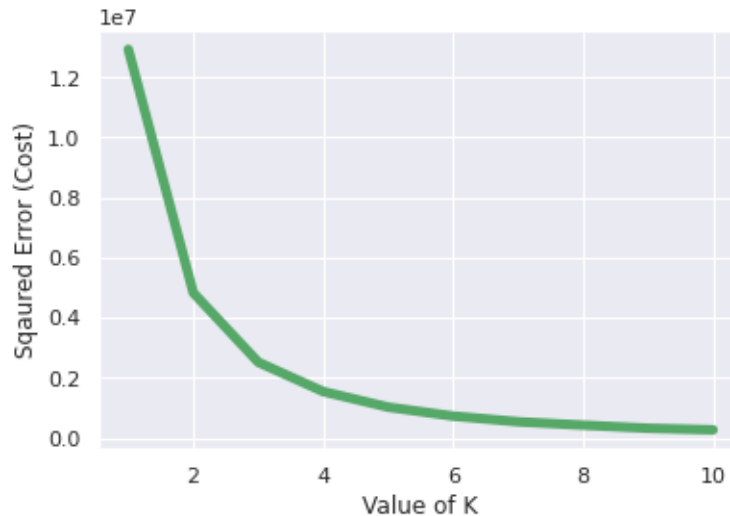
- Average Nutrient values in each category of dishes are plotted below.



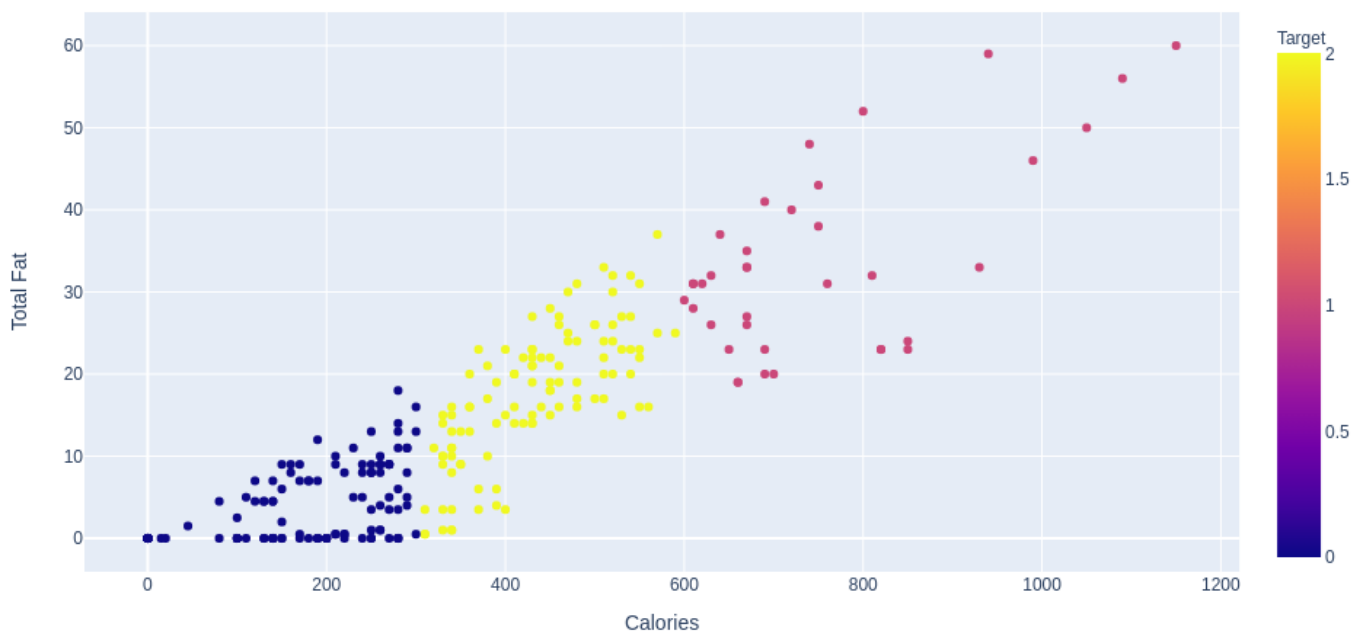
- In the same way nutrient content for each category was plotted and found that major components of a diet are Calories, Total Fat, Protein and carbohydrates

## CLUSTERING

- A subset of data is considered according to the major components of a diet and clustering algorithm is performed on it.
- The Elbow method is used in order to find the optimal number of clusters.



