PROJECT REPORT
Car Resale value Prediction

1. INTRODUCTION
1.1 Project Overview
1.2 Purpose
2. LITERATURE SURVEY
2.1 Existing problem
2.2 References
2.3 Problem Statement Definition
3. IDEATION & PROPOSED SOLUTION
3.1 Empathy Map Canvas
3.2 Ideation & Brainstorming
3.3 Proposed Solution
3.4 Problem Solution fit
4. REQUIREMENT ANALYSIS
4.1 Functional requirement
4.2 Non-Functional requirements
5. PROJECT DESIGN
5.1 Data Flow Diagrams
5.2 Solution & Technical Architecture
5.3 User Stories
6. PROJECT PLANNING & SCHEDULING
6.1 Sprint Planning & Estimation
6.2 Sprint Delivery Schedule
6.3 Reports from JIRA
7. CODING & SOLUTIONING
7.1 Feature 1
7.2 Feature 2
7.3 Database Schema (if Applicable)
8. TESTING
8.1 Test Cases
8.2 User Acceptance Testing
9. RESULTS
9.1 Performance Metrics
10. ADVANTAGES & DISADVANTAGES
11. CONCLUSION
12. FUTURE SCOPE
13. APPENDIX
13.1 Source Code
GitHub & Project Demo Link

INTRODUCTION

CHAPTER-1
INTRODUCTION

1.1 Project Overview

With difficult economic conditions, it is likely that sales of second-hand imported (reconditioned) cars and used cars will increase. In many developed countries, it is common to lease a car rather than buying it outright. After the lease period is over, the buyer has the possibility to buy the car at its residual value, i.e. its expected resale value. Thus, it is of commercial interest to sellers/financers to be able to predict the salvage value (residual value) of cars with accuracy.

In order to predict the resale value of the car, we proposed an intelligent, flexible, and effective system that is based on using regression algorithms. Considering the main factors which would affect the resale value of a vehicle a regression model is to be built that would give the nearest resale value of the vehicle. We will be using various regression algorithms and algorithm with the best accuracy will be taken as a solution, then it will be integrated to the web-based application where the user is notified with the status of his product.

1.2 Purpose

Considering the anomalies in the existing system computerization of the whole activity is being suggested after initial analysis. It might have happened so many times that you or someone yours need doctors help immediately, but they are not available due to some reason. The Heart Disease Prediction application is an end user support and online consultation project. Here, we propose a web application that allows users to get instant guidance on their heart disease through an intelligent system online.

Various details are fed in the application and the heart disease associated with those details. Users can share their heart related issues with this application. It then processes user specific details to check for various illness that could be associated with it. After getting the result from the system, system suggests various doctors for treatment. The system allows user to view doctor’s details. The system can be use in case of emergency.

LITERATURE SURVEY

CHAPTER-2

LITERATURE SURVEY

2.1 Existing problem

Problem statement Machine learning has become a tool used in almost every task that requires estimation. Companies like cars24 and cardekho. Com uses Regression analysis to estimate the used car prices. So we need to build a model to estimate the price of cars. The model should take car-related parameters and output a selling price. The selling price of a used car depends on certain features as mentioned below

- Fuel Type
- Manufacturing year
- Miles Driven
- Number of Historical Owners
- Maintenance Record

This is a supervised learning problem and can be solved using regression techniques. We need to predict the selling price of a car based on the given car's features. Supervised Regression problems require labeled data where our target or dependent variable is the selling price of a car. All other features are independent variables.

Following are some regression algorithms that can be used for predicting the selling price.

- Linear Regression
- Decision Tree Regressor
- Support Vector Regressor
- KNN Regressor
- Random Forest Regressor

Linear Models are relatively less complex and explainable, but linear models perform poorly on data containing the outliers. Linear models fail to perform well on non-linear datasets. In such cases, nonlinear regression algorithms Random Forest Regressor and XGBoost Regressor perform better in fitting the nonlinear data. we will use Random Forest Regressor for predicting the selling price of cars.

2.2 References

TITLE **:** Tu Weixing. Research on Used Car Evaluation System[J]
AUTHOR **:** Nanjing: Nanjing Forestry University
YEAR **:** 2008

In order to meet the needs of second-hand car value assessment, the used car value assessment system has been designed based on the improved replacement cost method. The system includes system management module, used car parameter management module, used car evaluation management module and evaluation information inquiry module. We enter the relevant basic information of second-hand car information, and figure out the used car's new rate, the purchase price and the selling price through the calculation to the system.

TITLE : Determinants of used vehicle resale value.
AUTHOR : Richardson, M. S
YEAR : 2009

In his theory it states more durable vehicles will be produced by vehicle producer. He compared the hybrid vehicles and traditional vehicles in hoe it actually retains their value for longer time using multiple regression techniques. This improves the environmental conditions, and also it helps to provide huge efficiency of using fuels.

TITLE : Used Cars Price Prediction
AUTHOR : Pattabiraman Venkatasubbu et al
YEAR : 2007

This paper is more concentrated on the relation between seller and buyer. In order to predict the price of four wheelers, more features are required such as already given price, mileage, make, model, trim, type, cylinder, liter, doors, cruise, sound, leather. Using these features the price of vehicle has been predicted with the help of statistical analysis system for exploratory data analysis

IDEATION & PROPOSED SOLUTION

CHAPTER-3

IDEATION & PROPOSED SOLUTION

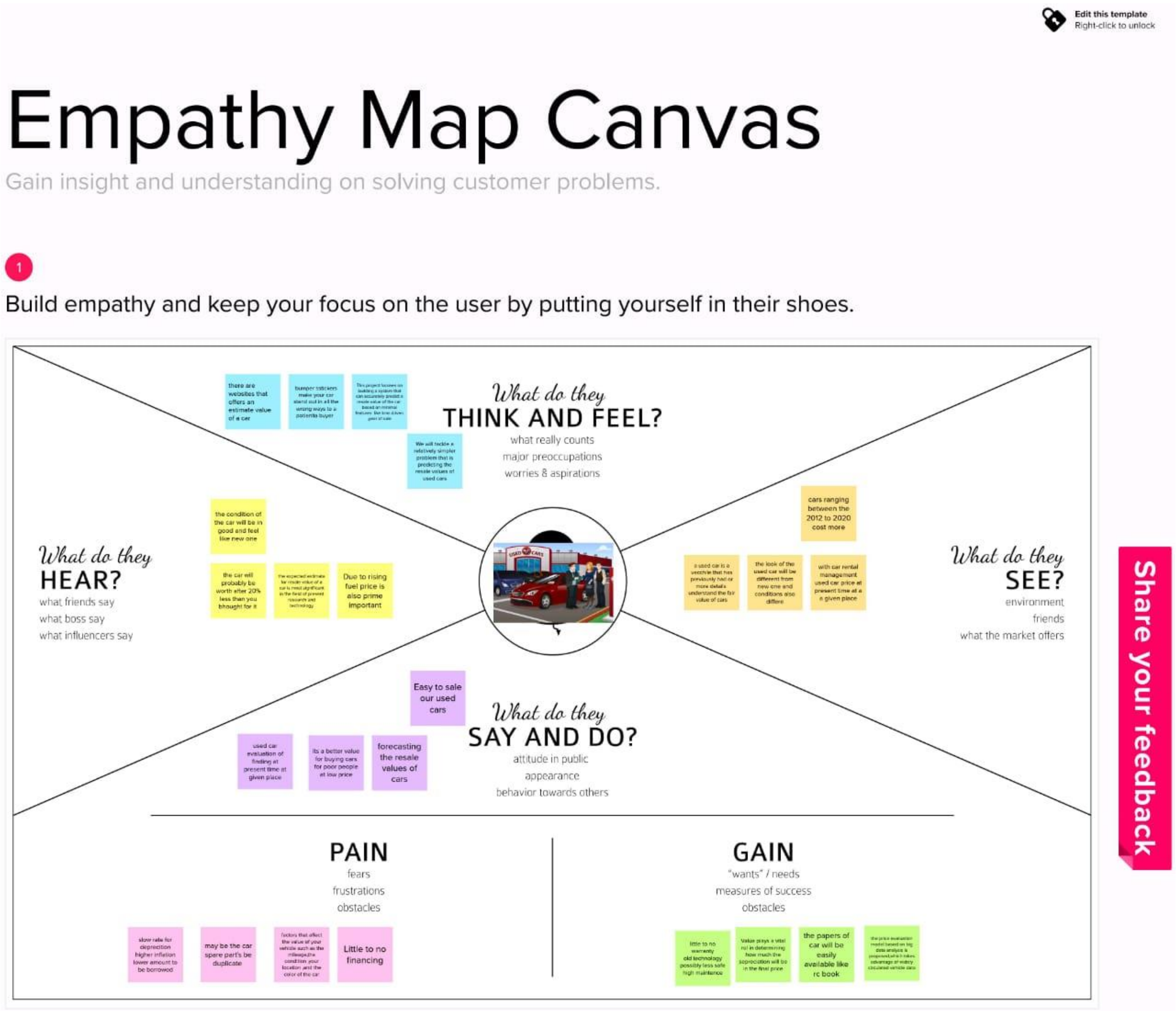
3.1 Empathy Map Canvas

An empathy map canvas is a more in-depth version of the original empathy map, which helps identify and describe the user's needs and pain points. And this is valuable information for improving the user experience.

Teams rely on user insights to map out what is important to their target audience, what influences them, and how they present themselves. This information is then used to create personas that help teams visualize users and empathize with them as individuals, rather than just as a vague marketing demographic or account number.

An empathy map canvas helps brands provide a better experience for users by helping teams understand the perspectives and mindset of their customers. Using a template to create an empathy map canvas reduces the preparation time and standardizes the process so you create empathy map canvases of similar quality

Empathy Map Canvas Visualizing and Predicting Heart Diseases with an Interactive Dashboard:



Edit this template
Right-click to unlock

Empathy Map Canvas

Gain insight and understanding on solving customer problems.

1

Build empathy and keep your focus on the user by putting yourself in their shoes.

What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations

there are websites that offers an estimate value of a car

bumper stickers make your car stand out in all the wrong ways to a potentia buyer

This project focuses on building a system that can accurately predict a resale value of the car based on minimal features like time driven year of sale

We will tackle a relatively simpler problem that is predicting the resale values of used cars

What do they
HEAR?
what friends say
what boss say
what influencers say

the condition of the car will be in good and feel like new one

the car will probably be worth after 20% less than you bhought for it

the expected estimate for resale value of a car is most significant in the field of present research and technology

Due to rising fuel price is also prime important

What do they
SEE?
environment
friends
what the market offers

cars ranging between the 2012 to 2020 cost more

a used car is a vecchile that has previously had or more details understand the fair value of cars

the look of the used car will be different from new one and conditions also differe

with car rental management used car price at present time at a a given place

What do they
SAY AND DO?
attitude in public
appearance
behavior towards others

Easy to sale our used cars

used car evaluation of finding at present time at given place

its a better value for buying cars for poor people at low price

forecasting the resale values of cars

PAIN
fears
frustrations
obstacles

slow rate for deprecition higher inflation lower amount to be borrowed

may be the car spere part's be duplicate

factors that affect the value of your vehicle such as the mileage,the condition your location and the color of the car

Little to no financing

GAIN
"wants" / needs
measures of success
obstacles

little to no warranty
old technology
possibly less safe
high maintence

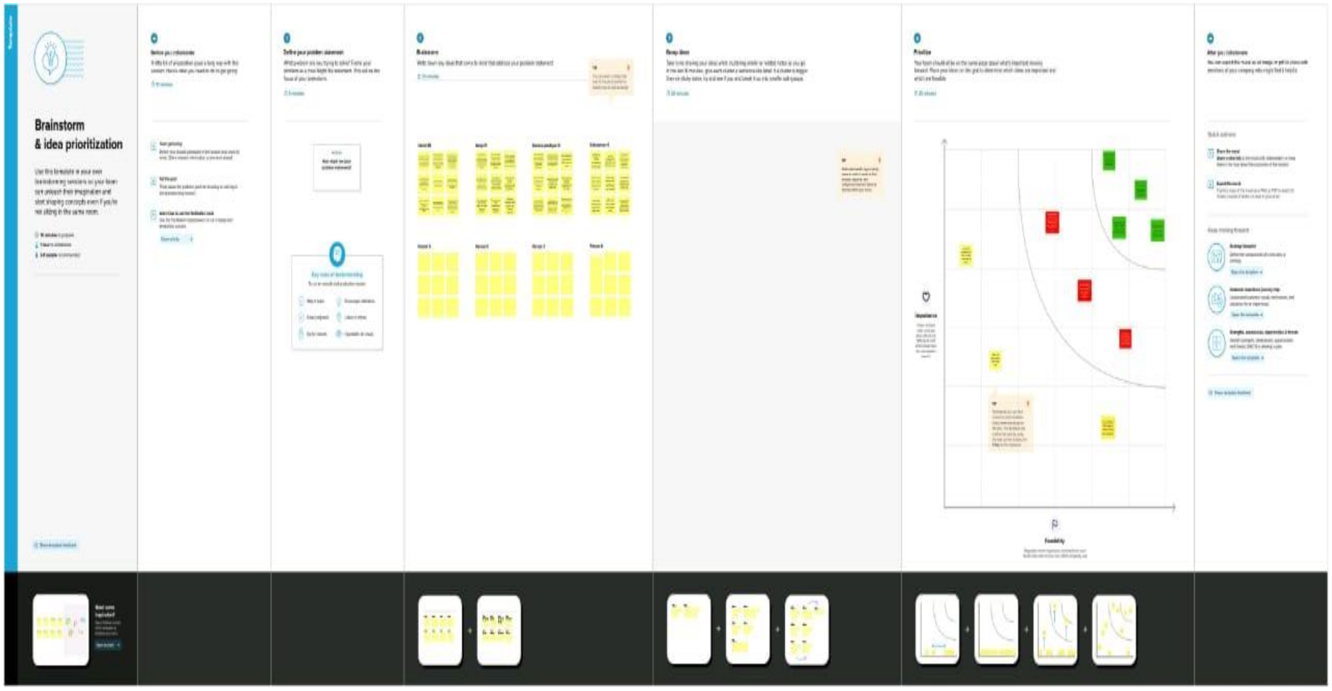
Value plays a vital rol in determining how much the depreciation will be in the final price

the papers of car will be easily available like rc book

the price evaluation model based on big data analysis is proposed,which takes advantage of widely circulated vehicle data

3.2 Ideation & Brainstorming

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich number of creative solutions.



3.3 Proposed Solution

Project team shall fill the following information in proposed solution template.

