MACHINE LEARNING MODEL TO PREDICT NUMBER OF PEOPLE ATTENDING COLLEGE MESS

A Socially Relevant Project-1 report submitted to

JAWAHARLAL NEHRU TECHONOLOGICAL UNIVERSITY ANANTAPUR
ANANTHAPURAMU

in partial fulfilment of the requirements for
the award of the degree of

BACHELOR OF TECHNOLOGY
in
ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

THOLUSURI POOJITHA	**19121A04M5**
TIRUTHANI SANTHOSH	**19121A04M9**
VUTUKURU KEERTHI REDDY	**19121A04P0**
KAKARLA SIVA GANESH	**19121A04P7**
POONAMALLI DIVYA SAI	**20125A0436**

Under the Esteemed Supervision of

DR.D. LEELA RANI, M. Tech., Ph.D.,
Professor, Department of ECE

SREE VIDYANIKETHAN Engineering College (Autonomous)

Department of Electronics and Communication Engineering
SREE VIDYANIKETHAN ENGINEERING COLLEGE
(AUTONOMOUS)
Sree Sainath Nagar, A. Rangampet, Tirupathi - 517102.

(2019-2023)

SREE VIDYANIKETHAN ENGINEERING COLLEGE

(AUTONOMOUS)

Sree Sainath Nagar, A.Rangampet - 517 102

VISION

To be one of the Nation's premier Engineering Colleges by achieving the highest order of excellence in Teaching and Research.

MISSION

- To foster intellectual curiosity, pursuit and dissemination of knowledge.
- To explore students' potential through academic freedom and integrity.
- To promote technical mastery and nurture skilled professionals to face competition in ever increasing complex world.

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

Vision

To be a center of excellence in Electronics and Communication Engineering through teaching and research producing high quality engineering professionals with values and ethics to meet local and global demands.

Mission

- The Department of Electronics and Communication Engineering is established with the cause of creating competent professionals to work in multicultural and multidisciplinary environments.
- Imparting knowledge through contemporary curriculum and striving for development of students with diverse background.
- Inspiring students and faculty members for innovative research through constant interaction with research organizations and industry to meet societal needs.
- Developing skills for enhancing employability of students through comprehensive training process.
- Imbibing ethics and values in students for effective engineering practice.

B. Tech. (Electronics and Communication Engineering)

Program Educational Objectives

After few years of graduation, the graduates of B.Tech (ECE) will be:

PEO1. Enrolled or completed higher education in the core or allied areas of electronics and communication engineering or management.

PEO2. Successful entrepreneurial or technical career in the core or allied areas of electronics and communication engineering.

PEO3. Continued to learn and to adapt to the world of constantly evolving technologies in the core or allied areas of electronics and communication engineering.

Program Outcomes

On successful completion of the Program, the graduates of B.Tech. (ECE) Program will be able to:

PO1 **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2 **Problem analysis**: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3 **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4 **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5 **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6 **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7 **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8 **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9 **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10 **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11 **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12 **Lifelong learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes

On successful completion of the Program, the graduates of B. Tech. (ECE) will be able to

PSO1. Design and develop customized electronic circuits for domestic and industrial applications.

PSO2. Use specific tools and techniques to design, analyze and synthesize wired and wireless communication systems for desired specifications and applications.

PSO3. Apply suitable methods and algorithms to process and extract information from signals and images in Radar, Satellite, Fiber optic and Mobile communication systems.

Department of Electronics and Communication Engineering

SREE VIDYANIKETHAN ENGINEERING COLLEGE

(AUTONOMOUS)

Sree Sainath Nagar, A.Rangampet - 517102.

Certificate

This is to certify that the Socially Relevant Project-1 Report entitled

MACHINE LEARNING MODEL TO PREDICT NUMBER OF PEOPLE ATTENDING COLLEGE MESS

is the bona fide work done & submitted by

THOLUSURI POOJITHA	**19121A04M5**
TIRUTHANI SANTHOSH	**19121A04M9**
VUTUKURU KEERTHI REDDY	**19121A04P0**
KAKARLA SIVA GANESH	**19121A04P7**
POONAMALLI DIVYA SAI	**20125A0436**

in the Department of Electronics and Communication Engineering, **Sree Vidyanikethan Engineering College, A.Rangampet** *and is submitted to* **Jawaharlal Nehru Technological University Anantapur, Ananthapuramu** *in partial fulfilment of the requirements for the award of degree of Bachelor of Technology in Electronics and Communication Engineering during 2019-2023.*

Supervisor
DR.D. LEELA RANI, M. Tech., Ph.D.,
Professor, Dept. of ECE

Head of the Department
Dr. N. Gireesh, M.Tech., Ph.D.
Professor & Head of Dept. of ECE

Examiner-1

Examiner-2

ACKNOWLEDGEMENTS

We are deeply indebted to the supervisor, **Dr. D. LEELA RANI, M.Tech., Ph.D.,** Professor of Department of Electronics and Communication Engineering for valuable guidance, constructive criticism and keen interest evinced throughout the course of our social relevant project work. We are really fortunate to associate ourselves with such an advising and helping guide in every possible way, at all stages, for the successful completion of this work.

We express our deep sense of gratitude to **Dr. N. GIREESH, M.Tech., Ph.D.,** Professor and Head of the Department of Electronic and Communication Engineering for his valuable guidance and constant encouragement given to us during this social relevant project and the course.

We express gratitude to our principal **Dr. B. M. SATISH, Ph.D.,** for supporting us in completion of our social relevant project work successfully by providing the facilities. We are pleased to express our heart full thanks to our faculty in Department of ECE of Sree Vidyanikethan Engineering College for their moral support and good wishes.

Finally, we have a notation to express our sincere thanks to friends and all those who guided, inspired and helped us in the completion of our social relevant project work.

T. Poojitha - 19121A04M5
T. Santhosh - 19121A04M9
V. Keerthi Reddy - 19121A04P0
K. Siva Ganesh - 19121A04P7
P. Divya Sai - 20125A0436

ABSTRACT

The scarcity and surplus of food in College Mess Management can be avoided by using Machine learning model. It can replace the Face Recognition system which is extravagant. The students' problem of waiting for tokens will be resolved by using this model. Face Recognition system generates paper tokens. Paper wastage can also be reduced. Main aim of this Machine Learning Model is to achieve better accuracy of predictions.

Machine Learning model will be trained by a Dataset. This Dataset contains previous data of number of people attending the Mess. All the parameters affecting the number of people attending the mess, will be analyzed and incorporated in the Dataset, so that the model will be trained effectively.

Python programming Language will be used to build this model and libraries like pandas and numpy are imported. Support Vector Regressor Algorithm is used for this model and to implement this, the dependencies from sklearn library is imported. To make the predictions better, few techniques like standardization, label encoding are used. The statistical analysis is done through plots using a dependency like matplotlib.pyplot.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER - 1

INTRODUCTION

1.1. Introduction

Predictive modeling is a statistical technique using machine learning and data mining to predict and forecast likely future outcomes with the aid of historical and existing data. It works by analyzing current and historical data and projecting what it learns on a model generated to forecast likely outcomes. Predictive modeling can be used to predict just about anything, from TV ratings and a customer's next purchase to credit risks and corporate earnings.

A predictive model is not fixed; it is validated or revised regularly to incorporate changes in the underlying data. In other words, it's not a one-and-done prediction. Predictive models make assumptions based on what has happened in the past and what is happening now. If incoming, new data shows changes in what is happening now, the impact on the likely future outcome must be recalculated, too.

Most predictive models work fast and often complete their calculations in real time. That's why banks and retailers can, for example, calculate the risk of an online mortgage or credit card application and accept or decline the request almost instantly based on that prediction.

Some predictive models are more complex, such as those used in computational biology and quantum computing; the resulting outputs take longer to compute than a credit card application but are done much more quickly than was possible in the past thanks to advances in technological capabilities, including computing power.

1.2. Problem Identification

Generally, in Educational Institutions, Mess management was unable to cook sufficient food as the number of people attending the mess becoming unpredictable. To overcome this problem, in some institutions Face Recognition system which generates paper tokens is implemented. Students should take the tokens by getting recognized by Face Recognition System and should avail it at the College Mess. This Machine learning model is an alternate solution for this problem.

1.3. Objectives

- To build a machine learning model with a good accuracy.
- To assist the model in prediction of number of people attending a college mess.
- To minimize the chances of misconceptions about the prediction of count.
- To reduce the risk that is faced by the face recognition system.
- To reduce the surplus and scarcity of food in mess.

1.4. Proposed Methodology

- The parameters which effect the number of people attending the mess is analyzed through survey and incorporated in the Data set.
- The Data related to food is collected from different college students through a survey via social media.
- To build this Machine Learning model, Support Vector Regression Algorithm is implemented using Python Programming language.
- Different techniques are used to improve the accuracy of the model.
- Statistical analysis is done and model is evaluated.