- It can be observed from the above figure that the optimal k is 3.



- Since there are three groups, meal combos can be provided in McDonalds an item from 0 and 2 or two items from 1 etc...

## DAILY VALUES

### MEAL PLAN

## Male Age 19-30

Macronutrients	Value	Vitamins	Value	DRI	%DRI
Calories	2640 kcal	Vitamin A (RAE)	1755 mcg	900 mcg	195%
Total Fat	155 g	Vitamin C	143 mg	90 mg	159%
Saturated Fat	22.7 g	Vitamin D	11 mcg	15 mcg	71%
Trans Fat	0 g	Vitamin E	26 mg	15 mg	176%
Cholesterol	460 mg	Vitamin K	515 mcg	120 mcg	429%
Carbohydrate	219 g	Thiamin	1.3 mg	1.2 mg	104%
Dietary Fiber	28 g	Riboflavin	1.4 mg	1.3 mg	111%
Sugars	103 g	Niacin	26.8 mg	16.0 mg	168%
Added sugar	0 g	Vitamin B6	2.4 mg	1.3 mg	183%
Protein	95 g	Folate (DFE)	336 mcg	400 mcg	84%
Free water	1437 mL	Vitamin B12	3.9 mcg	2.4 mcg	161%
8 pouches Real Food Blends		Pantothenic acid	6.3 mg	5 mg	125%
		Choline	453 mg	550 mg	82%
		<b>Minerals</b>			
		Calcium	476 mg	1000 mg	48%
		Copper	1.91 mg	0.90 mg	212%
		Iron	19 mg	8 mg	233%
		Magnesium	543 mg	400 mg	136%
		Manganese	6.9 mg	2.3 mg	301%
		Phosphorus	1685 mg	700 mg	241%
		Selenium	105 mcg	55 mcg	192%
		Zinc	13 mg	11 mg	120%
		Potassium	3633 mg	3400 mg	107%
		Sodium	480 mg	1500 mg	32%

- Above mentioned meal plan was taken from the internet and the same is available for all different age groups.
- These values can be compared with the dishes available in the McDonalds and dishes can be recommended to the customers.
- The nutrient values can be divided into three portions: breakfast, lunch and dinner. Where breakfast can be provided with 40% of nutrient values, lunch and dinner with 30%.
- In this way it can be made sure that proper and appropriate amounts of nutrients are taken at the same time satisfying one's desire to eat all kinds of food.
- If someone prefers two meals per day then the proportions can be divided accordingly.



## SOCIAL NETWORK GRAPH

- There are three characteristics of social networking graph:
  - Collection of entities, each node is an entity in the graph.
  - There must be at least one edge between these entities.
  - Locality of relationship, if there exists an edge from A to B and A to C, then there is a higher probability than average that B and C could be related to each other.