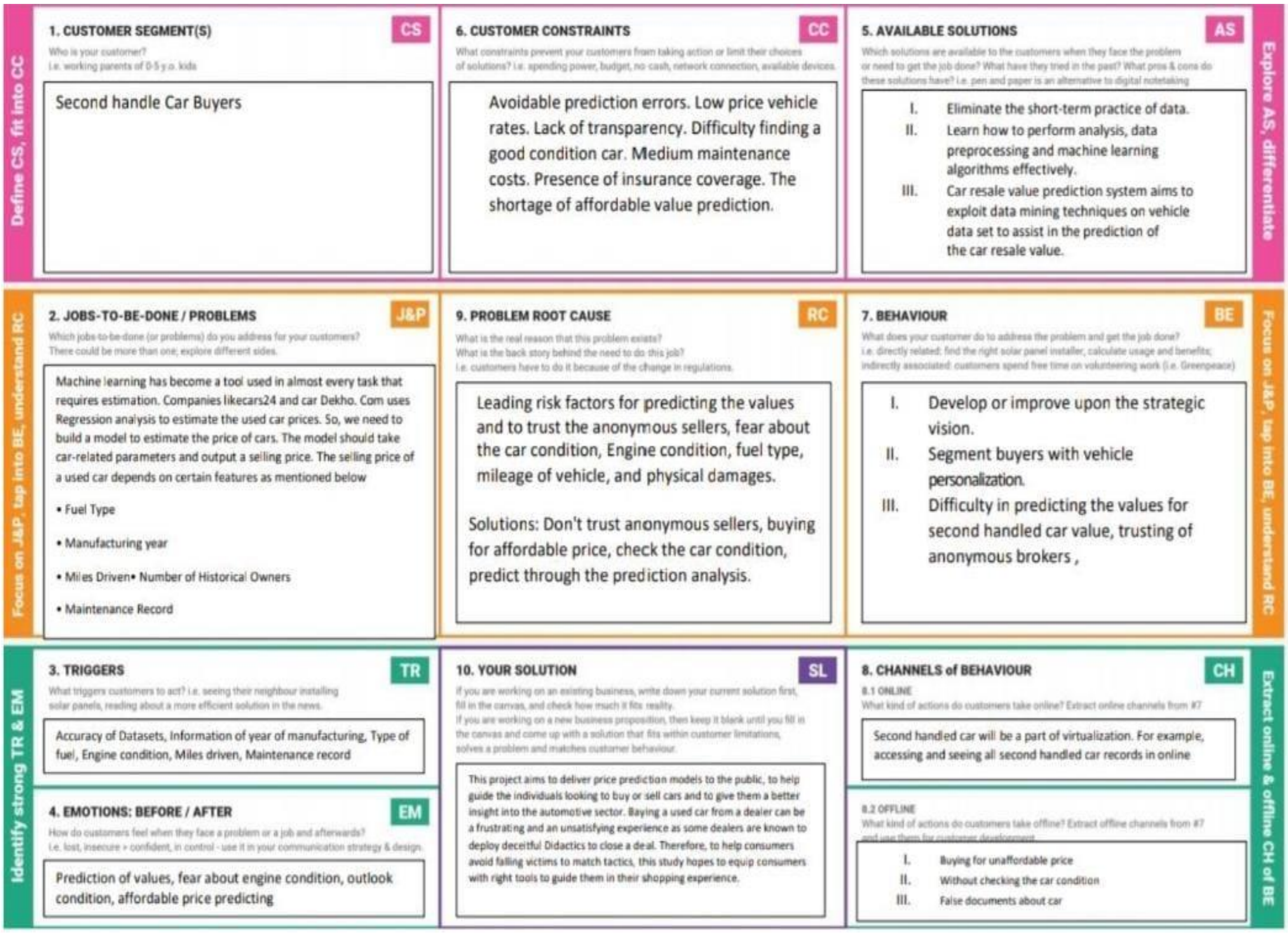
S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	Machine learning has become a tool used in almost every task that requires estimation. Companies like cars24 and cardekho. Com uses Regression analysis to estimate the used car prices. So we need to build a model to estimate the price of cars. The model should take carrelated parameters and output a selling price. The selling price of a used car depends on certain features as mentioned below • Fuel Type • Manufacturing year • Miles Driven • Maintenance Record This is a supervised learning problem and can be solved using regression techniques. We need to predict the selling price of a car based on the given car's features. Supervised Regression problems require labeled data where our target or dependent variable is the selling price of a car. All other features are independent variables.
2.	Idea / Solution description	This project aims to deliver price prediction models to the public, to help guide the individuals looking to buy or sell cars and to give them a better insight into the automotive sector. Baying a used car from a dealer can be a frustrating and an unsatisfying experience as some dealers are known to deploy deceitful Dic tactics to close a deal. Therefore, to help consumers avoid falling victims to mach tactics, this study hopes to equip consumers with right tools to guide them in their shopping experience. Another goal of the project is to

		explore new methods to evaluate used cars prices and to compare t their accuracies. Considering this is an interesting research topic in the research community, and in confirong heir footsteps, we hope to achieve significant results wing more advanced methods of previous work
3.	Novelty / Uniqueness	As there are so many ongoing experiments that use statistical approaches and some traditional methods to focus on predicting item sales. Most researches have experimented by taking a single algorithm to predict sales. In this thesis Machine Learning algorithms such as Simple Linear Regression, Support Vector Regression, Gradient Boosting algorithm, and Random Forest Regression are considered for prediction and the most effective metrics such as accuracy, mean absolute error, and max error are considered for measuring algorithm efficiency. This method will be very beneficial in the future for advanced item sales forecasting
4.	Social Impact / Customer Satisfaction	In the study, the variables having significant effects on the price of the second hand car were determined. A prediction model was established with these variables. The coefficient of determination (R2) of this model was calculated as 89.1%. The variables included in the estimation model are Brand, Model, Model Year, Fuel Type, Horse Power, Kilometer, Manual Air Conditioning, Fog Lights, Seat Air Cushion, Leather Steering Wheel, Wheel Rim, Automatic Air Conditioning, Start Stop, Rain Sensor, Sunroof, Electric Folding Mirrors, Xenon Headlight, Knee Airbag, Upholstery Leather, Memory

		Seat, 4X4, Parking Assistant, Vacuum Door.
5.	Business Model (Revenue Model)	Deciding whether a used car is worth the posted price when you see listings online can be difficult. Several factors, including mileage, make, model, year, etc. can influence the actual worth of a car. From the perspective of a seller, it is also a dilemma to price a used car appropriately[2-3]. Based on existing data, the aim is to use machine learning algorithms to develop models for predicting used car prices.
6.	Scalability of the Solution	We started with understanding the use case of machine learning in the Automotive industry and how machine learning has transformed the driving experience. Moving on, we looked at the various factors that affect the resale value of a used car and performed exploratory data analysis (EDA). Further, we build a Random Forest Regression model to predict the resale value of a used car. Finally, we evaluated the performance of the model using the R squared score and Residual Plot.

3.4 Problem Solution fit

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem.



Define CS, fit into CC

1. CUSTOMER SEGMENT(S)

CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

Second handle Car Buyers

6. CUSTOMER CONSTRAINTS

CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

Avoidable prediction errors. Low price vehicle rates. Lack of transparency. Difficulty finding a good condition car. Medium maintenance costs. Presence of insurance coverage. The shortage of affordable value prediction.

5. AVAILABLE SOLUTIONS

AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

I. Eliminate the short-term practice of data.
II. Learn how to perform analysis, data preprocessing and machine learning algorithms effectively.
III. Car resale value prediction system aims to exploit data mining techniques on vehicle data set to assist in the prediction of the car resale value.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS

J&P

Which jobs-to-be-done (or problems) do you address for your customers?
There could be more than one; explore different sides.

Machine learning has become a tool used in almost every task that requires estimation. Companies likecars24 and car Dekho. Com uses Regression analysis to estimate the used car prices. So, we need to build a model to estimate the price of cars. The model should take car-related parameters and output a selling price. The selling price of a used car depends on certain features as mentioned below

- Fuel Type
- Manufacturing year
- Miles Driven• Number of Historical Owners
- Maintenance Record

9. PROBLEM ROOT CAUSE

RC

What is the real reason that this problem exists?
What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

Leading risk factors for predicting the values and to trust the anonymous sellers, fear about the car condition, Engine condition, fuel type, mileage of vehicle, and physical damages.

Solutions: Don't trust anonymous sellers, buying for affordable price, check the car condition, predict through the prediction analysis.

7. BEHAVIOUR

BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

I. Develop or improve upon the strategic vision.
II. Segment buyers with vehicle personalization.
III. Difficulty in predicting the values for second handled car value, trusting of anonymous brokers ,

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS

TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

Accuracy of Datasets, Information of year of manufacturing, Type of fuel, Engine condition, Miles driven, Maintenance record

4. EMOTIONS: BEFORE / AFTER

EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

Prediction of values, fear about engine condition, outlook condition, affordable price predicting

10. YOUR SOLUTION

SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

This project aims to deliver price prediction models to the public, to help guide the individuals looking to buy or sell cars and to give them a better insight into the automotive sector. Baying a used car from a dealer can be a frustrating and an unsatisfying experience as some dealers are known to deploy deceitful Didactics to close a deal. Therefore, to help consumers avoid falling victims to match tactics, this study hopes to equip consumers with right tools to guide them in their shopping experience.

8. CHANNELS of BEHAVIOUR

CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

Second handled car will be a part of virtualization. For example, accessing and seeing all second handled car records in online

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

I. Buying for unaffordable price
II. Without checking the car condition
III. False documents about car

Extract online & offline CH of BE

REQUIREMENT ANALYSIS

CHAPTER-4

REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR No	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Website
FR-2	User Confirmation	Confirmation via website
FR-3	Car Registration	Registering the car details
FR-4	Value Prediction	Predicting the car resale value

4.2 Non-Functional requirements

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Predicting the resale value
NFR-2	Security	Providing security to the website
NFR-3	Reliability	Providing high reliability by predicting values for different types of cars
NFR-4	Performance	Providing high performance by using some machine learning techniques
NFR-5	Availability	It is used for all types of cars
NFR-6	Scalability	Predicting values for different types of cars

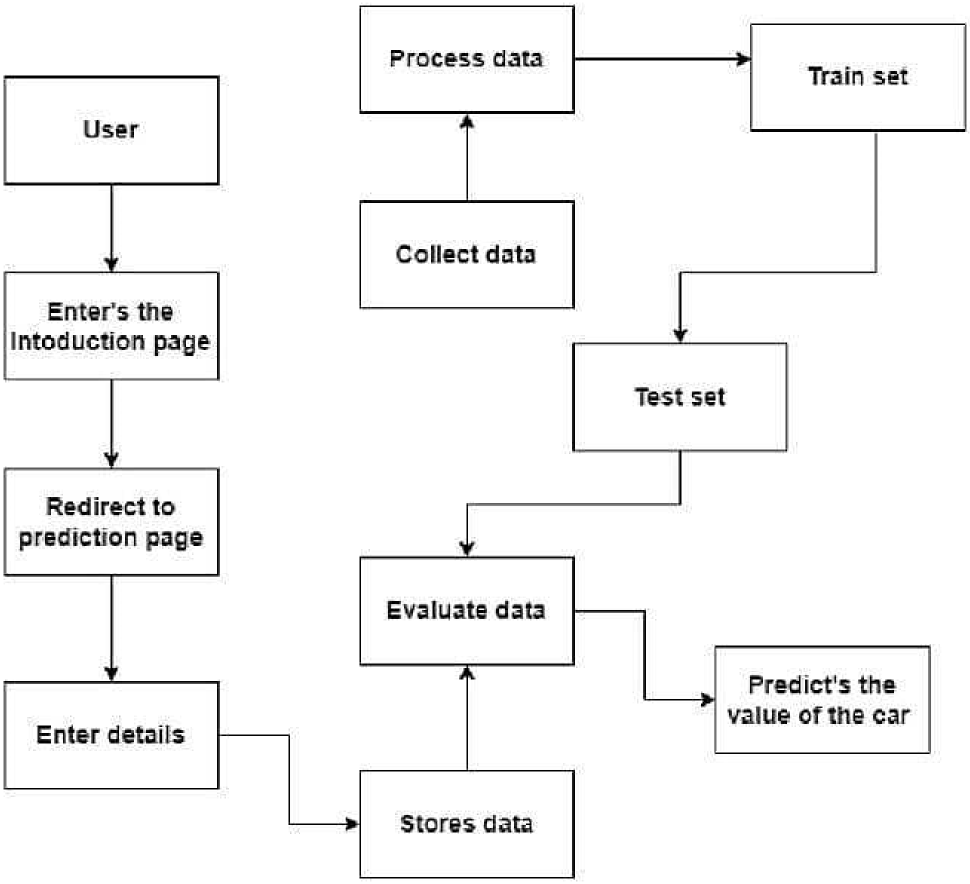
PROJECT DESIGN

CHAPTER-5

PROJECT DESIGN

5.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored



User
Enter's the Intoduction page
Redirect to prediction page
Enter details
Process data
Collect data
Train set
Test set
Evaluate data
Predict's the value of the car
Stores data

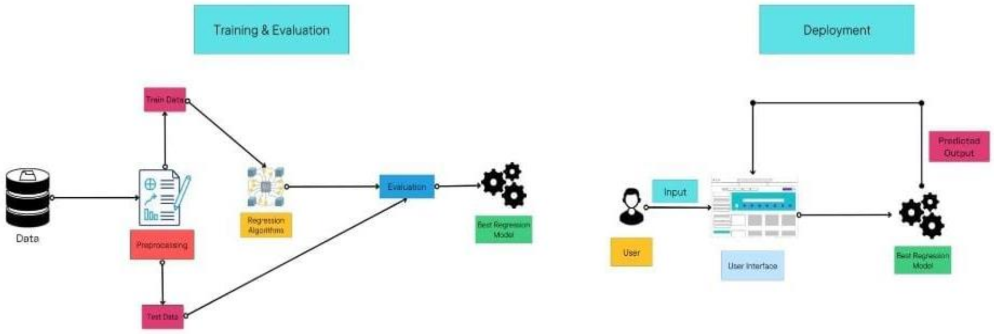
1). As a user, I can access to website using a web browser.
2). As a user, I can proceed to the prediction page by selecting the check value button in the home page.
3). As a user, I can use any of the appropriate mobile browser to enter into the website.

5.2 Solution & Technical Architecture

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behaviour, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

Solution Architecture Diagram:



5.3 User Stories

User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (web user)	Enters the browser	USN-1	As a user, I can access to website using a web browser	I can enter by selecting the appropriate web link	High	Sprint-1
		USN-2	As a user, I can proceed to the prediction page by selecting the check value button in the home page	I can enter into it without any acceptance	High	Sprint-1

Customer (mobile user)	Enters into a mobile browser	USN-3	As a user, I can use any of the appropriate mobile browser to enter into the website	I can enter by using an appropriate web link	Medium	Sprint-1

PROJECT PLANNING & SCHEDULING

CHAPTER-6

PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Product Backlog, Sprint Schedule, and Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Pre-process data	USN-1	Collect Dataset		Low	Sanjai P

Sprint-1		USN-2	Import required libraries		Low	Harish M
Sprint-1		USN-3	Read and clean data sets		Low	Kalai Selvan S
Sprint-2	Model building	USN-1	Split data into independent and dependent variables		Medium	Sanjai P
Sprint-2		USN-2	Apply using regression model		Medium	Eswara Pandiyan D
Sprint-3	Application building	USN-1	Build python flask application and HTML page		High	Kalai Selvan S & Eswara Pandiyan D
Sprint-3		USN-2	Execute and test		High	Harish M
Sprint-4	Training the model	USN-1	Train machine learning model		High	Sanjai P & Harish M
Sprint-4		USN-2	Integrate flask		High	Kalai Selvan S

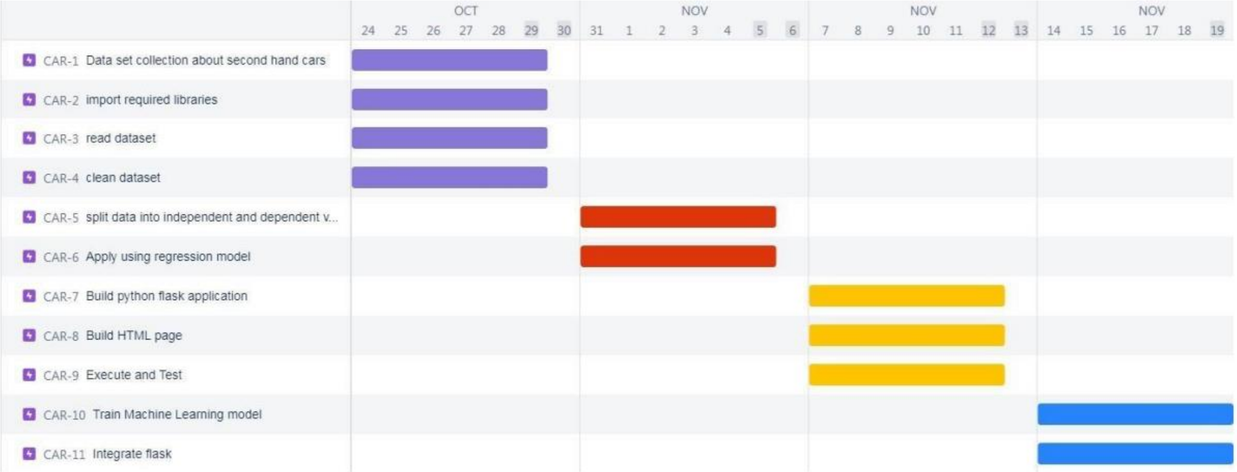
6.2 Sprint Delivery Schedule

Project Tracker, Velocity & Burndown Chart:

Sprint	**Total Story Points**	**Duration**	**Sprint Start Date**	**Sprint End Date (Planned)**	**Story Points Completed (as on Planned End Date)**	**Sprint Release Date (Actual)**
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022

Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Burndown Chart:



CAR-1 Data set collection about second hand cars

CAR-2 import required libraries

CAR-3 read dataset

CAR-4 clean dataset

CAR-5 split data into independent and dependent v...