1.5. Scope

The number of people attending a hypothetical college mess prediction using Machine Leaning model helped out to be a great advantage in food care industry. With this model, number of people attending a mess can be predicted before preparing food which turned out to be the prediction of count of the people according which the food can be prepared which can reduce the scarcity and surplus of food.

1.6. Societal Applications

- This model can replace the Face Recognition system which is extravagant.
- The students' problem of awaiting for tokens can be resolved by using this model.
- Paper wastage is abstained by this model.
- Machine learning utilization opens door to futuristic technologies such as estimating the count of people attending a hypothetical mess driving information for outcomes research, planning and assisting food preparation for the management.

1.7. Limitations

- This model gives better accuracy for high range of input values.
- This model shows variance in the accuracy rate between the high range of input values and small range of input values.
- So, the model is better for the usage only when there is a high range of input values need to be considered.

CHAPTER - 2

LITERATURE REVIEW

Predicting the number of attending the mess is identified as a Regression problem. Machine Learning provides many regression algorithms for predictions. There are four types of machine learning algorithms: supervised, semi-supervised, unsupervised and reinforcement. Supervised Machine Learning model is considered for this project. A supervised learning algorithm takes a known set of input data (the learning set) and known responses to the data (the output), and forms a model to generate reasonable predictions for the response to the new input data. It is used, if you have existing data for the output you are trying to predict. Sources for this project are documentations and examples provided by scikit-learn.org, **'Hands-on Machine learning with Scikit Learn'** by **Aurelien Geron.**

"Machine learning and its applications: A review," 2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC) by **Heena Angra** and **Sachin Ahuja** suggests that Gradient descent is an iterative optimization algorithm for finding the local minimum of a function. To find the local minimum of a function using gradient descent, steps proportional to the negative of the gradient (move away from the gradient) of the function at the current point must be taken. If steps proportional to the positive of the gradient (moving towards the gradient) are taken, local maximum of the function will be approached, and the procedure is called **Gradient Ascent.** Gradient descent was originally proposed by **CAUCHY** in 1847. It is also known as steepest descent.

These Algorithms are trained on a historical data set and applied to new data, then algorithms generate probable values as a prediction. As the data set of the project is assumed to possess outliers, Support Vector Regression Algorithm is opted. Support Vector Regression (SVR) is a supervised learning method used for regression and outliers. Support Vector Regression is a supervised learning algorithm that is used to predict discrete values. Support Vector Regression uses the same principle as the SVMs. The basic idea behind SVR is to find the best fit line. In SVR, the best fit line is the hyperplane that has the maximum number of points**.**

'scikit-learn.org' is free software machine learning library for the Python programming Language. It provides many algorithms including SVR, and is designed to interoperate with Python numerical and scientific libraries like NumPy and SciPy. Scikit-learn is largely written in Python, and uses NumPy extensively for high-performance linear algebra and array operations. Furthermore, some core algorithms are written in Cython to improve performance. Support vector machines are implemented by a Cython wrapper around LIBSVM; logistic regression and linear support vector machines by a similar wrapper around LIBLINEAR. In such cases, extending these methods with Python may not be possible.

Scikit-learn integrates well with many other Python libraries, such as Matplotlib and plotly for plotting, NumPy for array vectorization, Pandas dataframes, SciPy, and many more.

Jupyter notebook is used as a main platform for this python programming in this prediction model. Jupyter Notebook (formerly IPython Notebooks) is a web-based interactive computational environment for creating notebook documents.

A Jupyter Notebook document is a browser-based REPL containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and rich media. Underneath the interface, a notebook is a JSON document, following a versioned schema, usually ending with the ". ipynb" extension. Jupyter notebooks are built upon a number of popular open-source libraries

'Sklearn' (Sci kit-learn) library is imported and SVR algorithm is implemented. Seaborn library is imported to create the data visualizations in this model. Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

Mat-plot-lib library is used to generate the required graphs for this prediction model. **Matplotlib** is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged. SciPy makes use of Matplotlib.

CHAPTER - 3

ANALYSIS OF METHODOLOGY

Machine Learning has two phases like Learning and Prediction. In the fist phase there will be preprocessing, learning and testing. In the second phase model is trained by dataset using some algorithm for predictions.

3.1 Dataset Creation

A **Hypothetical Dataset** is created based on some *constraints*. These constraints are found through a survey and they are:

> Females eat 'less' comparable to males

> On few days people attending the mess becomes 'more' due to reasons like

--> Food Menu

--> Attending early to the mess because of few events like NSS, Guest Lecture

> On few days people attending the mess becomes 'less' due to reasons like

--> Exam

--> Holidays

> For some reasons, generally in spite of all these people attending the college are relatively high in one day and relatively low on another in a week.

A dataset with the below components is created to meet the above mentioned constraints

Table 3.1 Tabular column with all of the components of dataset

S.NO	GENDER	DAY	EVENT_i (increment)	EVENT_d (decrement)	No event	RATING	TIME SLOT	NUMBER OF PEOPLE ATTENDED AT MESS

Gender - Male or Female.

Day - Monday or Tuesday or Wednesday or Thursday or Friday or Saturday.

Event_i accounts for the increment in the number of attending the mess.

Event_d accounts for the decrement in the number of people.

No_event accounts for a day in which no events responsible for the change in the number is happening.

Time_Slot - *12:30-1:30* has a total number of *2403* attending the mess, *1:30-2:30* has total number of *864*

Rating here, can be defined as *The Strength of the event*. It tells how strongly that event effects the Number of People

The rating value is given according to *influence of the event*. A **Standard Rating Table** is defined so that Rating is selected from it.

As it is a **Hypothetical Dataset,** following percentage of people attending, for following range of ratings is assumed

Table 3.2 Standard Rating Table

CASE	RATING (in %)	Number of People Attending (in %)
No event	12-15%	80-85%
Normal Decrement	17-23%	74-79%
Normal Increment	34-40%	86-91%
Special Increment	46-52%	92-97%
Special Decrement	72-78%	68-73%

Number is the target variable. As it is a hypothetical dataset instead of collecting the data, a relation between Rating and Number is developed. For a certain Rating there will be a unique Number percentage. The relation is established based on proportionality of the ranges of *Rating* and *Number*.

Let's explore a example to understand this

Let's find Percentage of the Number for the rating = 47

From the above table the rating 47 comes under Special Increment case, hence the Number percent should lie between 92-97%

As discussed before the ranges of *Rating* and *Number* should be proportional, that means that their ratio should never change hence

(Number percent - 92) ÷ (47-46) = 5 ÷ 6

Number percent = 92.8

Hence the general formula can be deduced as

(Number percent - First number in that CASE in the range of Number percent) ÷ (Given Rating - First number in that CASE in the range of Rating) = 5 ÷ 6

This is result as a Target Variable will be used to train the ML model.

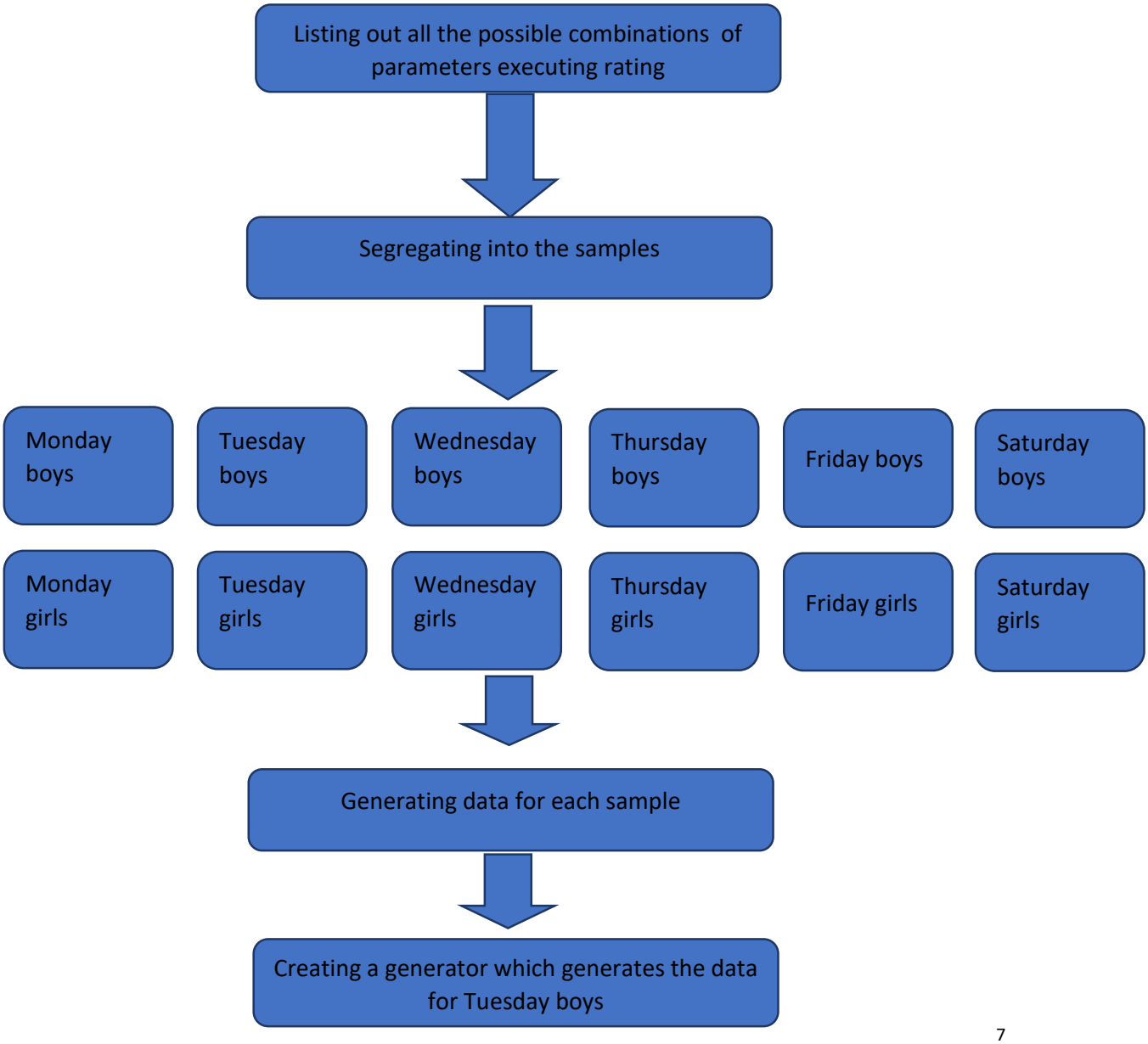
Our aim is to print "Comma Separated Values". Let's print all the possible outcomes excluding RATING and NUMBER