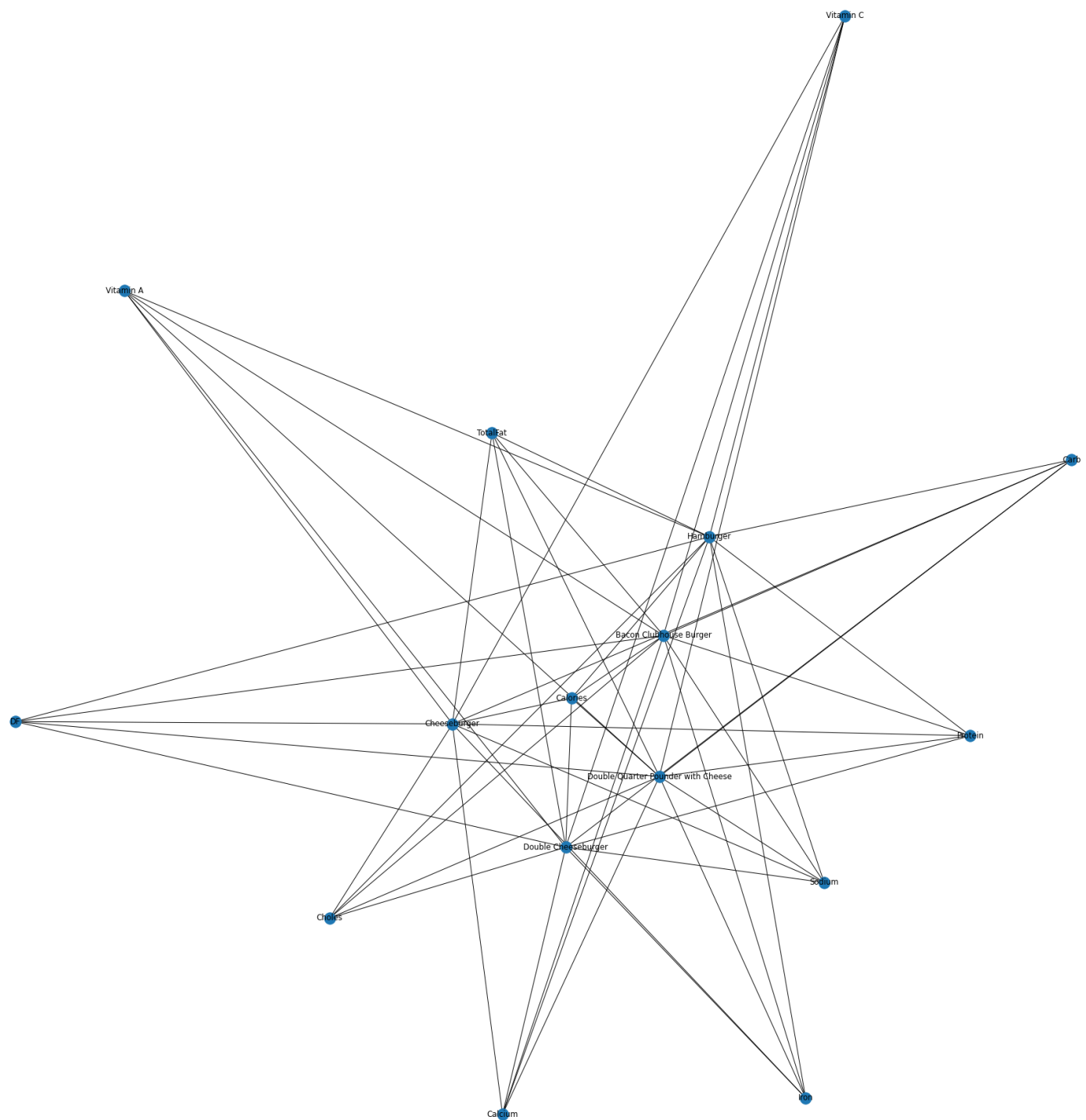
## DISTANCE MEASURE

- Edges represent relationship between entities
- To find Communities, a distance measure should be defined.
- $d(x,y) = 1$  ( if presence of edge between x and y )
- $d(x,y) = 1.5$  ( if edge doesn't exist between x and y ).