CAR-6 Apply using regression model

CAR-7 Build python flask application

CAR-8 Build HTML page

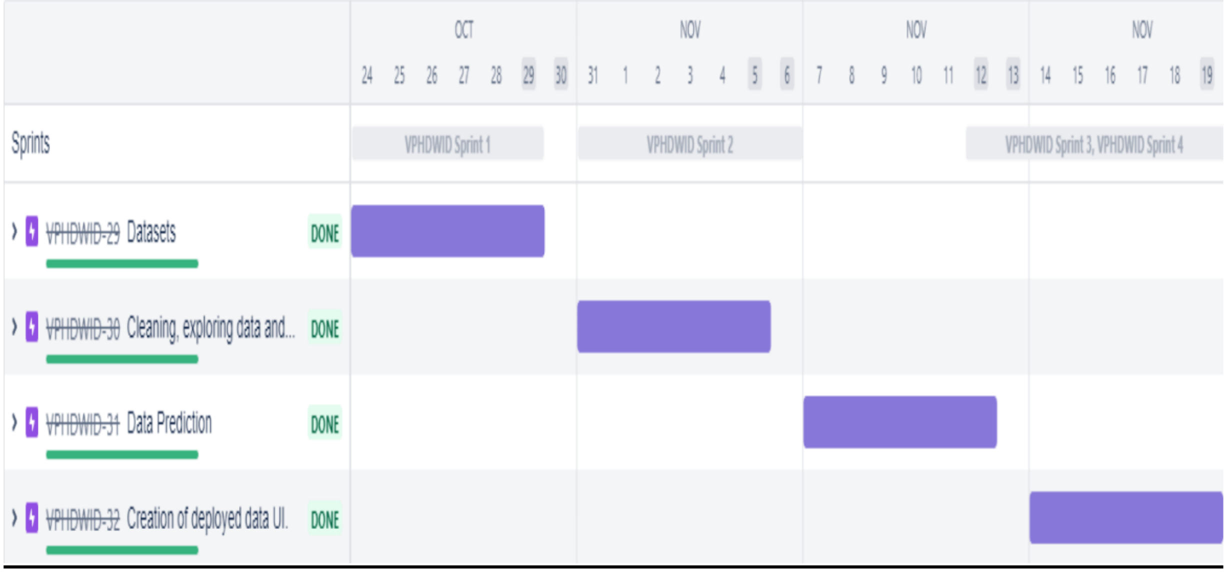
CAR-9 Execute and Test

CAR-10 Train Machine Learning model

CAR-11 Integrate flask

6.3 Reports from JIRA

Jira helps teams plan, assign, track, report, and manage work and brings teams together for everything from agile software development and customer support to start-ups and enterprises. Software teams build better with Jira Software, the #1 tool for agile teams. As a Jira administrator, you can create project categories so your team can view work across related projects in one place. Your team can use categories in advanced search, filters, reports, and more.

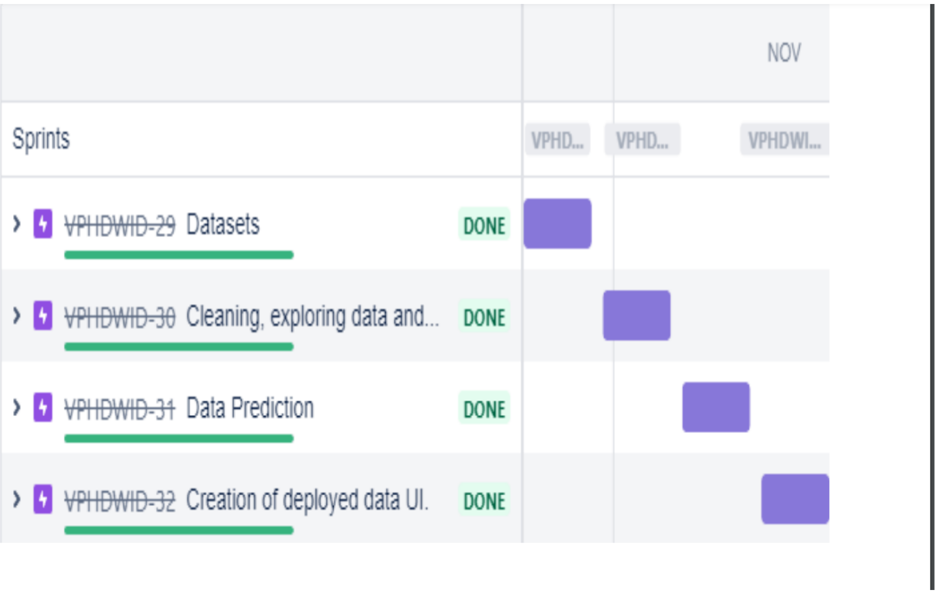


~~VPHDWID-29~~ Datasets DONE

~~VPHDWID-30~~ Cleaning, exploring data and... DONE

~~VPHDWID-31~~ Data Prediction DONE

~~VPHDWID-32~~ Creation of deployed data UI. DONE



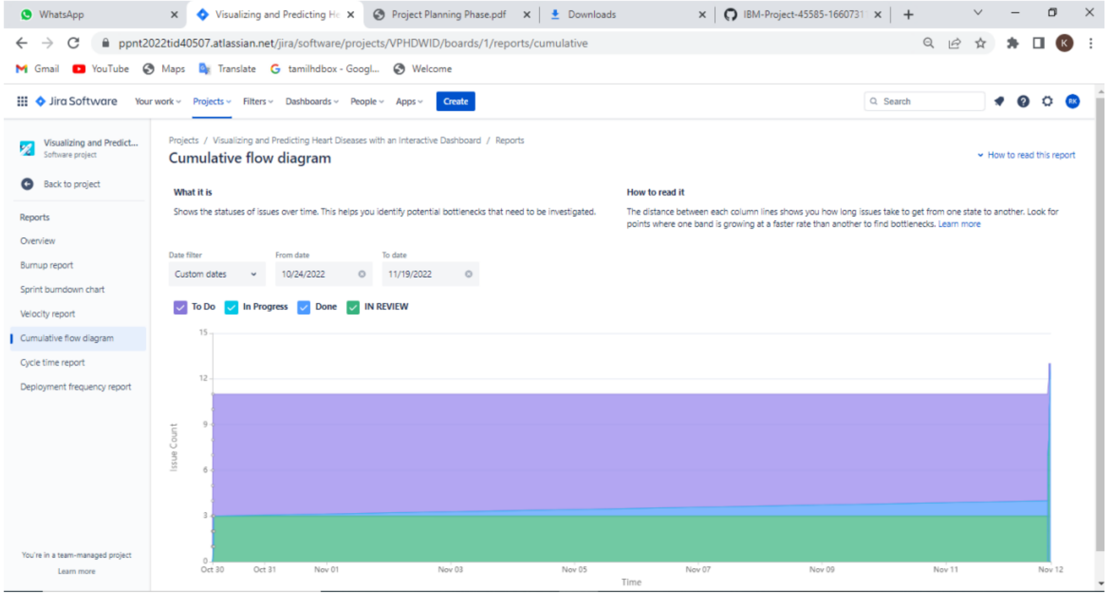
~~VPHDWID-29~~ Datasets DONE

~~VPHDWID-30~~ Cleaning, exploring data and... DONE

~~VPHDWID-31~~ Data Prediction DONE

~~VPHDWID-32~~ Creation of deployed data UI. DONE

CUMULATIVE JIRA FILE:



Cumulative flow diagram

What it is

Shows the statuses of issues over time. This helps you identify potential bottlenecks that need to be investigated.

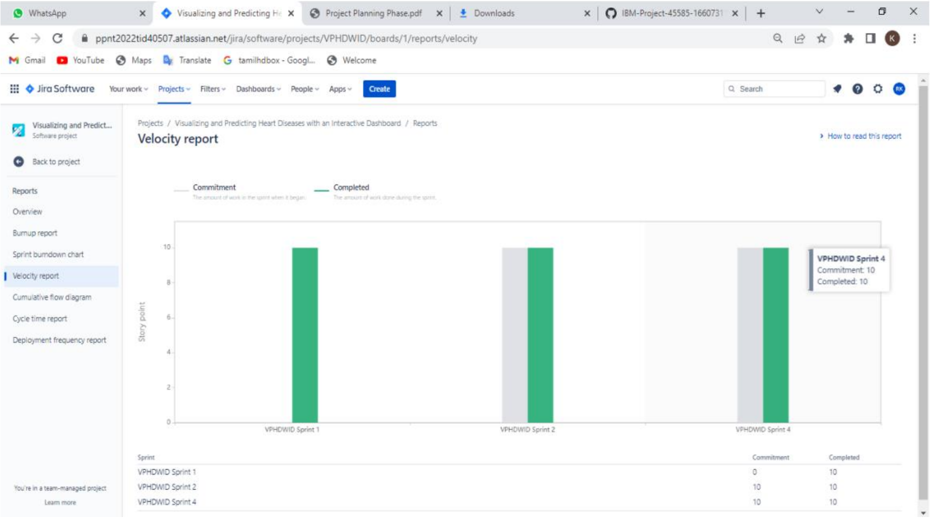
How to read it

The distance between each column lines shows you how long issues take to get from one state to another. Look for points where one band is growing at a faster rate than another to find bottlenecks. Learn more

Date filter: Custom dates
From date: 10/24/2022
To date: 11/19/2022

☑ To Do ☑ In Progress ☑ Done ☑ IN REVIEW

VELOCITY:

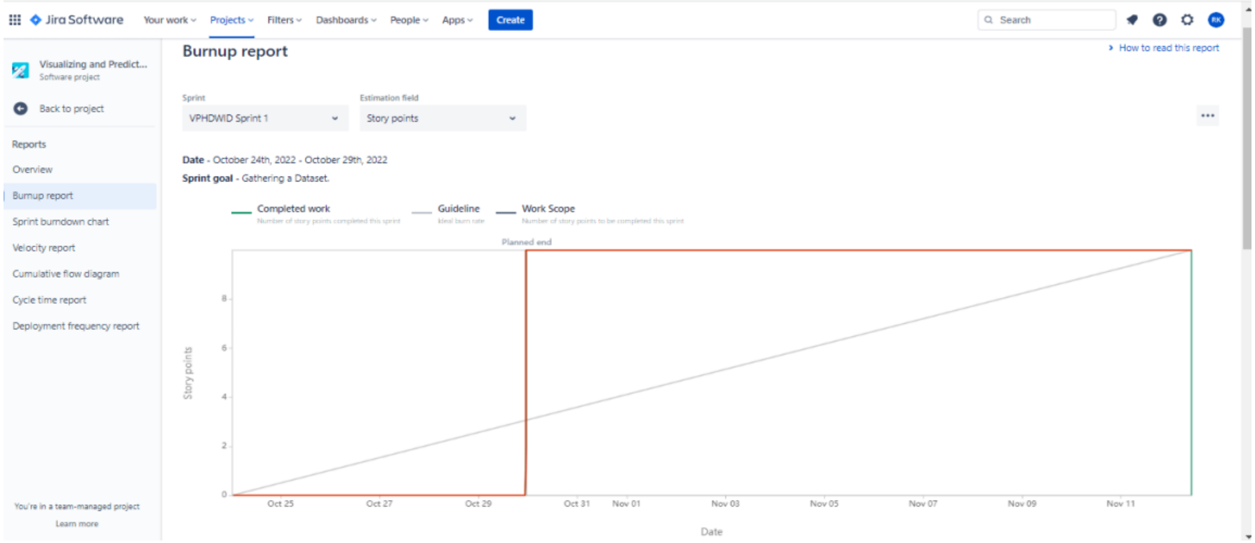


Velocity report

Sprint	Commitment	Completed
VPHDWID Sprint 1	0	10
VPHDWID Sprint 2	10	10
VPHDWID Sprint 4	10	10

BURNUP REPORT:

SPRINT-1:



Burnup report

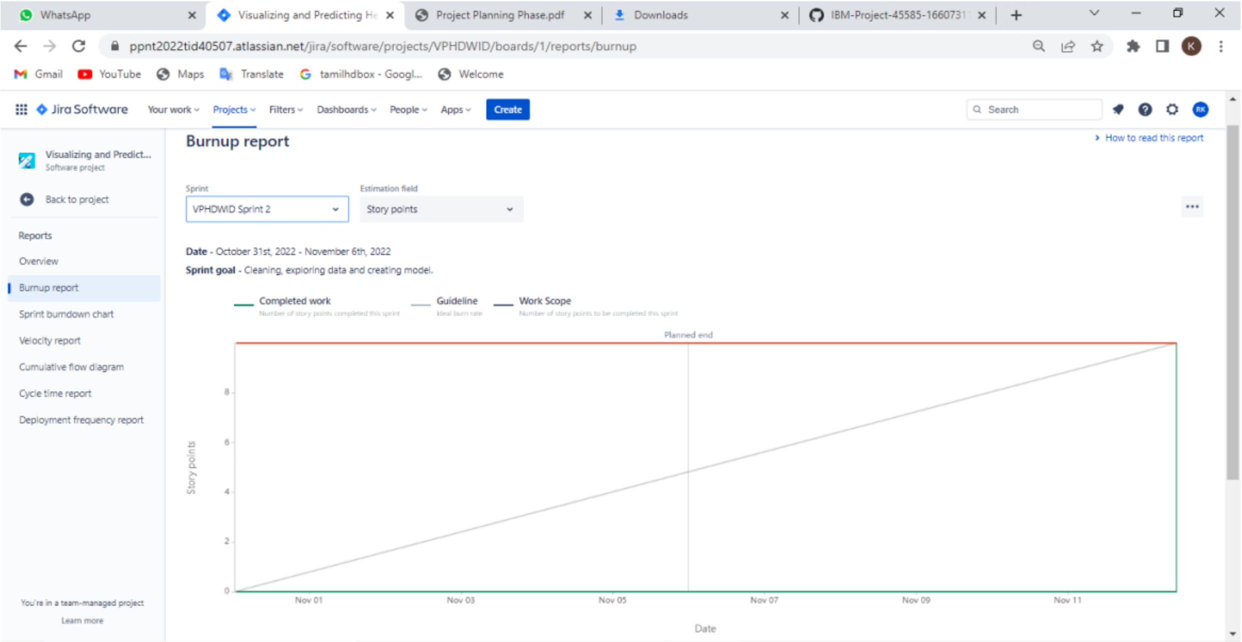
Sprint: VPHDWID Sprint 1

Estimation field: Story points

Date - October 24th, 2022 - October 29th, 2022

Sprint goal - Gathering a Dataset.

SPRINT-2



Burnup report

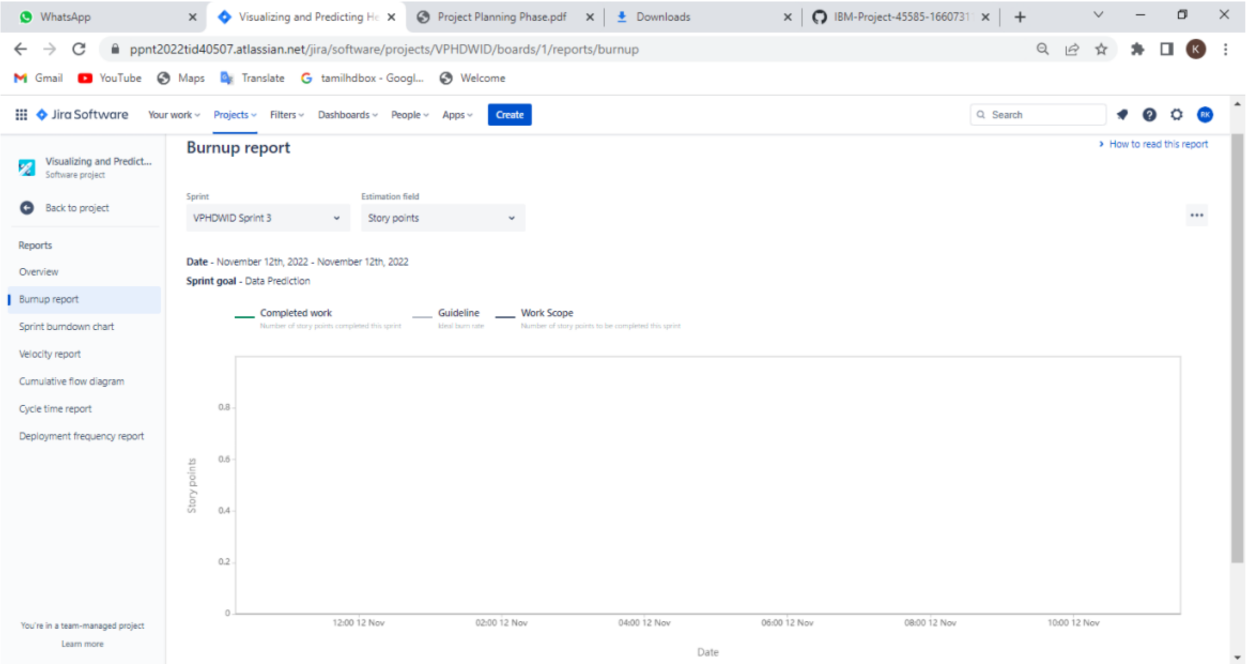
Sprint: VPHDWID Sprint 2

Estimation field: Story points

Date - October 31st, 2022 - November 6th, 2022

Sprint goal - Cleaning, exploring data and creating model.