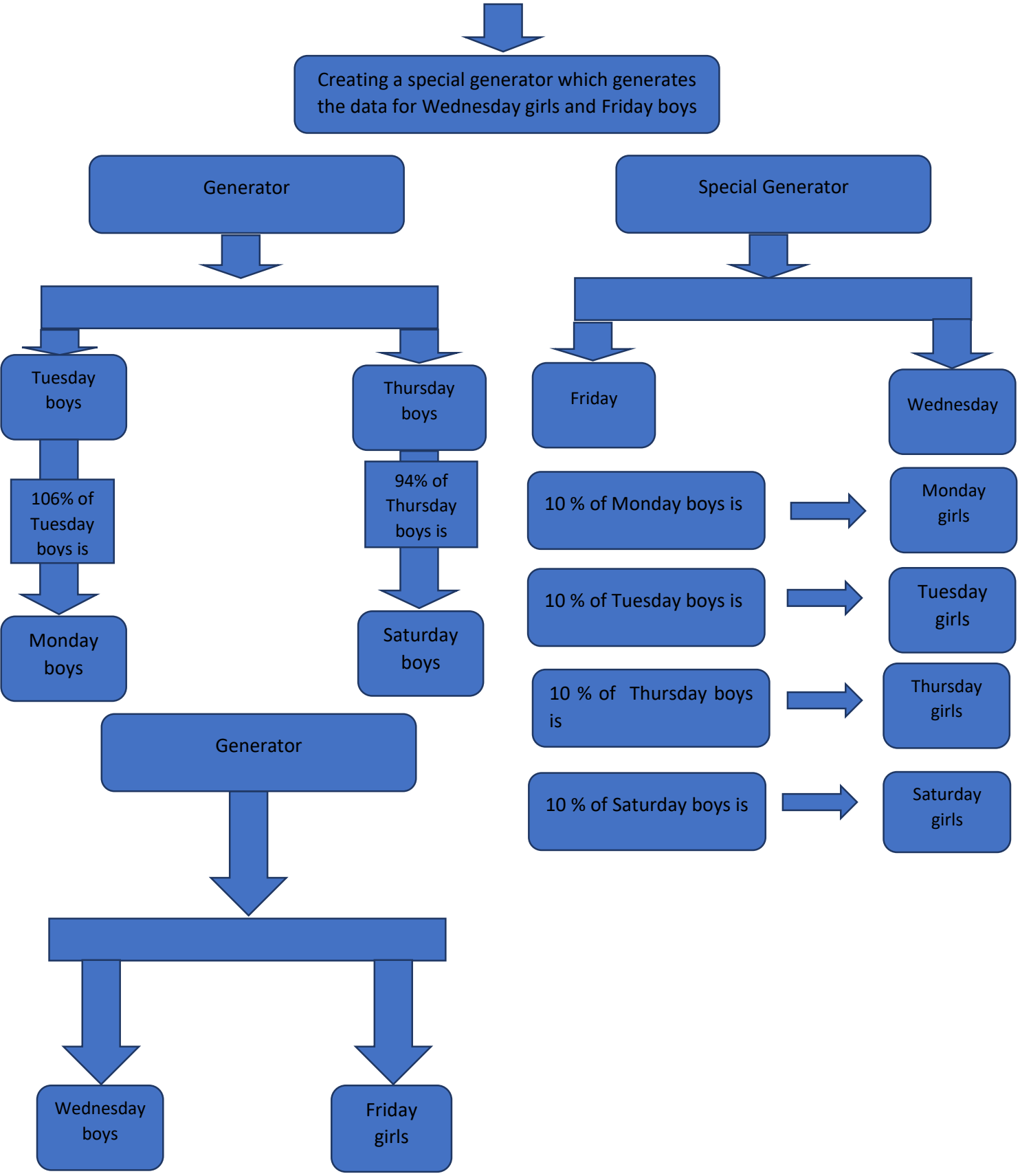
0,1 --> Female, Male

0,1,2,3,4,5 --> Monday, Tuesday, Wednesday, Thursday, Friday, Saturday

0,1 --> Event_i

0,1 --> Event_d




Creating a special generator which generates the data for Wednesday girls and Friday boys
Generator
Special Generator
Tuesday boys
Thursday boys
Friday
Wednesday
106% of Tuesday boys is
94% of Thursday boys is
10 % of Monday boys is
Monday girls
Monday boys
Saturday boys
10 % of Tuesday boys is
Tuesday girls
10 % of Thursday boys is
Thursday girls
10 % of Saturday boys is
Saturday girls
Generator
Wednesday boys
Friday girls

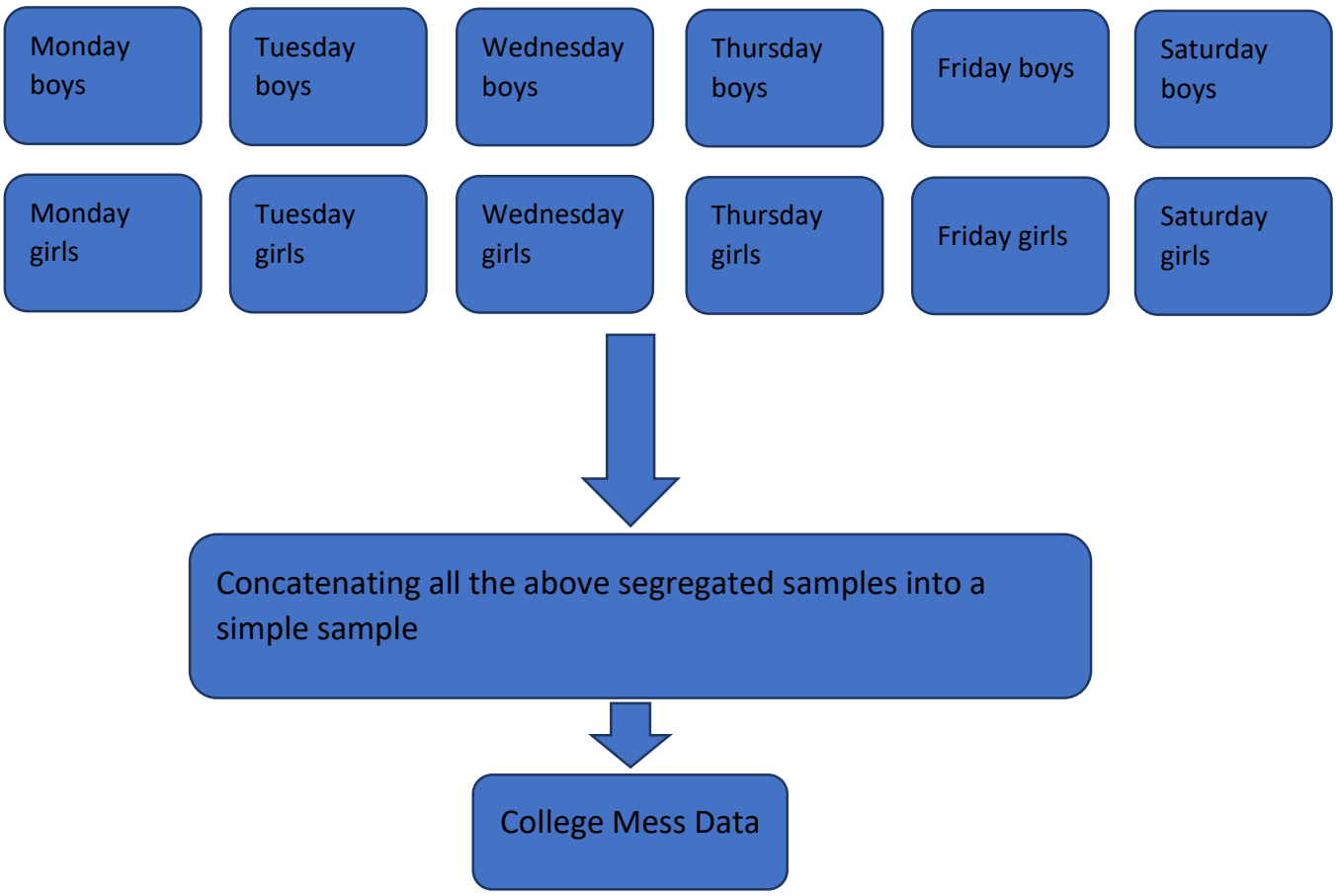


Figure 3.1 Pictorial Representation of Dataset Creation

3.2 Data Visualization

The dataset is analyzed and visualized using libraries like NumPy, matplotlib, seaborn. In this one can observe how the data is being distributed.