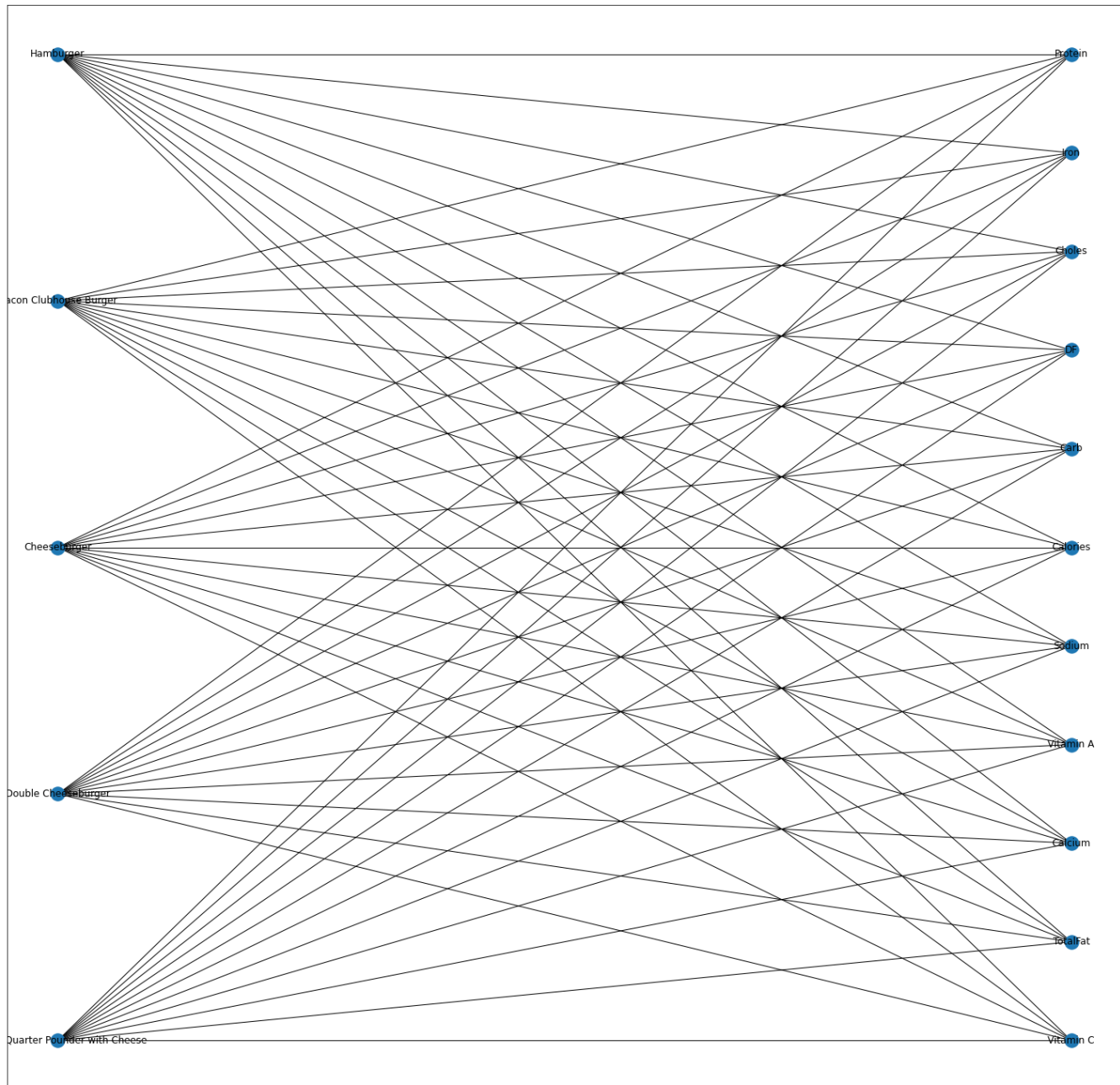
## REPRESENTATION OF DATASET AS A SOCIAL NETWORK GRAPH

- There is an edge from each food item to every other component of the food item (cal, fat, carbs etc..)
- Each edge in our Data set has a weight assigned to it (g,mg,ml)
- Since there is no edge between the food items and no edge between the components of food items. There are two disjoint sets of nodes in the graph.
- These kinds of graphs are called Partite Graphs.
- In our case since there are 2 disjoint sets, our graph is called a Bi-Partite Graph.

SOCIAL NETWORKING GRAPH WITH 5 ITEMS



## BI-PARTITE GRAPH



- It can be seen from the above figure that there are two disjoint sets dishes (5) in the left and nutrients (11) on the right respectively.

## CONCLUSION

Right proportion in every aspect is essential to lead a balanced life. Once proportion is lost, balance is lost. Only an appropriate input assures a right output - in case of human beings, the output being the energy to perform various activities throughout the day. For example, emotions in the right proportion play a pivotal role in shaping one's character.

Laughter, though a positive emotion, works as a medicine in the right proportion. However the same laughter in excess, may cause you to take medicine. Similarly consuming sweets in a higher quantity does not cause much harm in young age groups, but as one ages, it increases the risk of developing diabetes and associated conditions.

In conclusion the right item, in the right quantity at the right time is important to lead a healthy and balanced life.

## REFERENCES

1. <https://seaborn.pydata.org/> - Seaborn
2. <https://pypi.org/project/networkx/> - networkx
3. Mining of Massive Datasets - Jure Leskovec, Anand Rajaraman, Jeffrey David Ullman