SPRINT-3



Burnup report

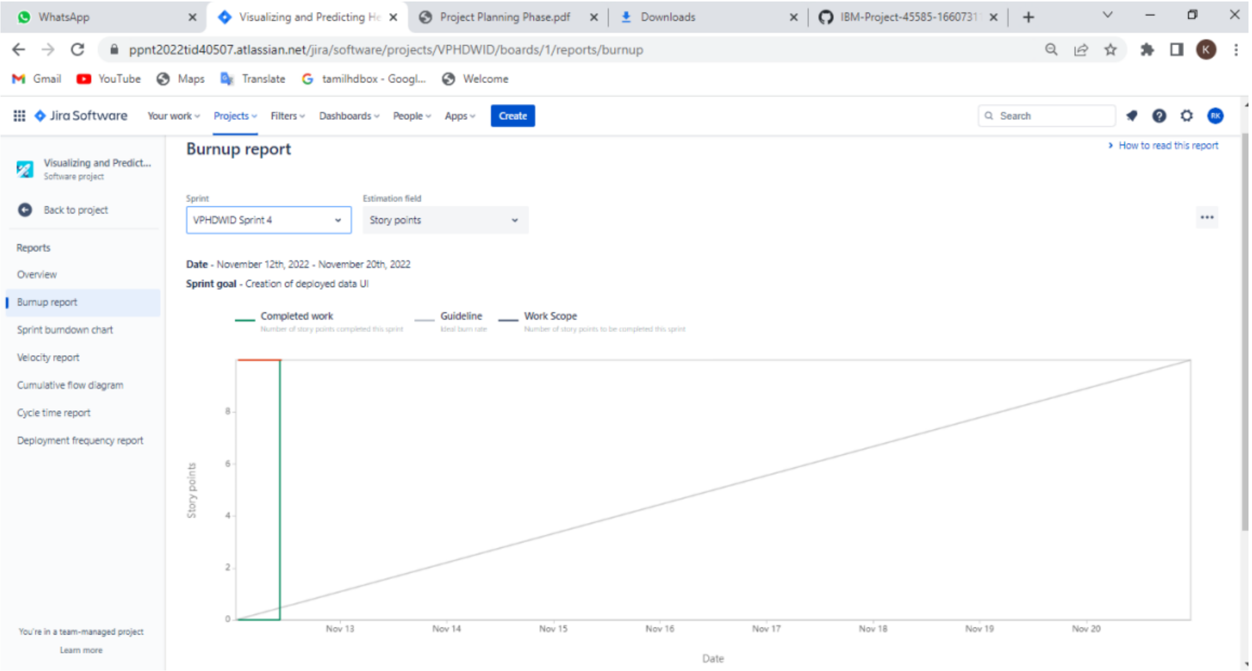
Sprint: VPHDWID Sprint 3

Estimation field: Story points

Date - November 12th, 2022 - November 12th, 2022

Sprint goal - Data Prediction

SPRINT-4



Burnup report

Sprint: VPHDWID Sprint 4

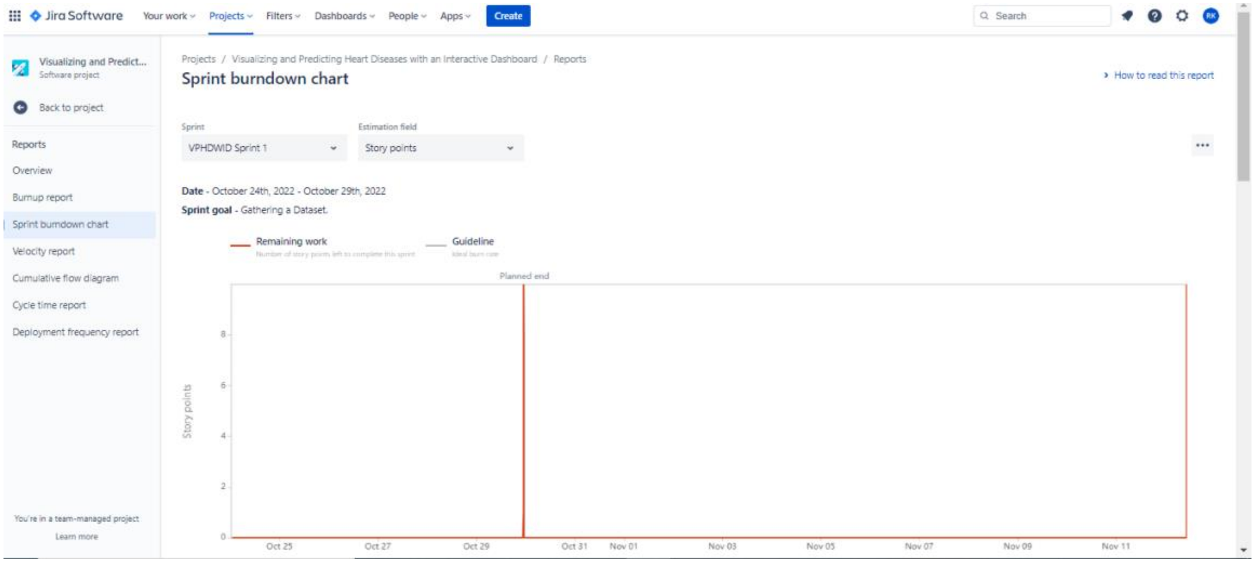
Estimation field: Story points

Date - November 12th, 2022 - November 20th, 2022

Sprint goal - Creation of deployed data UI

Burndown Chart:

Burndown Chart Sprint-1:



Projects / Visualizing and Predicting Heart Diseases with an Interactive Dashboard / Reports

Sprint burndown chart

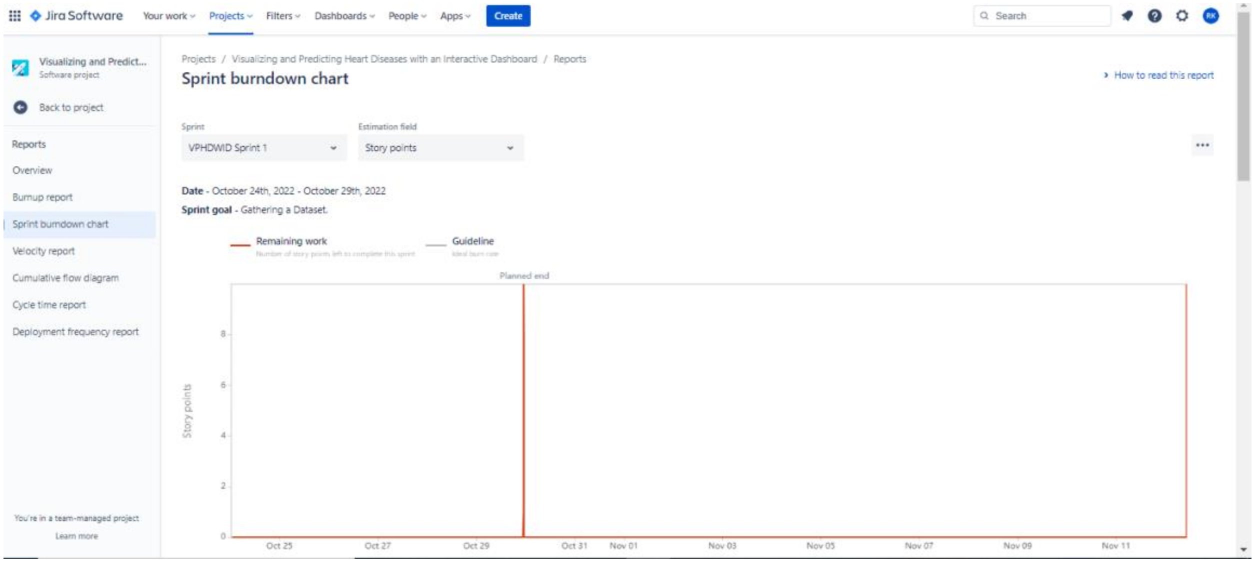
Sprint: VPHDWID Sprint 1

Estimation field: Story points

Date - October 24th, 2022 - October 29th, 2022

Sprint goal - Gathering a Dataset.

Burndown Chart Sprint-2:



Projects / Visualizing and Predicting Heart Diseases with an Interactive Dashboard / Reports

Sprint burndown chart

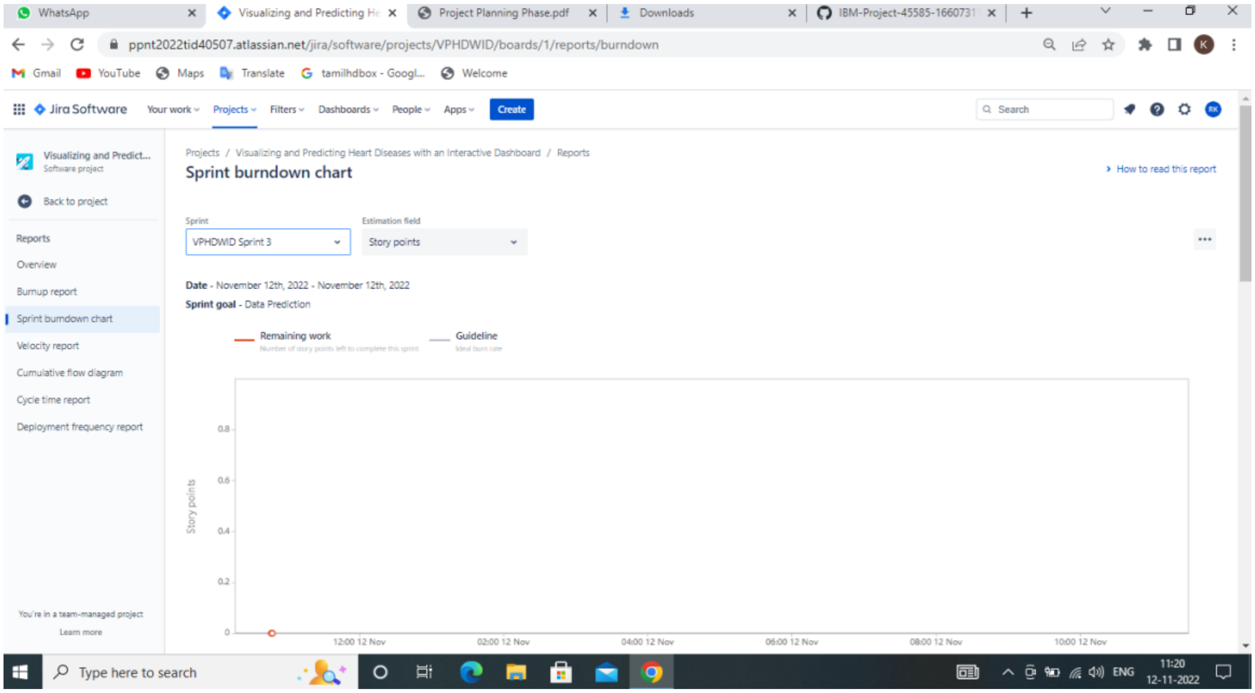
Sprint: VPHDWID Sprint 1

Estimation field: Story points

Date - October 24th, 2022 - October 29th, 2022

Sprint goal - Gathering a Dataset.

Burndown Chart Sprint-3:



Sprint burndown chart

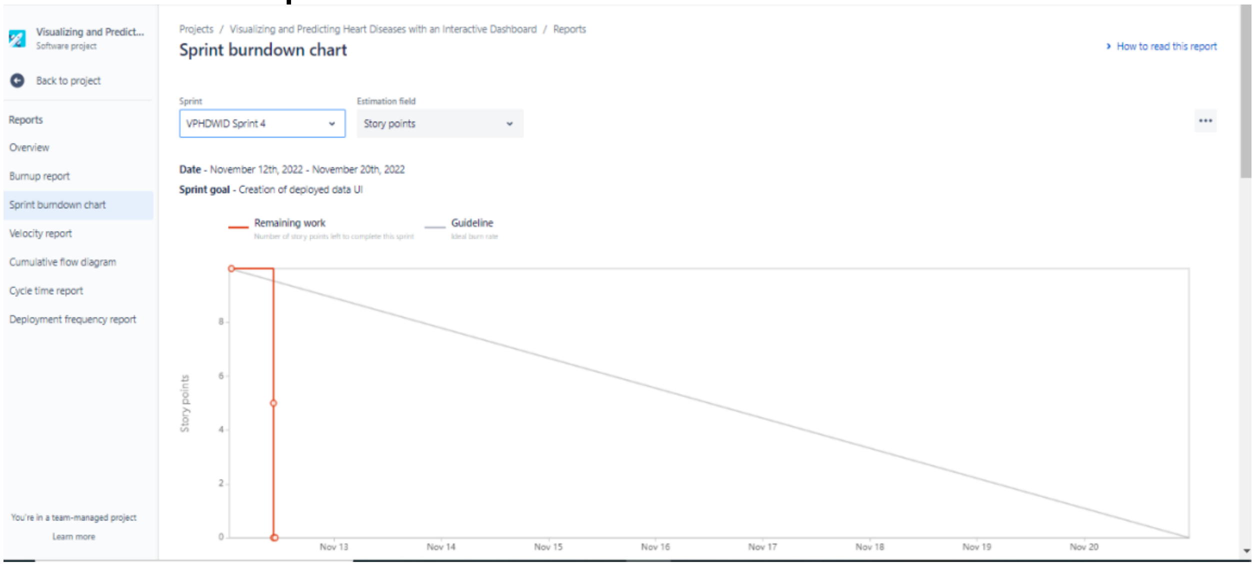
Sprint: VPHDWID Sprint 3

Estimation field: Story points

Date - November 12th, 2022 - November 12th, 2022

Sprint goal - Data Prediction

Burndown Chart Sprint-4:



Sprint burndown chart

Sprint: VPHDWID Sprint 4

Estimation field: Story points

Date - November 12th, 2022 - November 20th, 2022

Sprint goal - Creation of deployed data UI

CODING & SOLUTIONING

CHAPTER-7
CODING & SOLUTIONING

7.1 Random Forest

- Random Forest is a supervised learning algorithm. Random forest can be used for both classification and regression problems, by using random forest regressor we can use random forest on regression problems.
- But we have used random forest on classification in this internship project so we will only consider the classification part

7.1.1 Random Forest pseudocode

- Randomly select “k” features from total “m” features. Where k << m • Among the “k” features, calculate the node “d” using the best split point.• Split the node into daughter nodes using the best split. Repeat 1 to 3 steps until the “l” number of nodes has been reached.
- Build forest by repeating steps 1 to 4 for “n” number times to create “n” number of trees.

7.1.2 Random Forest

prediction pseudocode Takes the test features and use the rules of each randomly created decision tree to predict the outcome and stores the predicted outcome (target).

- Calculate the votes for each predicted target. ✓ Consider the highly voted predicted target as the final prediction from the random forest algorithm

Code:

```
max_accuracy = 0
for x in range(500):
rf_classifier = RandomForestClassifier(random_state=x)
rf_classifier.fit(X_train,Y_train)
Y_pred_rf = rf_classifier.predict(X_test)
current_accuracy =
round(accuracy_score(Y_pred_rf,Y_test)*100,2)
if(current_accuracy>max_accuracy):
max_accuracy = current_accuracy
best_x = x
print(max_accuracy)
print(best_x)
rf_classifier =
RandomForestClassifier(random_state=best_x)
rf_classifier.fit(X_train,Y_train)
Y_pred_rf = rf_classifier.predict(X_test)
Y_pred_rf.shape
score_rf = round(accuracy_score(Y_pred_rf,Y_test)*100,2)
score_rf
```

7.2. K-Nearest Neighbors

We can implement a KNN model by following the below steps:

- Load the data
- Initialize the value of k

● For getting the predicted class, iterate from 1 to total number of training data points Calculate the distance between test data and each row of training data. Here we will use Euclidean distance as our distance metric since it’s the most popular method. The other metrics that can be used are Chebyshev, cosine, etc.
○ Sort the calculated distances in ascending order based on distance values
○ Get top k rows from the sorted array
○ Get the most frequent class of these rows
○ Return the predicted class

Code:

```
knn_classifier=
KNeighborsClassifier(n_neighbors=31,leaf_size=30)
knn_classifier.fit(X_train,Y_train)
Y_pred_knn = knn_classifier.predict(X_test)
score_knn = round(accuracy_score(Y_pred_knn,Y_test)*100,2)
score_knn
```

7.3 Decision Tree

Pseudocode:

- Place the best attribute of the dataset at the root of the tree.
- Split the training set into subsets. Subsets should be made in such a way that each subset contains data with the same value for an attribute.
- Repeat step 1 and step 2 on each subset until you find leaf nodes in all the branches of the tree. Assumptions while creating a Decision Tree- At the beginning, the whole training set is considered as the root.
- Feature values are preferred to be categorical. If the values are continuous then they are discretized prior to building the model.
- Records are distributed recursively on the basis of attribute values.
- Order to place attributes as root or internal node of the tree is done by using some statistical approach.