From the dataset following calculations are made:

- Average number of people attending the college for the "Increment case" in the Time Slot 12:30-1:30 for all days.
- Average number of people attending the college for the "Decrement case" in the Time Slot 12:30-1:30 for all days.
- Average number of people attending the college for the "No Event case" in the Time Slot 12:30-1:30 for all days.

In total 3 plots are made. From the above calculated data 2 plots and 1 using Seaborn Library.

- Plot of Average number of people attending the College Mess in each day of the week in the Time slot 12:30-1:30.
- Plot of Average number of people attending the College Mess in each day of the week in the Time slot 1:30-2:30.
- Plot of number of Boys, Girls attending on each day in a week.

3.3 Implementation of SVR

To implement this Machine Learning Model, Support Vector Regression Algorithm is used. **Support Vector Machines (SVM)** are one of the state-of-the-art machine learning algorithms based on **Maximal Margin Classifier**. SVM support linear as well as non-linear regression called **Support Vector Regression (SVR).**

Linear Regression : Support vector regression : : Logistic regression : Support vector Machine.

The difference dwells on the input feature space dimensions.
A linear regression models a line to classify the data points, while a support vector regression can also model a hyperplane.

Hyperplane

Hyperplane acts as a classifier that splits the input feature space. For a two-variable space, the hyperplane would be a line given below,

$$b0+b1*x1+b2*x2=0$$

Figure 3.2 Two variable Space Hyperplane Equation

In case of SVM,

- A point above the line gets a label `1`
- A point below the line gets a label `0`
- A point close to the line return a value close to `0` and the point may be difficult to classify.
- If the magnitude of value is large for a test data point, the model will have more confidence in prediction.

Thus, comes the concept of maximizing the margin.

Margin

The points closest to the hyperplane has an uncertain class label or has a class label with nearly equal probability as the other class label. Margin is the perpendicular distance between the hyperplane and the closest points.

Therefore, SVM tries to maximize this margin and works as ‘Maximal Margin Classifier’

Support Vectors

The training instances closest to the hyperplane that help define the margin are called Support Vectors.

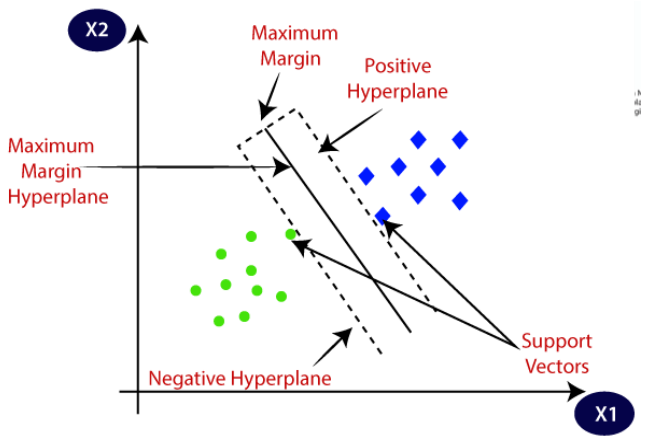


Figure 3.3 Intuition of SVR

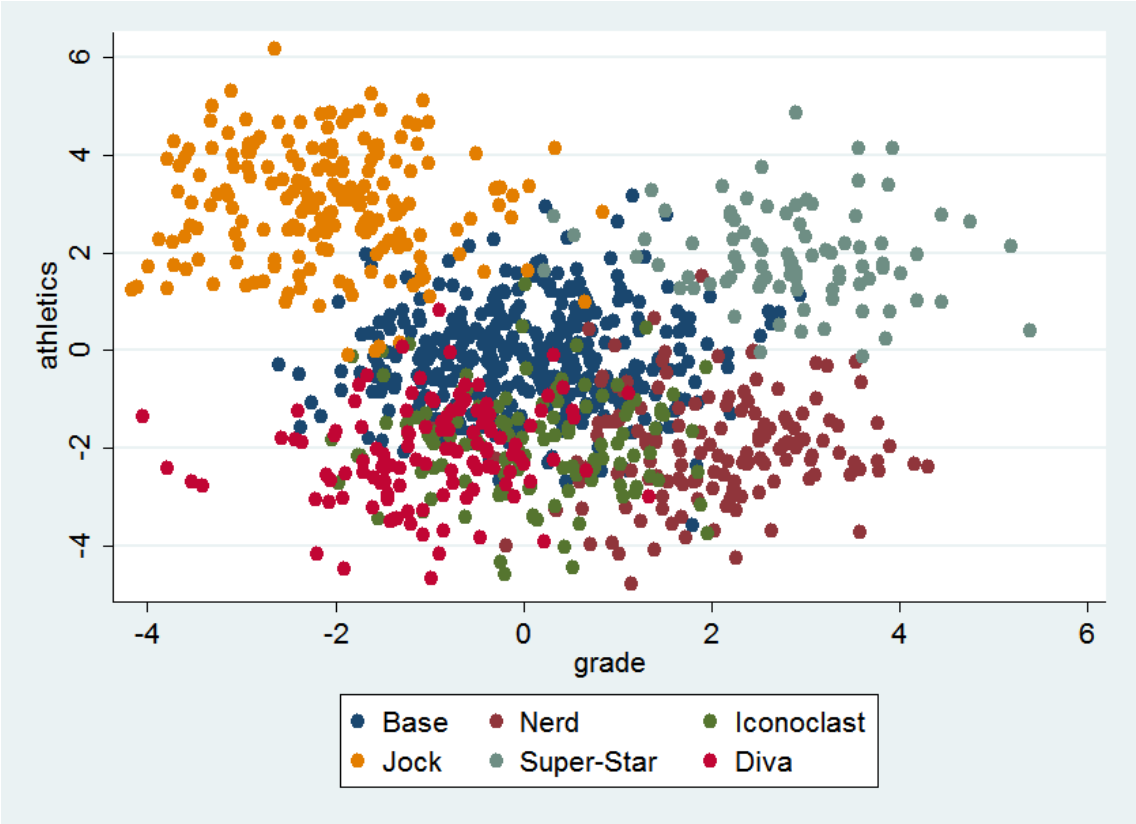


Figure 3.4 Sample Data

Real data is messy!

Thus, there won't be a perfect hyperplane splitting the data points perfectly into two classes. Constraint of maximizing the margin are tend to be relaxed and some points to violate the margin are allowed (that is, allow some points to have wrong label as per our model for a good overall performance).

Now that being said, SVR works a little different. SVR tries to have as many support vectors as possible within the boundary lines without much margin violation, thus keeping the error within the threshold decided by the boundary lines.

Intuitively, these support vectors contribute to the error made by the SVR and thusmmost of the support vectors are desired to be in that threshold. SVM as well as SVR through **kernels** that indicates the similarity measure between the test data point and the support vectors can be modelled. The following methodology is followed to implement the SVR.