The popular attribute selection measures:

- Information gain
- Gini index

Attribute selection method- A dataset consists of “n” attributes then deciding which attribute to place at the root or at different levels of the tree as internal nodes is a complicated step. By just randomly selecting any node to be the root can’t solve the issue. If we follow a random approach, it may give us bad results with low accuracy. To solve this attribute selection problem, researchers worked and devised some solutions. They suggested using some criterion like information gain, Gini index, etc. These criterions will calculate values for every attribute. The values are sorted, and attributes are placed in the tree by following the order i.e., the attribute with a high value (in case of information gain) is placed at the root. While using information Gain as a criterion, we assume attributes to be categorical, and for Gini index, attributes are assumed to be continuous.

Gini Index - Gini Index is a metric to measure how often a randomly chosen element would be incorrectly identified. It means an attribute with a
lower Gini index should be preferred.

Code:

```
dt_classifier = DecisionTreeClassifier(
max_depth=20,
min_samples_split=2,
min_samples_leaf=1,
min_weight_fraction_leaf=0.00001,
max_features='auto',
random_state=46)
dt_classifier.fit(X_train, Y_train)
Y_pred_dt=dt_classifier.predict(X_test)
score_dt = round(accuracy_score(Y_pred_dt,Y_test)*100,2)
score_dt
```

7.4 Naïve Bayes

Bayes'Theorem is stated as:
P(h|d) = (P(d|h) * P(h)) / P(d)
P(h|d) is the probability of hypothesis h given the data d. This is called the posterior probability.
P(d|h) is the probability of data d given that the hypothesis h was true.
P(h) is the probability of hypothesis h being true (regardless of the data). This is called the prior probability of h.
P(d) is the probability of the data (regardless of the hypothesis).

We are interested in calculating the posterior probability of P(h|d) from the prior probability p(h) with P(D) and P(d|h). After calculating the posterior probability for a number of different hypotheses, we will select the hypothesis with the highest probability. This is the maximum probable hypothesis and may formally be called the (MAP) hypothesis.

This can be written as:

MAP(h) = max(P(h|d)) or
MAP(h) = max((P(d|h) * P(h)) / P(d)) or
MAP(h) = max(P(d|h) * P(h))

The P(d) is a normalizing term which allows us to calculate the probability. We can drop it when we are interested in the most probable hypothesis as it is constant and only used to normalize. Back to classification, if we have an even number of instances in each class in our training data, then the probability of each class (e.g. P(h)) will be equal. Again, this would be a constant term in our equation, and we could drop it so that we end up with:

MAP(h) = max(P(d|h))

Naive Bayes is a classification algorithm for binary (two-class) and multi-class classification problems. The technique is easiest to understand when described using binary or categorical input values. It is called Naive Bayes or Idiot Bayes because the calculation of the probabilities for each hypothesis are simplified to make their calculation tractable. Rather than attempting to calculate the values of each attribute value P (d1, d2, d3|h), they are assumed to be conditionally independent given the target value and calculated as P(d1|h) * P(d2|H) and so on. This is a very strong assumption that is most unlikely in real data, i.e. that the attributes do not interact. Nevertheless, the approach performs surprisingly well on data where this assumption does not **hold. MAP(h) = max(P(d|h) * P(h)**

Gaussian Naïve Bayes:

mean(x) = 1/n * sum(x)

Where n is the number of instances and x are the values for an input variable in your trainingdata. We can calculate the standard deviation usingthe following equation:

standard deviation(x) = sqrt (1/n *sum(xi-mean(x)^2))

This is thesquare root of the averagesquared difference of eachvalue of x from the meanvalue of x, where n is the number of instances, sqrt() isthe square root function, sum() is the sum function,xi is a specific valueof the x variable for the i'th instance and mean(x) is described above,and
^2 is the square. Gaussian PDF with a new input for the variable, and in return the GaussianPDF will provide an estimateof the probability ofthat new input value for that class.

pdf (x, mean,sd) = (1 / (sqrt (2 * PI) * sd)) * exp (-((x-mean^2)/(2*sd^2)))

Where pdf(x) is the GaussianProbability Density Function(PDF), sqrt () is the square root, mean and sd are the mean and standarddeviation calculated above,Pi is the numerical constant, exp () is the numericalconstant e or Euler's numberraised to powerand x is the input value for theinput variable.

Code:

```
nb_classifier = GaussianNB( var_smoothing=1e-50)
nb_classifier.fit(X_train,Y_train)
nb_classifier.predict(X_test)
Y_pred_nb = nb_classifier.predict(X_test)
score_nb = round(accuracy_score(Y_pred_nb,Y_test)*100,2)
score_nb
```

7.5 Web App Code

```
import pandas
from flask import Flask, render_template, request

import pandas as pd
import numpy as np


from sklearn.preprocessing import LabelEncoder

app =Flask(__name__)
import pickle
filename = 'resale_model.sav'
model_rand= pickle.load(open(filename, 'rb'))
@app.route('/')
```

```
def index():
    return render_template('home.html')

@app.route('/predict')
def predict():
    return render_template('predict.html')
@app.route('/y_predict', methods= ['POST'])
def y_predict():
    regyear = int(request.form['Registrationyear'])
    powerps = int(request.form['PowerofcarinPS'])
    kms =int(request.form['KilometersDriven'])
    regmonth =int(request.form.get('Registrationmonth'))
    gearbox =(request.form['Geartype'])
    damage =request.form['cd']
    model = request.form.get('model')
    brand =request.form.get('brand')
    fuelType =request.form.get('fueltype')
    vehicletype = request.form.get('vechicletype')
    row = {'vehicleType': vehicletype,'yearOfRegistration':regyear,
           'gearbox': gearbox,'powerPS': powerps, 'model':model,'kilometer': kms,
           'monthOfRegistration': regmonth,'fuelType': fuelType,
           'brand': brand,'notRepairedDamage': damage}
    print(row)
    new_row = pd.DataFrame([row])
    new_df = pd.DataFrame(columns = ['vehicleType', 'yearOfRegistration', 'gearbox',
                            'powerPS','model', 'kilometer', 'monthOfRegistration', 'fuelType',
                            'brand','notRepairedDamage' ])
    new_df=pd.concat([new_df,new_row], ignore_index = True)
    new_df['monthOfRegistration']= new_df['monthOfRegistration'].astype(int)
    labels = ['gearbox', 'notRepairedDamage', 'model', 'brand', 'fuelType', 'vehicleType']
    mapper = {}
    for i in labels:
        mapper[i] = LabelEncoder()
        mapper[i].fit(new_df[i])
        mapper[i].classes_ = np.load(str('classes'+i+'.npy'),allow_pickle=True)
        tr = mapper[i].fit_transform(new_df[i])
        new_df.loc[:, i + '_labels'] = pd.Series (tr, index=new_df.index)
    labeled = new_df[ ['yearOfRegistration', 'powerPS','kilometer','monthOfRegistration']
        + [x+'_labels' for x in labels]]
    X = labeled.values
    print(X)
    y_prediction = model_rand.predict(X)
    print(y_prediction)
    df_ev = np.exp(y_prediction)
    print("df_ev: {} ".format(df_ev))
    return render_template('predict.html',y = 'The resale value predicted is
{:.2f}$'.format(df_ev[0]))

if __name__=='__main__':
    app.run(debug= False)
```

7.6 Libraries used

Python has a vast reserve of inbuilt standard libraries which includes areas like web services tools, string operation, data analysis, and machine learning, etc. The complex programming tasks can be dealt with ease using these inbuilt libraries as it reduces the size of code with many inbuilt functions that do the job pretty well for its user.

7.6.1 Data Visualization

Matplotlib: Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical mathematics extension NumPy, a big data numerical handling resource.

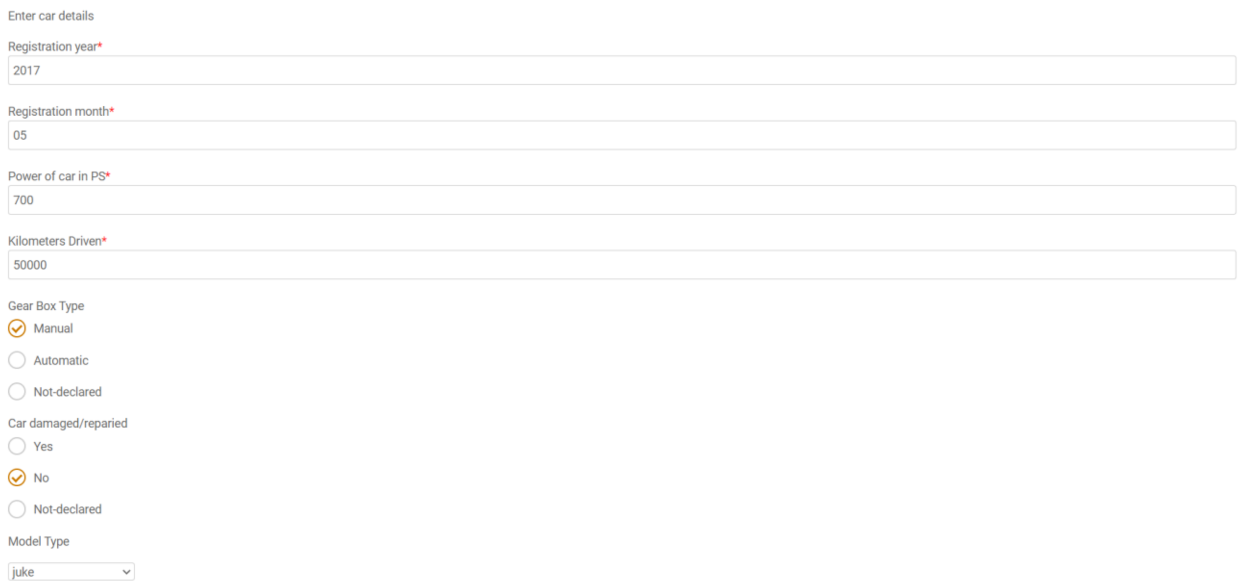
- pyplot
- rcParams
- rainbow

Seaborn: Seaborn is an open-source Python library built on top of matplotlib. It is used for data visualization and exploratory data analysis. Seaborn works easily with dataframes and the Pandas library. The graphs created can also be customized easily.

7.6.2 Data Manipulation

- NumPy: The NumPy library in python is used for scientific computing and array manipulation. It can perform different operations such as indexing of an array, sequencing, and slicing, etc.
- Pandas: The Pandas library in python is used for structuring, manipulating, and organizing data in a tabular structure called the data frame which is further used for data analysis.
- Scikit-learn:
- sklearn.model_selection
- train_test_split
- sklearn.preprocessing
- StandardScaler
- LabelEncoder

Car Resale value Prediction Test 1



Enter car details

Registration year*
2017

Registration month*
05

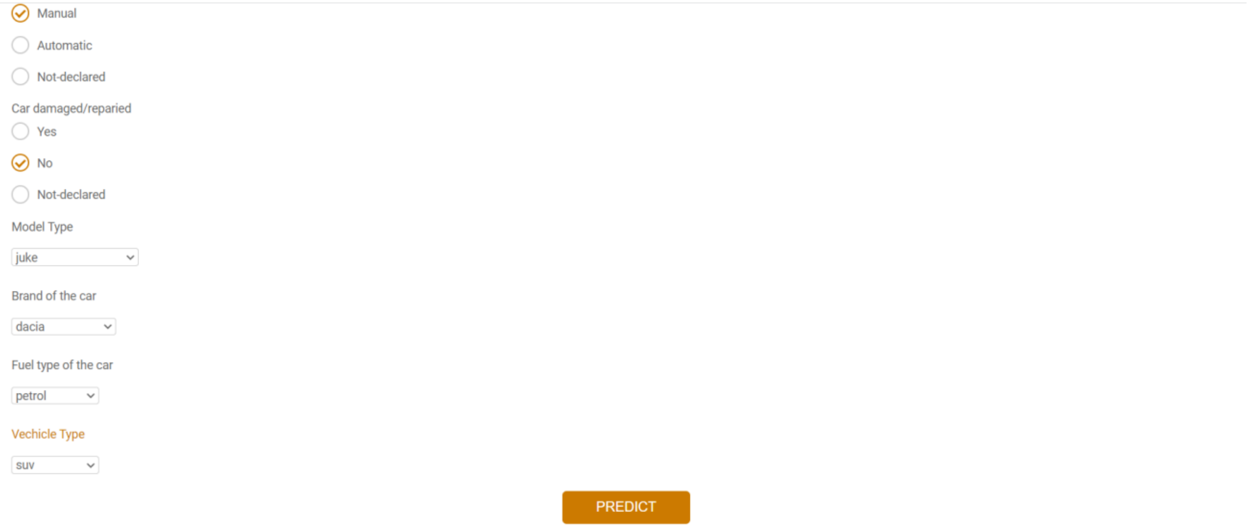
Power of car in PS*
700

Kilometers Driven*
50000

Gear Box Type
● Manual
○ Automatic
○ Not-declared

Car damaged/reparied
○ Yes
● No
○ Not-declared

Model Type
juke



● Manual
○ Automatic
○ Not-declared

Car damaged/reparied
○ Yes
● No
○ Not-declared

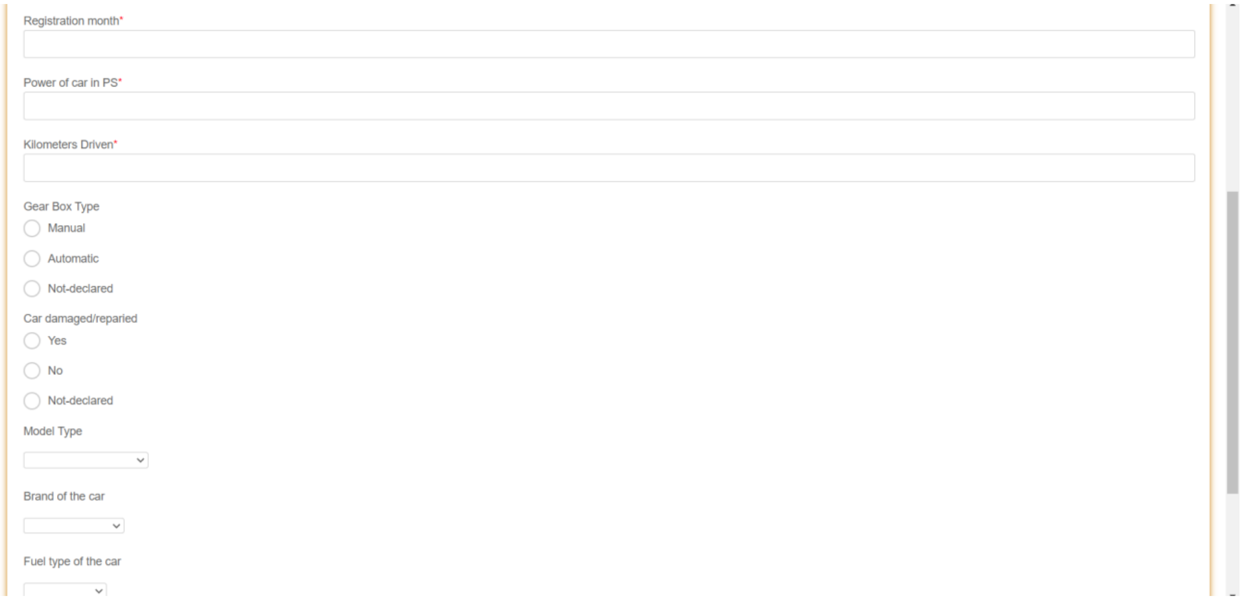
Model Type
juke

Brand of the car
dacia

Fuel type of the car
petrol

Vechicle Type
suv

Car Resale value Prediction Test Result 1



Registration month*

Power of car in PS*

Kilometers Driven*

Gear Box Type

○ Manual

○ Automatic

○ Not-declared

Car damaged/reparied

○ Yes

○ No

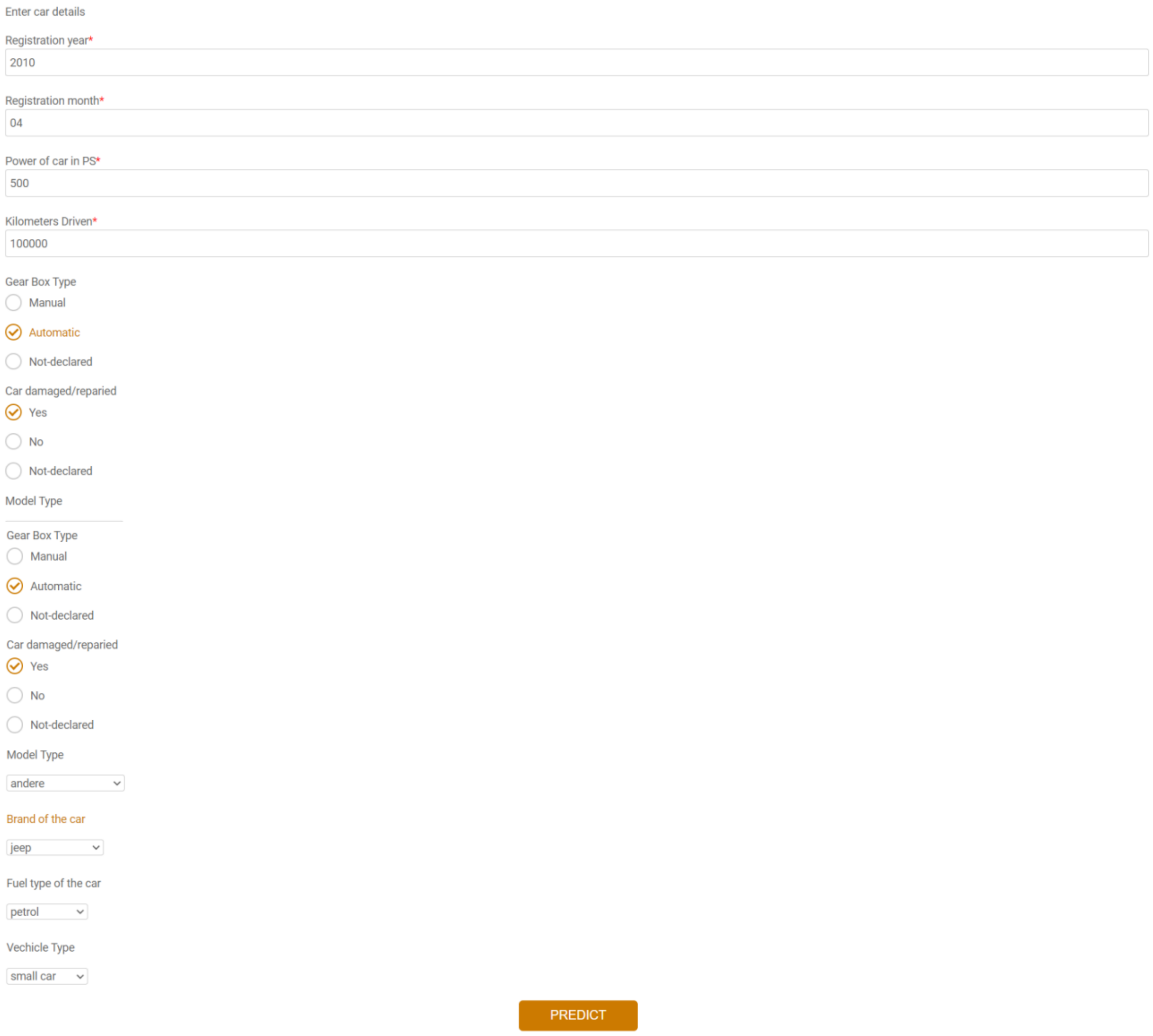
○ Not-declared

Model Type

Brand of the car

Fuel type of the car

Car Resale value Prediction Test 2



Enter car details

Registration year*

2010

Registration month*

04

Power of car in PS*

500

Kilometers Driven*

100000

Gear Box Type

○ Manual

✓ Automatic

○ Not-declared

Car damaged/reparied

✓ Yes

○ No

○ Not-declared

Model Type

Gear Box Type

○ Manual

✓ Automatic

○ Not-declared

Car damaged/reparied

✓ Yes

○ No

○ Not-declared

Model Type

andere

Brand of the car

jeep

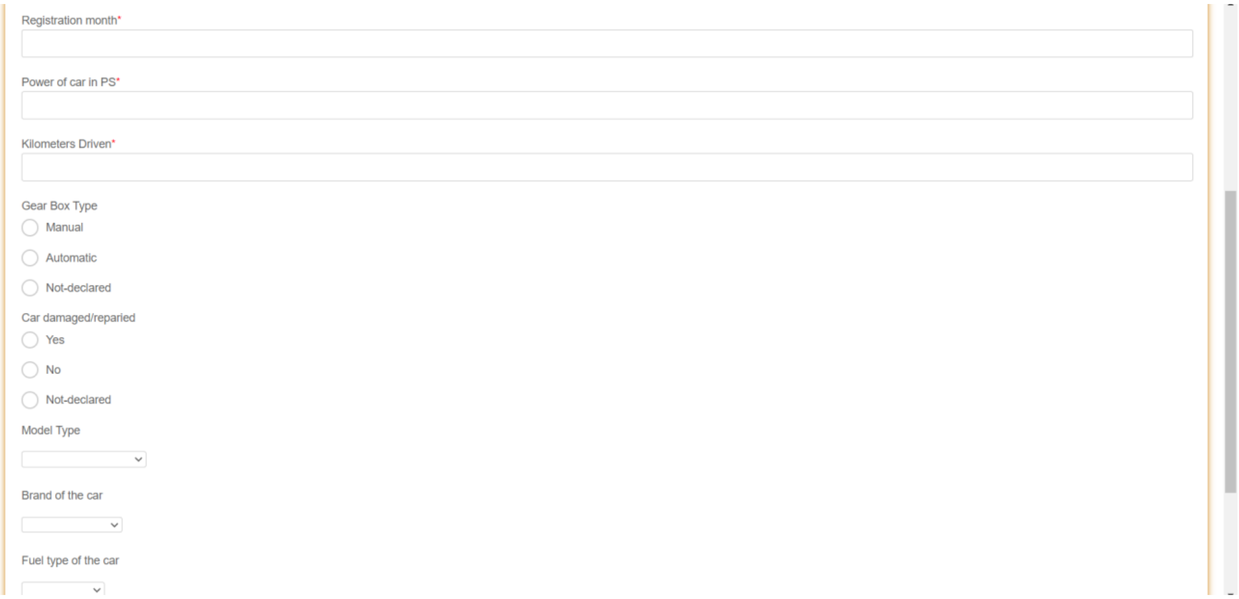
Fuel type of the car

petrol

Vechicle Type

small car

Car Resale value Prediction Test Result 2



Registration month*

Power of car in PS*

Kilometers Driven*

Gear Box Type

○ Manual

○ Automatic

○ Not-declared

Car damaged/reparied

○ Yes

○ No

○ Not-declared

Model Type

Brand of the car

Fuel type of the car

TESTING

CHAPTER-8

TESTING

8. Testing

8.1 Testing and Validations

Validation is a complex process with many possible variations and options, so specifics vary from database to database, but the general outline is:

• Requirement Gathering

o The Sponsor decides what the database is required to do based on regulations,company needs, and any other important factors.
o The requirements are documented and approved.

• System Testing

o Procedures to test the requirements are created and documented.
o The version of the database that will be used for validation is set up.
o The Sponsor approves the test procedures.
o The tests are performed and documented.
o Any needed changes are made. This may require another, shorter round of testing and documentation.

• System Release

o The validation documentation is finalized.
o The database is put into production.

8.2 Testing Levels

8.2.1 Functional Testing:

This type of testing is done against the functional requirements of the project.

Types:

- Unit testing: Each unit /module of the project is individually tested to check for bugs. If any bugs found by the testing team, it is reported to the developer for fixing.
- Integration testing: All the units are now integrated as one single unit and checked for bugs. This also checks if all the modules are working properly with each other.
- System testing: This testing checks for operating system compatibility. It includes both functional and non functional requirements.

- Sanity testing: It ensures change in the code doesn’t affect the working of the project

- Smoke testing: this type of testing is a set of small tests designed for each build.

- Interface testing: Testing of the interface and its proper functioning.

- Regression testing: Testing the software repetitively when a new requirement is added, when bug fixed etc.

- Beta/Acceptance testing: User level testing to obtain user feedback on the product.

8.2.2 Non-Functional Testing:

- This type of testing is mainly concerned with the non-functional requirements such as performance of the system under various scenarios.

- Performance testing: Checks for speed, stability and reliability of the software, hardware or even the network of the system under test.
- Compatibility testing: This type of testing checks for compatibility of the system with different operating systems, different networks etc.

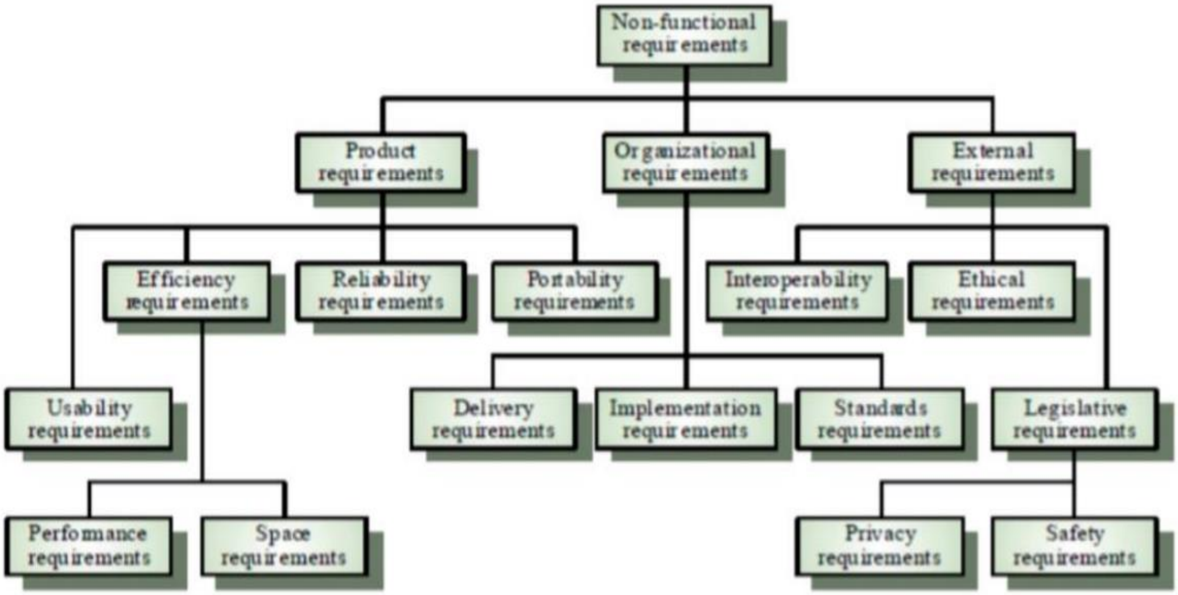
- Localization testing: This checks for the localized version of the product mainly concerned with UI.

- Security testing: Checks if the software has vulnerabilities and if any, fix them. Reliability testing: Checks for the reliability of the software

- Stress testing: This testing checks the performance of the system when it is exposed to different stress levels.

- Usability testing: Type of testing checks the easily the software is being used by the customers

- Compliance testing: Type of testing to determine the compliance of a system with internal or external standards



• Reliability

The structure must be reliable and strong in giving the functionalities. The movements must be made unmistakable by the structure when a customer has revealed a couple of enhancements. The progressions made by the Programmer must be Project pioneer and in addition the Test designer.

• Maintainability

The system watching and upkeep should be fundamental and focus in its approach. There should not be an excess of occupations running on diverse machines such that it gets hard to screen whether the employments are running without lapses.

• Performance

The framework will be utilized by numerous representatives all the while. Since the system will be encouraged on a single web server with a lone database server outside of anyone's ability to see, execution transforms into a significant concern. The structure should not capitulate when various customers would use everything the while. It should allow brisk accessibility to each and every piece of its customers. For instance, if two test specialists are all the while attempting to report the vicinity of a bug, then there ought not to be any irregularity at the same time

• Portability

The framework should to be effectively versatile to another framework. This is obliged when the web server, which s facilitating the framework gets adhered because of a few issues, which requires the framework to be taken to another framework.

• Scalability

The framework should be sufficiently adaptable to include new functionalities at a later stage. There should be a run of the mill channel, which can oblige the new functionalities.

• Flexibility

Flexibility is the capacity of a framework to adjust to changing situations and circumstances, and to adapt to changes to business approaches and rules. An adaptable framework is one that is anything but difficult to reconfigure.

8.3 White Box Testing

White Box Testing is defined as the testing of a software solution's internal structure, design, and coding. In this type of testing, the code is visible to the tester. It focuses primarily on verifying the flow of inputs and outputs through the application, improving design and usability, strengthening security. White box testing is also known as Clear Box testing, Open Box testing, Structural testing, Transparent Box testing, Code-Based testing, and Glass Box testing. It is usually performed by developers.
It is one of two parts of the "Box Testing" approach to software testing. Its counterpart, Blackbox testing, involves testing from an external or enduser type perspective. On the other hand, Whitebox testing is based on the inner workings of an application and revolves around internal testing.
The term "WhiteBox" was used because of the see-through box concept. The clear box or WhiteBox name symbolizes the ability to see through the software's outer shell (or "box") into its inner workings. Likewise, the "black box" in "Black Box Testing" symbolizes not being able to see the inner workings of the software so that only the end-user experience can be tested.

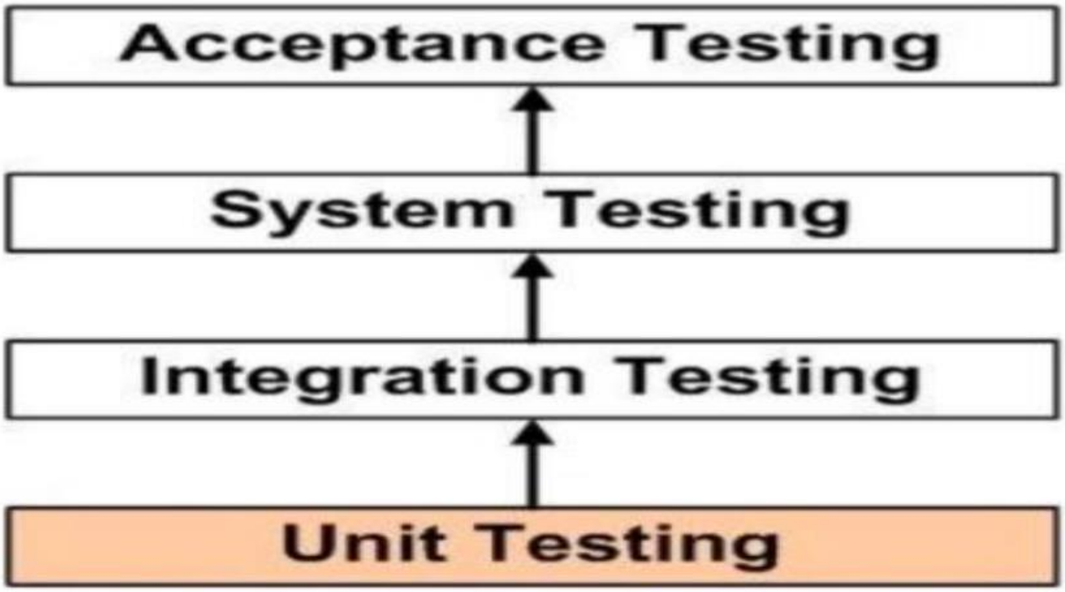
- □ Internal security holes
- □ Broken or poorly structured paths in the coding processes
- □ The flow of specific inputs through the code
- □ Expected output
- □ The functionality of conditional loops
- □ Testing of each statement, object, and function on an individual basis

The testing can be done at system, integration and unit levels of software development. One of the basic goals of whitebox testing is to verify a working flow for an application. It involves testing a series of predefined inputs against expected or desired outputs so that when a specific input does not result in the expected output, you have encountered a bug

8.4 Different Stages of Testing

8.4.1 Unit Testing

UNIT TESTING is a level of software testing where individual units/ components of software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing



Unit Test Plan:

□ Unit Test Plan

o Prepare

o Review

o Rework

o Baseline

□ Unit Test Cases/Scripts

o Prepare

o Review

o Rework

o Baseline

Unit Test

o Perform

Benefits

- Unit testing increases confidence in changing/ maintaining code. If good unit tests are written and if they are run every time any code is changed,
- we will be able to promptly catch any defects introduced due to the change. Also, if codes are already made less interdependent to make unit
- testing possible,the unintended impact of changes to any code is less.
- Codes are more reusable. In order to make unit testing possible, codes need to be modular. This means that codes are easier to reuse.
- Development is faster. How? If you do not have unit testing in place, you write your code and perform that fuzzy ‘developer test’ (You set some
- breakpoints, fire up the GUI, provide a few inputs that hopefully hit your code and hope that you are all set.) But, if you have unit testing in place,
- you write the test, write the code and run the test. Writing tests takes time but the time is compensated by the less amount of time it takes to run
- the tests;

- You need not fire up the GUI and provide all those inputs. And, of course, unit tests are more reliable than ‘developer tests’. Development is faster
- in the long run too. How? The effort required to find and fix defects found during unit testing is very less in comparison to the effort required to
- fix defects found during system testing or acceptance testing.
- The cost of fixing a defect detected during unit testing is lesser in comparison to that of defects detected at higher levels. Compare the cost
- (time, effort, destruction, humiliation) of a defect detected during acceptance testing or when the software is live.
- Debugging is easy. When a test fails, only the latest changes need to be debugged. With testing at higher levels, changes made over the span of
- several days/weeks/months need to be scanned
- 8.4.2 **Integration Testing**
- INTEGRATION TESTING is a level of software testing where individual units are combined and tested as a group. The purpose of this level of
- testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.
- Integration testing: Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems. See
- also component integration testing, system integration testing.
- Component integration testing: Testing performed to expose defects in the interfaces and interaction between integrated components.
- System integration testing: Testing the integration of systems and packages; testing interfaces to external organizations (e.g. Electronic Data
- Interchange,Internet).