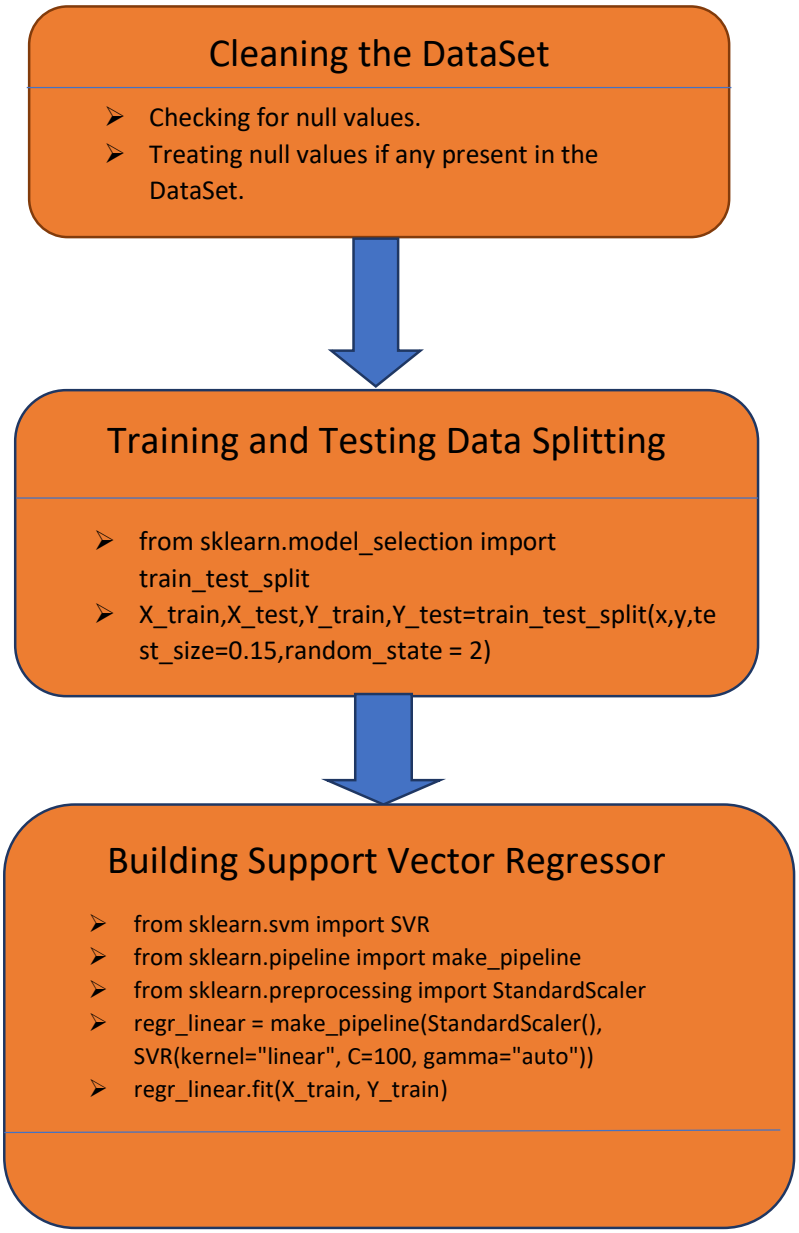


Figure 3.5 Pictorial Representation of SVR building

3.4 Model Evaluation and Visualization

Data which is split as Testing Data is now fed to Machine Learning Model to get the Predictions. These predictions are compared with the Actual Data Values.

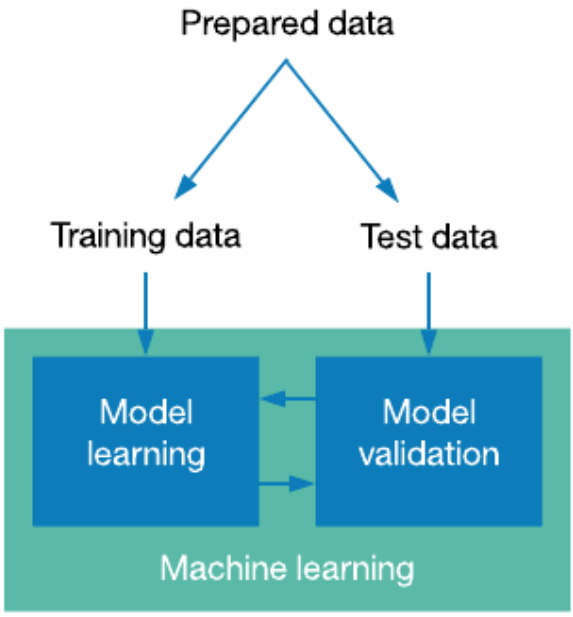


Figure 3.6 Cross Validation

The following evaluation metrics are used:

- Mean Squared Error.
- Root Mean Squared Error.
- R2 Score.

Model's result is visualized through the plots like:

- Actual vs Predicted values plot.
- Regressor line plot of SVR.

CHAPTER - 4

DATASET CREATION AND ANALYSIS

Dataset is the important part in supervised machine learning process. A well cleansed dataset and a appropriate algorithm for the dataset provide good results.

4.1 Creating Generator

Generator prints the Comma Separated Values for general cases like Tuesday boys, Thursday boys. In the general case there will be all the possibilities of increments, decrements, no event cases. It is created using Python Programming Language

4.1.1 Generator program code

```
# This is for generating Tuesday boys

import random as rd # This is used to generate the Ratings randomly

for i in range(6):

# This is for generating increment case in the Time Slot 12:30-1:30
    if i==0:
        # For Normal Increment
        for k in range(10):
            print(1,1,sep=',',end=',')
            a=rd.uniform(34,40)
            b=((86+(a-34)*(5/6))/100)*2403
            for j in range(4):
                if j!=3:
                    print(arr[i,j],sep=',',end=',')
                else:
                    print(arr[i,j],int(a),int(b),sep="","")
                    print('\n')
    # This is for generating Decrement case in the Time Slot 12:30-1:30
    elif i==1:
        # For Normal Decrement
        for k in range(10):
            print(1,1,sep=',',end=',')
```

```
        a=rd.uniform(17,23)
        b=((74+(a-17)*(5/6))/100)*2403
        for j in range(4):
            if j!=3:
                print(arr[i,j],sep=',',end=',')
            else:
                print(arr[i,j],int(a),int(b),sep='',')
                print('"\n"')
    # For Special Decrement
    for k in range(10):
        print(1,1,sep=',',end=',')
        a=rd.uniform(72,78)
        b=((68+(a-72)*(5/6))/100)*2403
        for j in range(4):
            if j!=3:
                print(arr[i,j],sep=',',end=',')
            else:
                print(arr[i,j],int(a),int(b),sep='',')
                print('\n')
# This is for generating No event case in the Time Slot 12:30-1:30
elif i==2:
    #For No event
    for k in range(10):
        print(1,1,sep=',',end=',')
        a=rd.uniform(12,15)
        b=((80+(a-12)*(5/3))/100)*2403
        for j in range(4):
            if j!=3:
                print(arr[i,j],sep=',',end=',')
            else:
```

```
            print(arr[i,j],int(a),int(b),sep=",")
            print('\n')
# This is for generating increment case in the Time Slot 1:30-2:30
elif i==3:
    # For Normal Increment
    for k in range(10):
        print(1,1,sep=',',end=',')
        a=rd.uniform(34,40)
        b=((86+(a-34)*(5/6))/100)*864
        for j in range(4):
            if j!=3:
                print(arr[i,j],sep=',',end=',')
            else:
                print(arr[i,j],int(a),int(b),sep=",")
                print('\n')
# This is for generating Decrement case in the Time Slot 1:30_2:30
elif i==4:
    # For Normal Decrement
    for k in range(10):
        print(1,1,sep=',',end=',')
        a=rd.uniform(17,23)
        b=((74+(a-17)*(5/6))/100)*864
        for j in range(4):
            if j!=3:
                print(arr[i,j],sep=',',end=',')
            else:
                print(arr[i,j],int(a),int(b),sep=",")
                print("\n")
    # For Special Decrement
    for k in range(10):
```

```
        print(1,1,sep=',',end=',')
        a=rd.uniform(72,78)
        b=((68+(a-72)*(5/6))/100)*864
        for j in range(4):
            if j!=3:
                print(arr[i,j],sep=',',end=',')
            else:
                print(arr[i,j],int(a),int(b),sep=",")
                print('\n')
    # This is for generating No event case in the Time Slot 1:30-2:30
    elif i==5:
        #For No event
        for k in range(10):
            print(1,1,sep=',',end=',')
            a=rd.uniform(12,15)
            b=((80+(a-12)*(5/3))/100)*864
            for j in range(4):
                if j!=3:
                    print(arr[i,j],sep=',',end=',')
                else:
                    print(arr[i,j],int(a),int(b),sep=",")
                    print("\n")
```

4.2 Creating Special Case Generator

Special Case Generator Print Comma Separated values for Special cases like Wednesday Girls, Friday Boys in which the number of people attending is high. It is also written in Python programming language.

4.2.1 Special Case Generator program code

```
for i in range(4):
    # For Wednesday girls in the both Time Slots
```

```
if i==0 or i==2:
    for k in range(50):
        a=rd.uniform(46,52)
        b=((92+(a-46)*(5/3))/100)*2403
        for j in range(6):
            if j!=5:
                print(arr1[i,j],sep=',',end=',')
            else:
                print(arr1[i,j],int(a),int(b),sep=",")
                print("\n")
# For Friday in the both Time Slots
else:
    for k in range(50):
        a=rd.uniform(46,52)
        b=((92+(a-46)*(5/3))/100)*864
        for j in range(6):
            if j!=5:
                print(arr1[i,j],sep=',',end=',')
            else:
                print(arr1[i,j],int(a),int(b),sep=",")
                print("\n")
```

4.3 Creating the College Mess Data

College Mess Data is created by first segregating into individual samples and then created the values of each individual sample by using the above generators. (As mentioned in Section 3.1)

Finally integrating all the samples into a single sample. This is done using Python DataFrames. Values from Segregated Individual samples are copies onto notepad and saved with '.csv' extension. The csv files converted into DataFrames of python and required operations are done there. All the equivalent values are converted from numericals to characters (For e.g., '0' to No; '1' to Yes). Finally, all the Segregated DataFrames are concatenated into single DataFrame and this DataFrame is converted into .csv file.

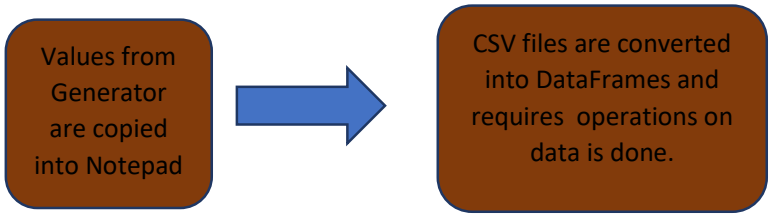


Figure 4.1 Operations on DataSet.

4.4 Data Visualization

The Data is visualized in such a way that how the data is being distributed is analyzed. It is done using basic iterative statement like for loop, conditional statement like if-else suite. To do this all the required average values (already mentioned in the Section 3.2) are found out and required graphs (mentioned in Section 3.2) are plotted.

4.4 Data Visualization plots

```
Out[32]: <function matplotlib.pyplot.show(close=None, block=None)>
```

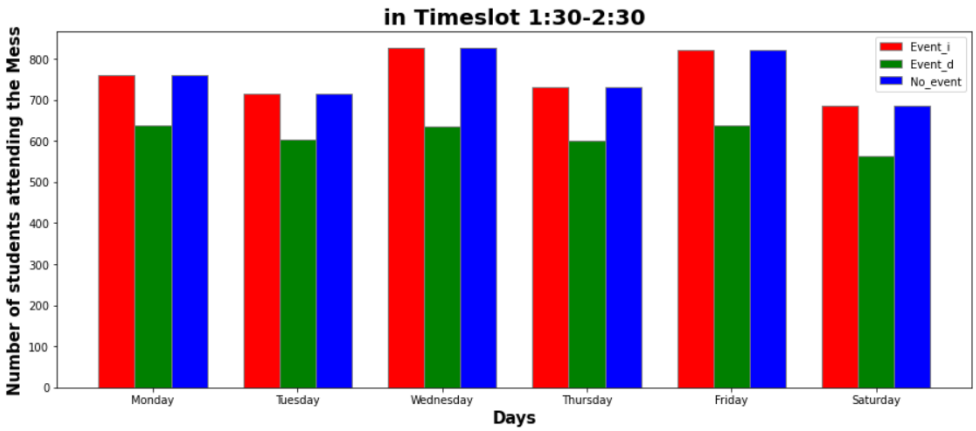


Figure 4.2 Plot of Average number of people attending the College Mess in each day of the week in the Time slot 1:30-2:30.

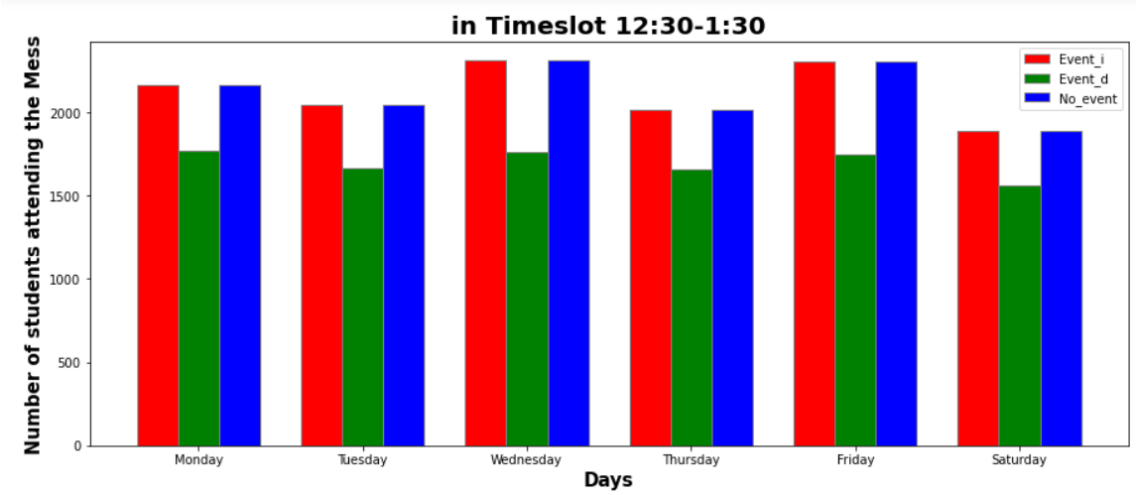


Figure 4.3 Plot of Average number of people attending the College Mess in each day of the week in the Time slot 12:30-1:30.

```
In [34]: import seaborn as sns  # Seaborn is used widely used library for dat Visualization

         plt.figure(figsize=(20,5))
         sns.barplot(x='Day',y='Number',hue='Gender',data=final_df)

Out[34]: <AxesSubplot:xlabel='Day', ylabel='Number'>
```

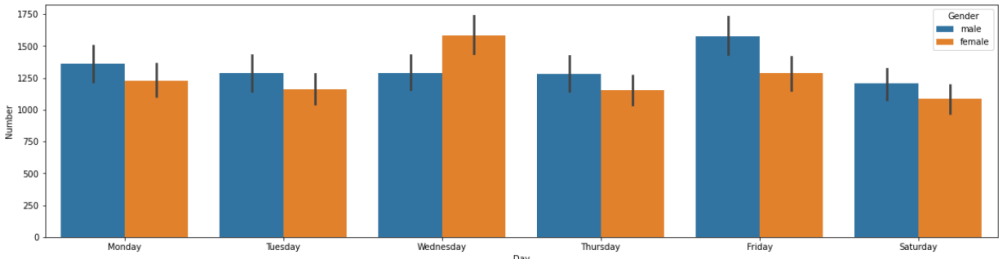


Figure 4.4 Plot of number of Boys, Girls attending on each day in a week.

CHAPTER - 5

SVR BUILDING AND ANALYSIS

SVR Machine Learning Model is built from the following steps. (As mentioned in the Section 3.3)

Step 1: Importing Dependencies and Insurance DataSet

Importing Dependencies

```
In [2]: '''
        The Required basic dependencies are imported for general operations

        '''

        import pandas as pd    # Pandas library is imported,so that Pandas DataFrame can be used

        import numpy as np     # Numpy is library used for Mathematical Operations

        import matplotlib.pyplot as plt   # Matplotlib Dependency is used  visualize the required plots
        %matplotlib inline
```

Importing Insurance DataSet

```
In [3]: df=pd.read_csv('College_Mess_Data.csv')  # To read the csv file

        df.head() # To print frist 5 Datapoints
```

Out[3]:

	Unnamed: 0	Gender	Day	Event_i	Event_d	No_event	Time_Slot	Rating	Number
0	0	male	Monday	yes	no	no	12:30-1:30	34	2209
1	1	male	Monday	yes	no	no	12:30-1:30	39	2309
2	2	male	Monday	yes	no	no	12:30-1:30	36	2252
3	3	male	Monday	yes	no	no	12:30-1:30	38	2283
4	4	male	Monday	yes	no	no	12:30-1:30	39	2302

Step 2: Deleting the column which is not required

Deleting the column which is not required

```
In [4]: df.pop('Unnamed: 0')
        df
```

Out[4]:

	Gender	Day	Event_i	Event_d	No_event	Time_Slot	Rating	Number
0	male	Monday	yes	no	no	12:30-1:30	34	2209
1	male	Monday	yes	no	no	12:30-1:30	39	2309
2	male	Monday	yes	no	no	12:30-1:30	36	2252
3	male	Monday	yes	no	no	12:30-1:30	38	2283
4	male	Monday	yes	no	no	12:30-1:30	39	2302
...	...	...	...	...	...	...	...	...
983	female	Saturday	no	no	yes	1:30-2:30	12	588
984	female	Saturday	no	no	yes	1:30-2:30	14	613
985	female	Saturday	no	no	yes	1:30-2:30	12	589
986	female	Saturday	no	no	yes	1:30-2:30	12	586
987	female	Saturday	no	no	yes	1:30-2:30	13	595

988 rows × 8 columns

Step 3 : Checking for Null values and getting Statistical Data

Checking for null values

In [5]:
```
df.isnull().sum() # found no null values bacuse it is a created DataSet
```

Out[5]:
```
Gender       0
Day          0
Event_i      0
Event_d      0
No_event     0
Time_Slot    0
Rating       0
Number       0
dtype: int64
```

Getting the Statistical Data

In [6]:
```
df.describe()
```

Out[6]:

	Rating	Number
count	988.000000	988.000000
mean	38.283401	1302.768219
std	22.077891	651.324695
min	12.000000	496.000000
25%	18.000000	675.750000
50%	36.000000	871.500000
75%	50.000000	1886.500000
max	77.000000	2448.000000

Step 4: Label Encoding

Label Encoding

In [7]:
```
# As machine deals with Numericals Characerters are converted into Integers

from sklearn import preprocessing

label_encoder=preprocessing.LabelEncoder()


df['Gender']=label_encoder.fit_transform(df['Gender'])
df['Day']=label_encoder.fit_transform(df['Day'])
df['Event_i']=label_encoder.fit_transform(df['Event_i'])
df['Event_d']=label_encoder.fit_transform(df['Event_d'])
df['No_event']=label_encoder.fit_transform(df['No_event'])
df['Time_Slot']=label_encoder.fit_transform(df['Time_Slot'])

df['Gender'].unique()
df['Day'].unique()
df['Event_i'].unique()
df['Event_d'].unique()
df['No_event'].unique()
df['Time_Slot'].unique()
```

Out[7]:
```
array([0, 1])
```

Step 5: Train and Test Splitting

Train Test Splitting

In [9]:
```
from sklearn.model_selection import train_test_split

x=df.drop(columns='Number',axis=1)  # Target Variable Column deleted Dataframe

y=df['Number']   # Target Variable Dataframe

X_train,X_test,Y_train,Y_test=train_test_split(x,y,test_size=0.15,random_state = 2)
```

Step 6: Building SVR using Linear Function

Building Suport Vector Regressor using *linear function*

In [10]:
```
from sklearn.svm import SVR                          # Importing Support Vector Regressor
from sklearn.pipeline import make_pipeline           # To create A pipeline
from sklearn.preprocessing import StandardScaler     # Used for the Strandardization

regr_linear = make_pipeline(StandardScaler(), SVR(kernel="linear", C=100, gamma="auto"))

# Training the Data to ML model
regr_linear.fit(X_train, Y_train)
```

Out[10]:
```
Pipeline(steps=[('standardscaler', StandardScaler()),
                ('svr', SVR(C=100, gamma='auto', kernel='linear'))])
```

In [11]:
```
# Getting the predictions
Y_pred_linear=regr_linear.predict(X_test)
```

Step 7: Building SVR using Robust Function

Building Suport Vector Regressor using *Robust function*

In [13]:
```
regr_RBF = make_pipeline(StandardScaler(),SVR(kernel="rbf", C=100, gamma=0.1, epsilon=0.1))

# Training the Data to ML model
regr_RBF.fit(X_train, Y_train)
```

Out[13]:
```
Pipeline(steps=[('standardscaler', StandardScaler()),
                ('svr', SVR(C=100, gamma=0.1))])
```

In [14]:
```
# Getting the Predictions

Y_pred_RBF=regr_RBF.predict(X_test)
```

Robust function for predictions is considered, after model evaluation it is found that it gives better prediction.

CHAPTER - 6

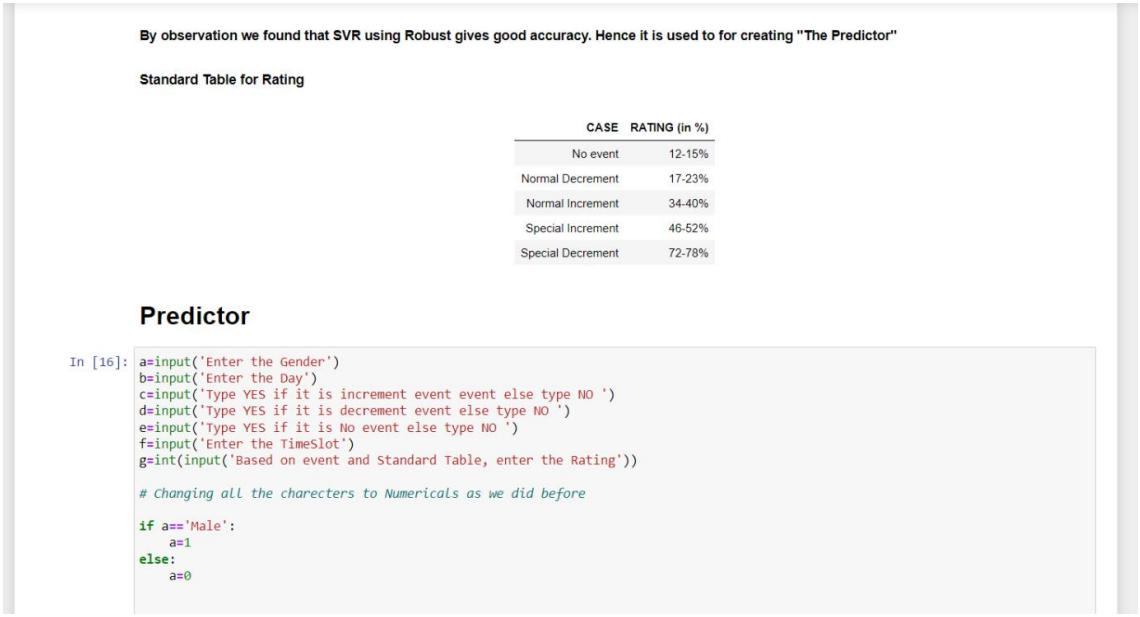
RESULTS AND MODEL EVALUATION

Model Evaluation is the key part in the machine learning. It tells about the performance of the model. Results from both the model(i.e, Linear and Robust models) are compared and better is model is judged.

6.1 The Predictor

The Predictor which actually predicts the Number of people attending based on the given inputs is created.

Figure 6.1 The Predictor



By observation we found that SVR using Robust gives good accuracy. Hence it is used to for creating "The Predictor"

Standard Table for Rating

CASE	RATING (in %)
No event	12-15%
Normal Decrement	17-23%
Normal Increment	34-40%
Special Increment	46-52%
Special Decrement	72-78%

Predictor

In [16]:
```
a=input('Enter the Gender')
b=input('Enter the Day')
c=input('Type YES if it is increment event event else type NO ')
d=input('Type YES if it is decrement event else type NO ')
e=input('Type YES if it is No event else type NO ')
f=input('Enter the TimeSlot')
g=int(input('Based on event and Standard Table, enter the Rating'))

# Changing all the charecters to Numericals as we did before

if a=='Male':
    a=1
else:
    a=0
```

```
if b=='Monday':
    b=0
elif b=='Tuesday':
    b=1
elif b=='Wednesday':
    b=2
elif b=='Thursday':
    b=3
elif b=='Friday':
    b=4
elif b=='Saturday':
    b=5

if c=='YES':
    c=1
else:
    c=0


if d=='YES':
    d=1
else:
    d=0

if e=='YES':
    e=1
else:
    e=0


if f=='12:30-1:30':
    f=0
else:
    f=1
```

```
if f=='12:30-1:30':
    f=0
else:
    f=1



arr=regr_RBF.predict([[a,b,c,d,e,f,g]])
print('Number of people attending is',int(arr[0]))
```

```
Enter the GenderFemale
Enter the DayWednesday
Type YES if it is increment event event else type NO YES
Type YES if it is decrement event else type NO NO
Type YES if it is No event else type NO NO
Enter the TimeSlot12:30-1:30
Based on event and Standard Table, enter the Rating51
Number of people attending is 2155
```

6.2 Model Evaluation

Two SVR models (one using Linear function, other using Robust function) are created. Both the models are evaluated.

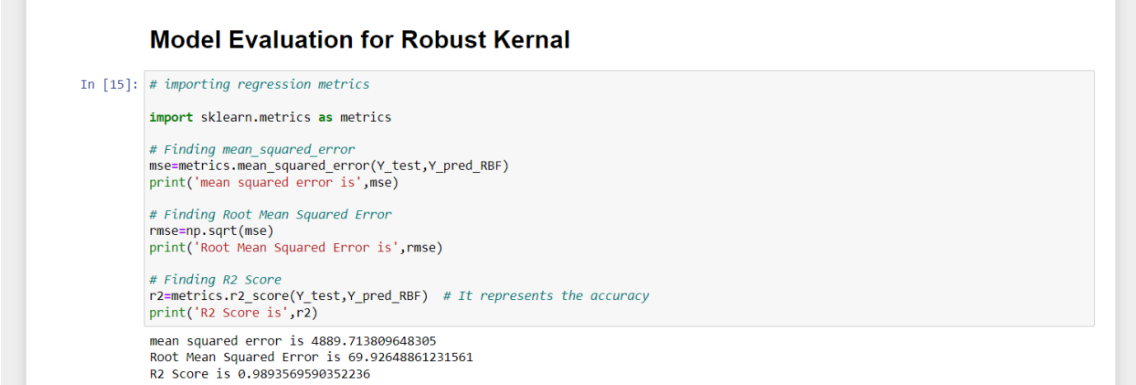
Figure 6.2 Model Evaluation of Linear Kernel

Model Evaluation for Linear Kernal

In [12]:
```
# importing regression metrics

import sklearn.metrics as metrics

# Finding mean_squared_error
mse=metrics.mean_squared_error(Y_test,Y_pred_linear)
print('mean squared error is',mse)

# Finding Root Mean Squared Error
rmse=np.sqrt(mse)
print('Root Mean Squared Error is',rmse)

# Finding R2 Score
r2=metrics.r2_score(Y_test,Y_pred_linear)  # It represents the accuracy
print('R2 Score is',r2)
```

```
mean squared error is 21475.208347471307
Root Mean Squared Error is 146.54421976820277
R2 Score is 0.953256666736149
```

Figure 6.3 Model Evaluation for Robust function



From the Figure 6.2, Figure 6.3 it can be said that Robust Function has good performance.