Tasks
Integration Test Plan
o Prepare
o Review
o Rework
o Baseline
Integration Test Cases/Scripts
o Prepare
o Review
o Rework
o Baseline
Integration Test

8.4.3 System Testing

SYSTEM TESTING is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the
system’s compliance with the specified requirements.
system testing: The process of testing an integrated system to verify that it meets specified requirements.

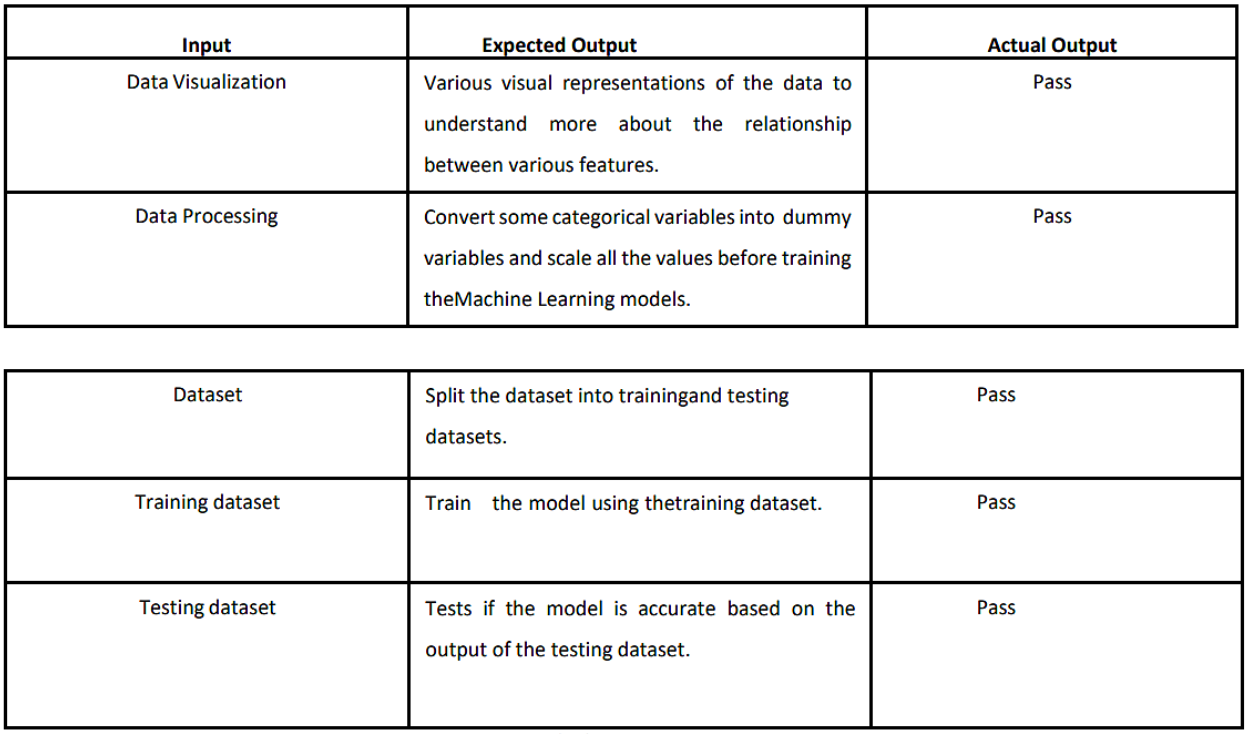
8.4.4 Acceptance Testing

ACCEPTANCE TESTING is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the
system’s compliance with the business requirements and assess whether it is acceptable for delivery.
acceptance testing: Formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a

system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system.

TESTING CAR RESALE VLAUE PRIDICTION:

- Testing is the process used to help identify the correctness, completeness, security and quality of the developed computer software. Testing is the
- process of technical investigation and includes the process of executing a program or application with the intent of finding errors.
- In the training process, our model learns to associate a particular input (i.e. features) to the corresponding output (tag) based on the test samples
- used for training. Input features and tags (e.g. 1-normal 2-heart disease) are fed into the machine learning algorithm to generate a model.
- A comparative analysis of different classifiers was performed for the classification of the Heart Disease dataset in order to correctly classify and
- predict Heart Disease cases with minimal attributes.



Input	Expected Output	Actual Output
Data Visualization	Various visual representations of the data to understand more about the relationship between various features.	Pass
Data Processing	Convert some categorical variables into dummy variables and scale all the values before training theMachine Learning models.	Pass
Dataset	Split the dataset into trainingand testing datasets.	Pass
Training dataset	Train the model using thetraining dataset.	Pass
Testing dataset	Tests if the model is accurate based on the output of the testing dataset.	Pass

Training and subsequent testing

8.5 Model Evaluation

The most important evaluation metrics for this problem domain are Accuracy, Sensitivity, Specificity, Precision, F1-measure, Log Loss, ROC and Mathew correlation coefficient.

Precision: which is how consistent results are when measurements are repeated and can be calculated using the following formula:

Precision = True Positive / (True Positive + False Positive)

Sensitivity: Sensitivity is a measure of the proportion of actual positive cases that gotpredicted as positive (or true positive). Sensitivity is also termed as Recall.

Sensitivity = True Positive / (True Positive + False Negative

Specificity: Specificity is defined as the proportion of actual negatives, which got predicted asthe negative (or true negative).

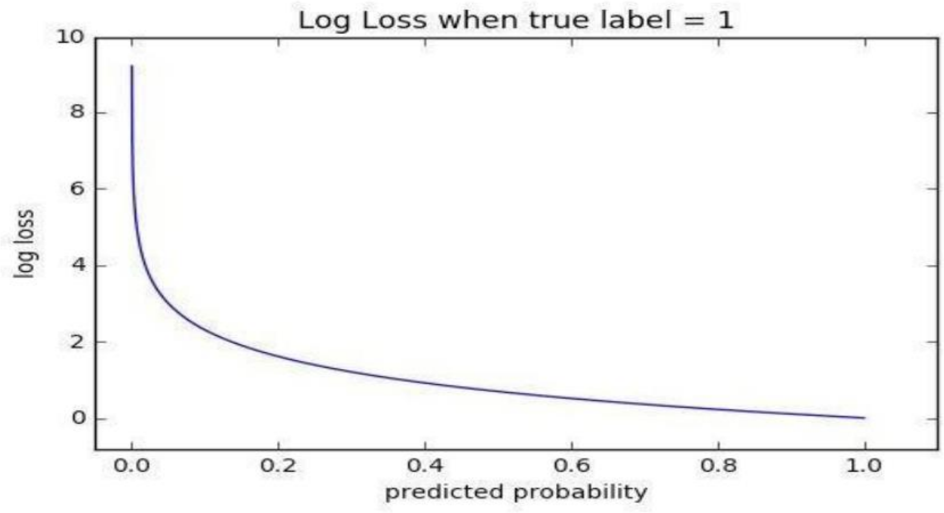
Specificity = True Negative / (True Negative + False Positive)

Mathew Correlation coefficient (MCC):

The Matthews correlation coefficient (MCC), instead, is a more reliable statisticalrate which produces a high score only if the prediction obtained good results in all ofthe four confusion matrix categories (true positives, false negatives, true negatives, and false positives), proportionally both to the size of positive elements and the size of negative elements in the dataset.

Logic loss:

Logarithmic loss measures the performance of a classification model where the prediction input is a probability value between 0 and 1. The goal of our machine learning models is to minimize this value. A perfect model would have a log loss of 0. Log loss increases as the predicted probability diverges from the actual label. So, predicting a probability of .012 when the actual observation label is 1 would be bad and result in a high log loss.



Log Loss Graph

F1 Score:

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 score is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost.

F1 Score = 2(Recall Precision) / (Recall + Precision)

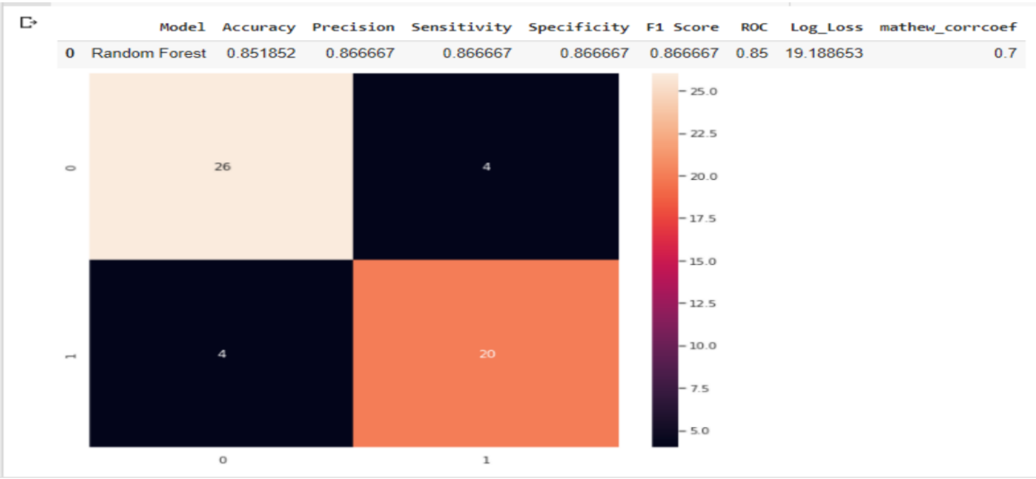
ROC Curve

An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters: True Positive Rate & False Positive Rate.

8.5.1 Random Forest Classifier

Code:

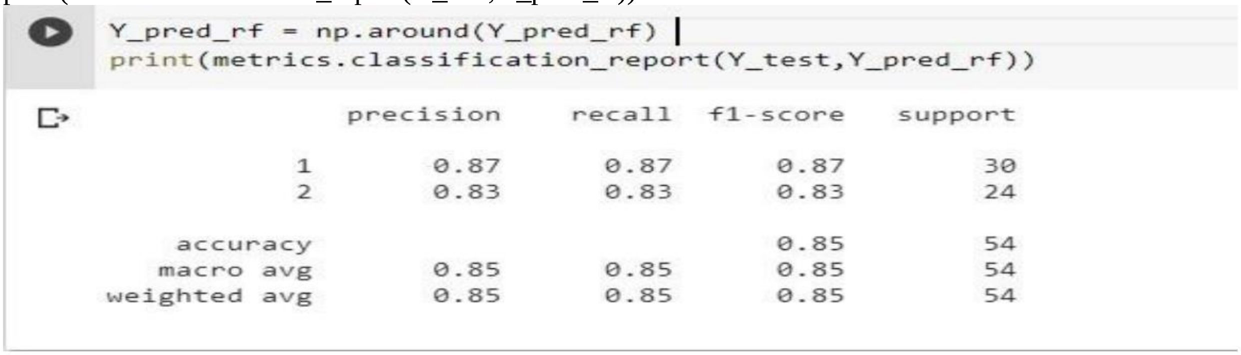
```
y_pred_rfe = rf_classifier.predict(X_test)
plt.figure(figsize=(10, 8)) CM=confusion_matrix(Y_test,y_pred_rfe) sns.heatmap(CM, annot=True)
TN = CM[0][0]
FN = CM[1][0]
TP = CM[1][1]
FP = CM[0][1]
specificity = TN/(TN+FP)
loss_log = log_loss(Y_test, y_pred_rfe) acc= accuracy_score(Y_test, y_pred_rfe)
roc=roc_auc_score(Y_test, y_pred_rfe)
prec = precision_score(Y_test, y_pred_rfe) rec = recall_score(Y_test, y_pred_rfe)
f1 = f1_score(Y_test, y_pred_rfe)
mathew = matthews_corrcoef(Y_test, y_pred_rfe)
model_results =pd.DataFrame([['Random Forest',acc, prec,rec,specificity, f1,roc, loss_log,mathew]],
columns = ['Model', 'Accuracy','Precision', 'Sensitivity','Specificity', 'F1
Score','ROC','Log_Loss','mathew_corrcoef'])
model_results
```



	Model	Accuracy	Precision	Sensitivity	Specificity	F1 Score	ROC	Log_Loss	mathew_corrcoef
0	Random Forest	0.851852	0.866667	0.866667	0.866667	0.866667	0.85	19.188653	0.7

Random Forest Confusion Matrix

```
Y_pred_rf = np.around(Y_pred_rf)
print(metrics.classification_report(Y_test,Y_pred_rf))
```

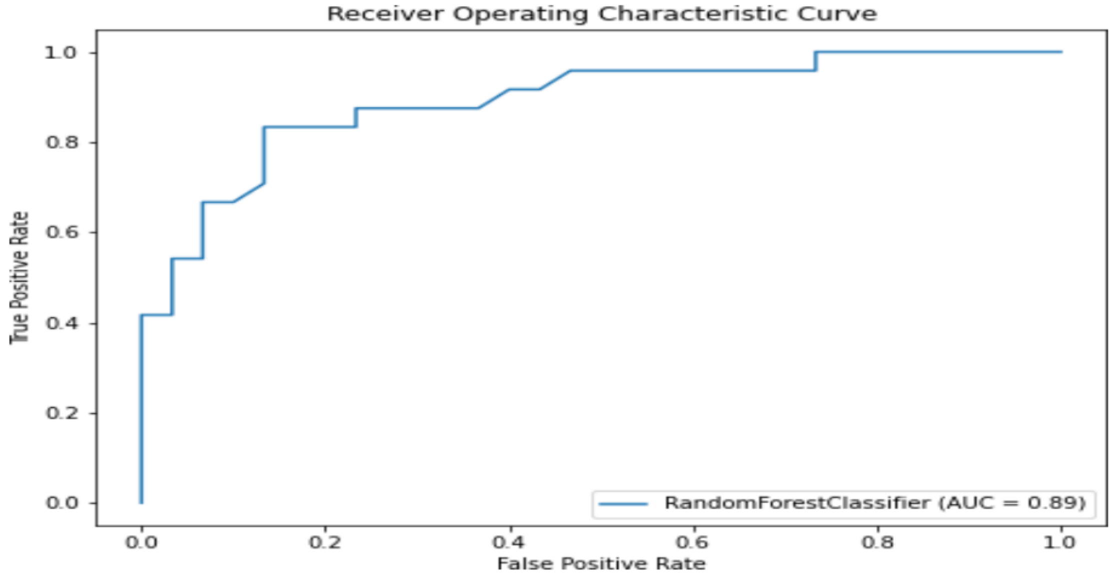


```
Y_pred_rf = np.around(Y_pred_rf)
print(metrics.classification_report(Y_test,Y_pred_rf))
```

```
              precision    recall  f1-score   support

           1       0.87      0.87      0.87        30
           2       0.83      0.83      0.83        24

    accuracy                           0.85        54
   macro avg       0.85      0.85      0.85        54
weighted avg       0.85      0.85      0.85        54
```

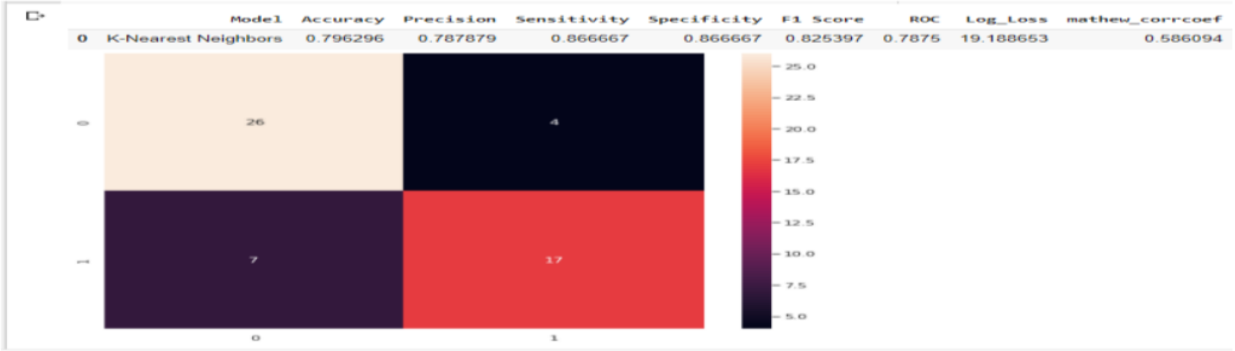
Random Forest Classification Report

```
plot_roc_curve(rf_classifier,X_test,Y_test)
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic Curve')
```