Table 6.1 Model Evaluation

Evaluation Metrics	Linear Function	Robust Function
Mean Squared Error	21475.208	4889.71
Root Mean Squared Error	146.5	69.92
R2 Score	0.9532	0.989

From the above table it can be said that Robust Function has better desirable evaluation metrics.

6.3 Visualization of results

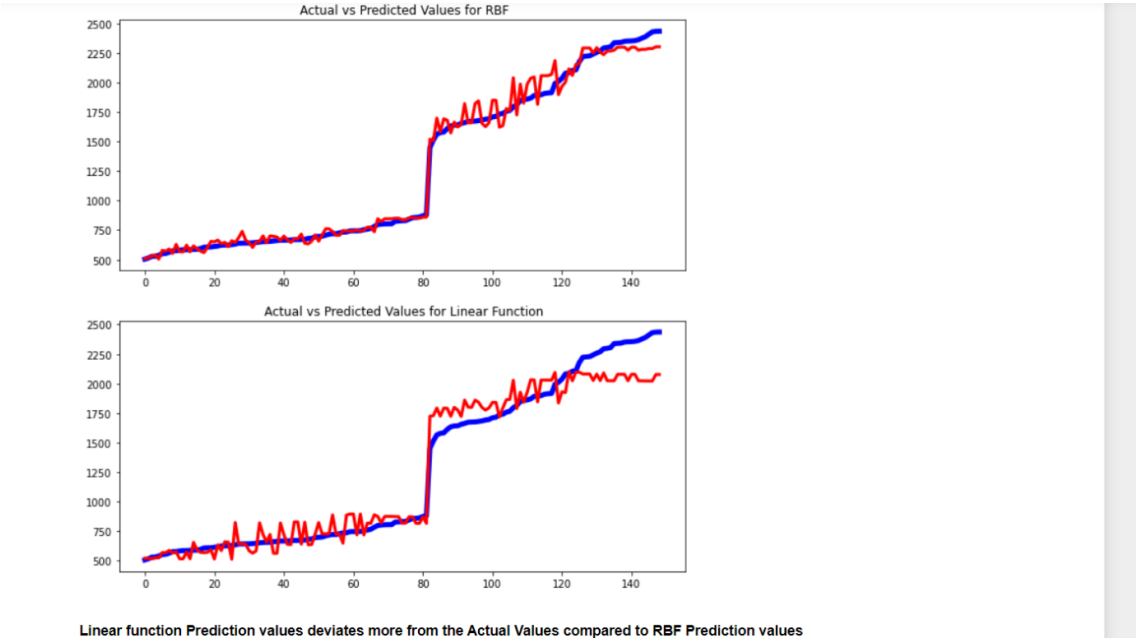
The results are visualized through

- Actual vs Predicted values plot.
- Regressor line plots of SVR.

6.3.1 Regressor Plots of SVR

Regressor Plots of SVR using Linear, Robust Functions are plotted here. Regressor plot visualizes the relationship between their parameters. The blue line in Figure 6.4 represents the actual data and the red line is the regressor line plot which is plotted based on the predictions from the model.

Figure 6.4 Regressor Plots of SVR

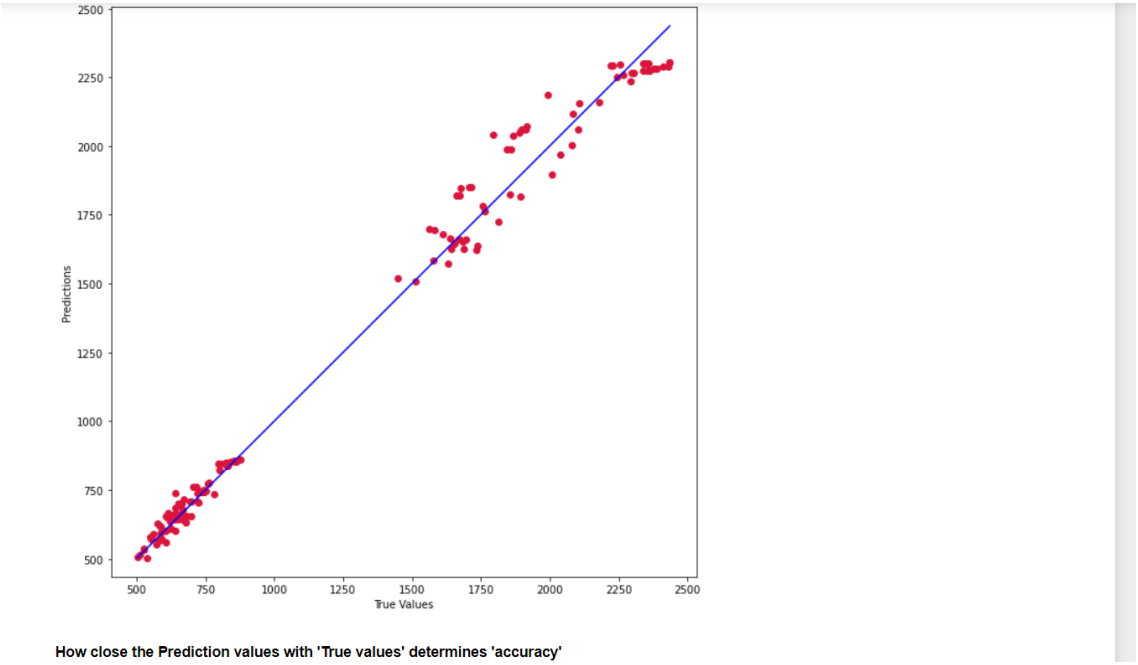


Linear function Prediction values deviates more from the Actual Values compared to RBF Prediction values

6.3.2 Actual vs Predicted values plot

This plot shows us the difference between Actual values and Predicted values. The red dots represent the Actual Data and the blue line represents the predicted values. The accuracy is based on how close the red dots and blue line is.

Figure 6.5 Actual vs Predicted values



How close the Prediction values with 'True values' determines 'accuracy'

CONCLUSION

The machine learning model that predicts the number of people attending a college mess can overcome the problem of awaiting for tokens. Jupyter Notebook as IDE which works better for large data. Libraries and dependencies for different purposes such as pandas, numpy, matplotlib, seaborn etc are used.

Scikit-learn is the most used library for machine learning in python. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.
This machine learning based model could provide the very efficient and fast proxy for complex numbers.

In the decision framework of decision analysis, in this study, the value of machine learning in a particular decision context was analyzed. This analysis I.e., considering the final result of the data analytics as the information for decision will be helpful for the college mess management to avoid the scarcity and surplus of food.

There is an essential need in the domain of the educational institutions to build a fast model for optimization where the current face recognition system is leading to paper wastage. The robust optimization framework proposed in this work has a clear application: gives the prior count of people attending the mess faster to help decision maker make a decision in a timely manner and the paper wastage is abstained. This offers a practical new alternative to face recognition system.

REFERENCES

- "Machine Learning." *Wikipedia, (n.d)* https://en.m.wikipedia.org/wiki/machine_learning
- "Support-vector machine." *Wikipedia, (n.d)* https://en.m.Wikipedia.org/wiki/Support-vector_machine
- "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow", book by Aurelien Geron.
- Scikit-Learn Library, https://scikit-learn.org/stable/
- Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.
- P. Louridas and C. Ebert, "Machine Learning," in IEEE Software, vol. 33, no. 5, pp. 110-115, Sept.-Oct. 2016, doi: 10.1109/MS.2016.114
- H. Yu, J. Lu and G. Zhang, "An Online Robust Support Vector Regression for Data Streams," in IEEE Transactions on Knowledge and Data Engineering, doi: 10.1109/TKDE.2020.2979967.
- S. Ray, "A Quick Review of Machine Learning Algorithms," 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), 2019, pp. 35-39, doi: 10.1109/COMITCon.2019.8862451.
- S. Angra and S. Ahuja, "Machine learning and its applications: A review," 2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC), 2017, pp. 57-60, doi: 10.1109/ICBDACI.2017.8070809.
- D. Gündüz, P. de Kerret, N. D. Sidiropoulos, D. Gesbert, C. R. Murthy and M. van der Schaar, "Machine Learning in the Air," in IEEE Journal on Selected Areas in Communications, vol. 37, no. 10, pp. 2184-2199, Oct. 2019, doi: 10.1109/JSAC.2019.2933969.