8.5.2 K-Nearest Neighbors

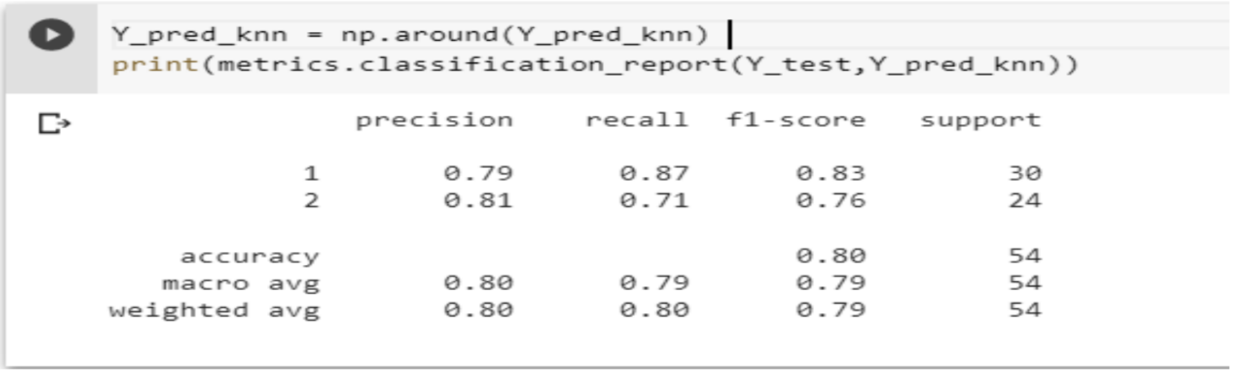
```
 Classifier y_pred_knne = knn_classifier.predict(X_test)
plt.figure(figsize=(10, 8)) CM=confusion_matrix(Y_test,y_pred_knne) sns.heatmap(CM,
annot=True)
TN = CM[0][0]
FN = CM[1][0]
TP = CM[1][1]
FP = CM[0][1]
specificity = TN/(TN+FP)
loss_log = log_loss(Y_test, y_pred_knne) acc= accuracy_score(Y_test, y_pred_knne)
roc=roc_auc_score(Y_test,
y_pred_knne)
prec = precision_score(Y_test, y_pred_knne) rec = recall_score(Y_test, y_pred_knne)
f1 = f1_score(Y_test, y_pred_knne)
mathew = matthews_corrcoef(Y_test, y_pred_knne)
model_results =pd.DataFrame([['K-Nearest Neighbors ',acc, prec,rec,specificity, f1,roc,
loss_log,mathew]],
columns = ['Model', 'Accuracy','Precision', 'Sensitivity','Specificity', 'F1
Score','ROC','Log_Loss','mathew_corrcoef'])
model_results
```



	Model	Accuracy	Precision	Sensitivity	Specificity	F1 Score	ROC	Log_Loss	mathew_corrcoef
0	K-Nearest Neighbors	0.796296	0.787879	0.866667	0.866667	0.825397	0.7875	19.188653	0.586094

K-Nearest Neighbors Confusion Matrix

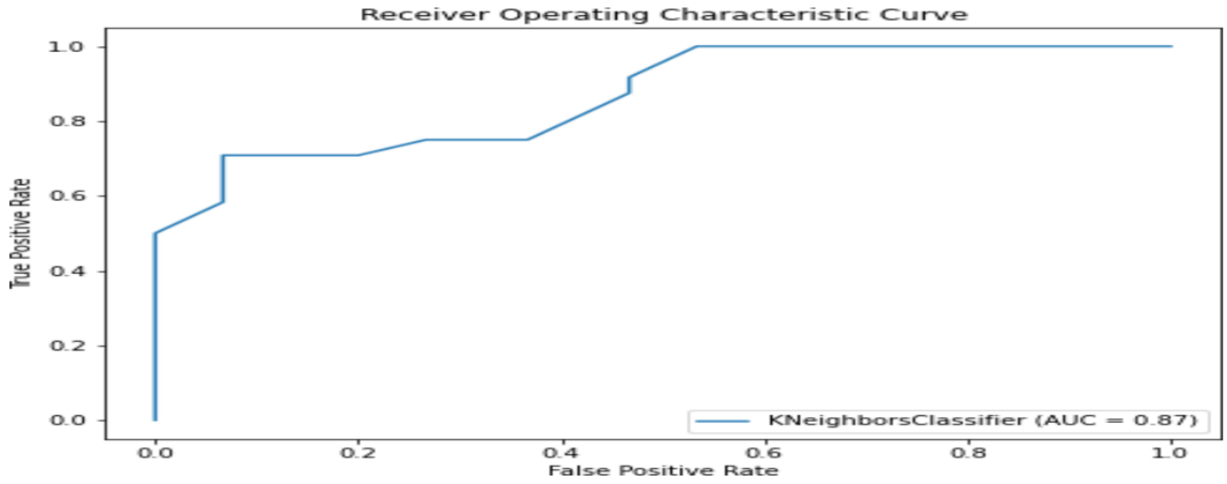
Y_pred_knn = np.around(Y_pred_knn)
 print(metrics.classification_report(Y_test,Y_pred_knn))



```
Y_pred_knn = np.around(Y_pred_knn)
print(metrics.classification_report(Y_test,Y_pred_knn))

              precision    recall  f1-score   support

           1       0.79      0.87      0.83        30
           2       0.81      0.71      0.76        24

    accuracy                           0.80        54
   macro avg       0.80      0.79      0.79        54
weighted avg       0.80      0.80      0.79        54
```

K-Nearest Neighbors Classification Report

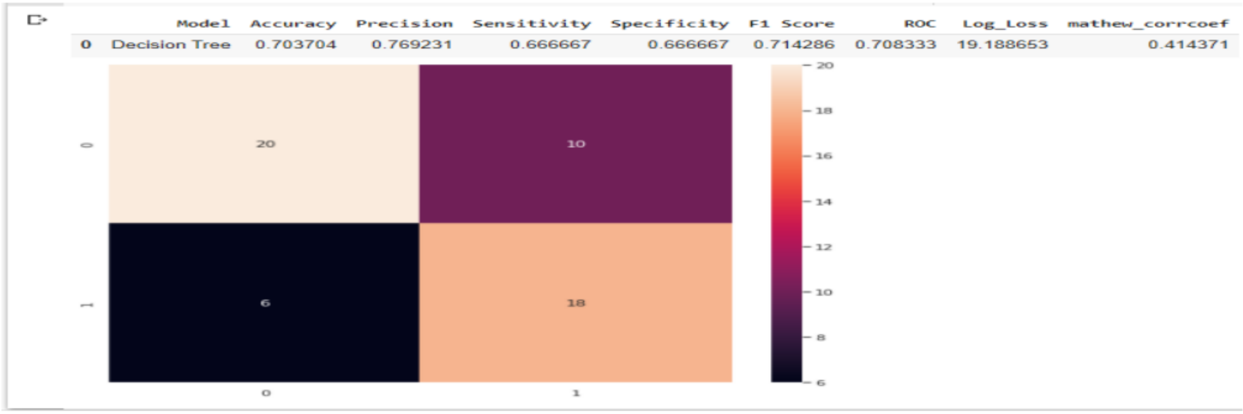
plot_roc_curve(knn_classifier,X_test,Y_test)plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic Curve')



K-Nearest Neighbors ROC Curve

8.5.3 Decision Tree Classifier:

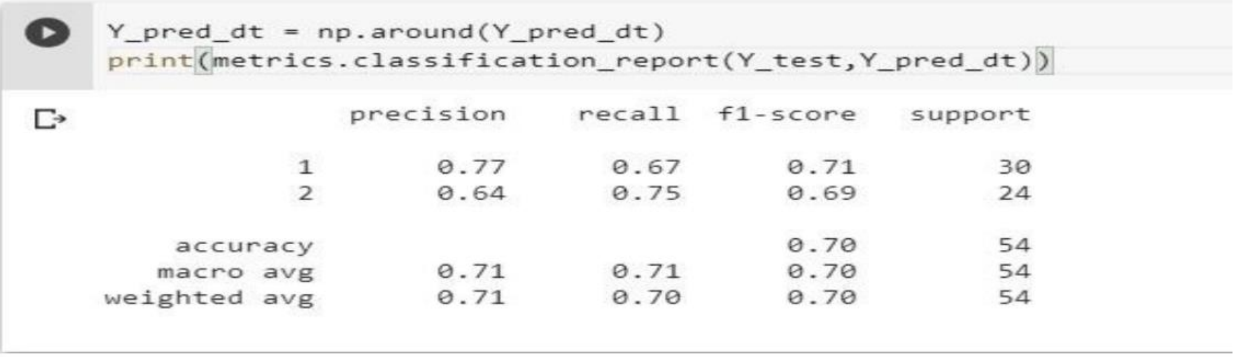
```
y_pred_dte = dt_classifier.predict(X_test)
plt.figure(figsize=(10, 8)) CM=confusion_matrix(Y_test,y_pred_dte) sns.heatmap(CM, annot=True)
TN = CM[0][0]
FN = CM[1][0]
TP = CM[1][1]
FP = CM[0][1]
specificity = TN/(TN+FP)
loss_log = log_loss(Y_test, y_pred_dte) acc= accuracy_score(Y_test, y_pred_dte)
roc=roc_auc_score(Y_test,
y_pred_dte)
prec = precision_score(Y_test, y_pred_dte) rec = recall_score(Y_test, y_pred_dte)
f1 = f1_score(Y_test, y_pred_dte)
mathew = matthews_corrcoef(Y_test, y_pred_dte)
model_results =pd.DataFrame([['Decision Tree',acc, prec,rec,specificity, f1,roc, loss_log,mathew]],
columns = ['Model', 'Accuracy','Precision', 'Sensitivity','Specificity', 'F1
Score','ROC','Log_Loss','mathew_corrcoef'])
model_results
```



	Model	Accuracy	Precision	Sensitivity	Specificity	F1 Score	ROC	Log_Loss	mathew_corrcoef
0	Decision Tree	0.703704	0.769231	0.666667	0.666667	0.714286	0.708333	19.188653	0.414371

Decision Tree Confusion Matrix

```
Y_pred_dt = np.around(Y_pred_dt)
 print(metrics.classification_report(Y_test,Y_pred_dt))
```



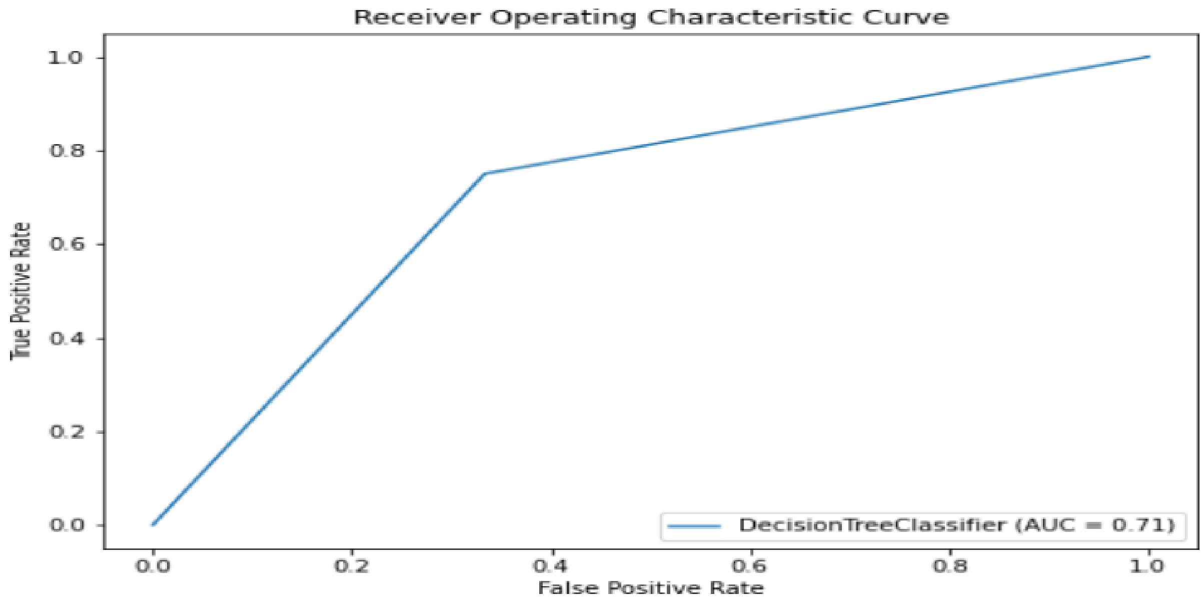
```
Y_pred_dt = np.around(Y_pred_dt)
print(metrics.classification_report(Y_test,Y_pred_dt))
```

```
              precision    recall  f1-score   support

           1       0.77      0.67      0.71        30
           2       0.64      0.75      0.69        24

    accuracy                           0.70        54
   macro avg       0.71      0.71      0.70        54
weighted avg       0.71      0.70      0.70        54
```

Decision Tree Classification Report

```
plot_roc_curve(dt_classifier,X_test,Y_test)
```

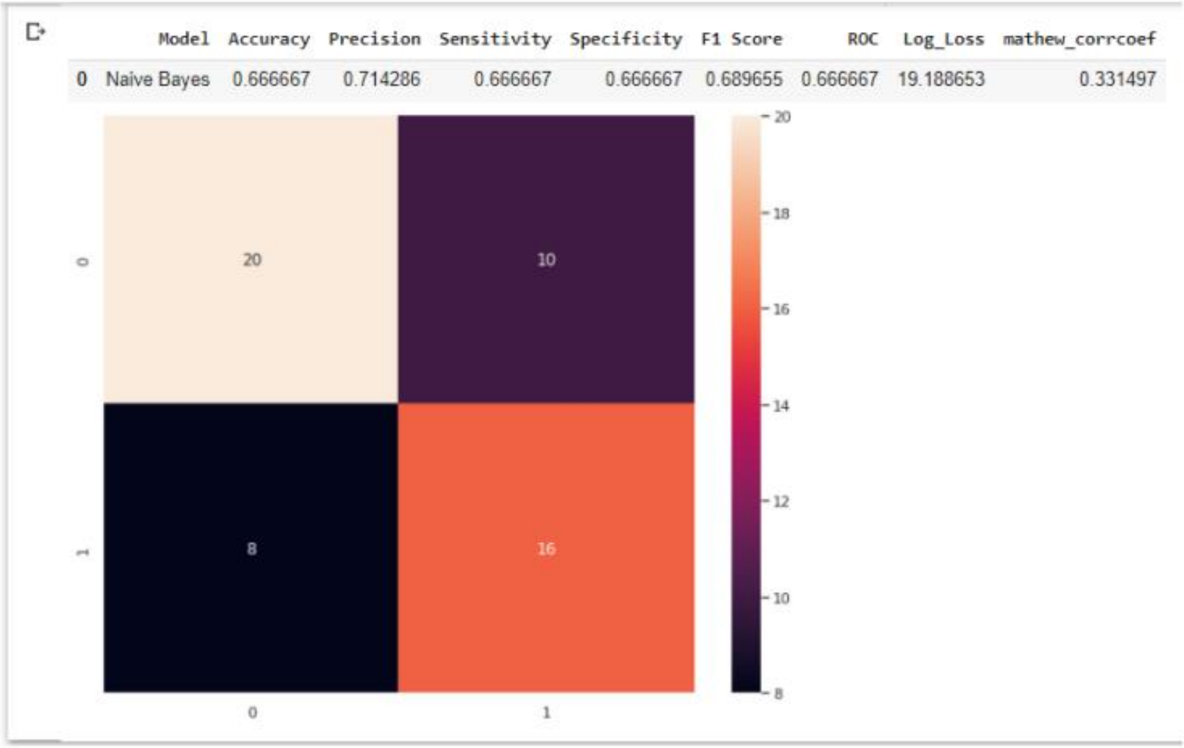
```
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic Curve')
```



Decision Tree ROC Curve

8.5.4 Naive Bayes Classifier

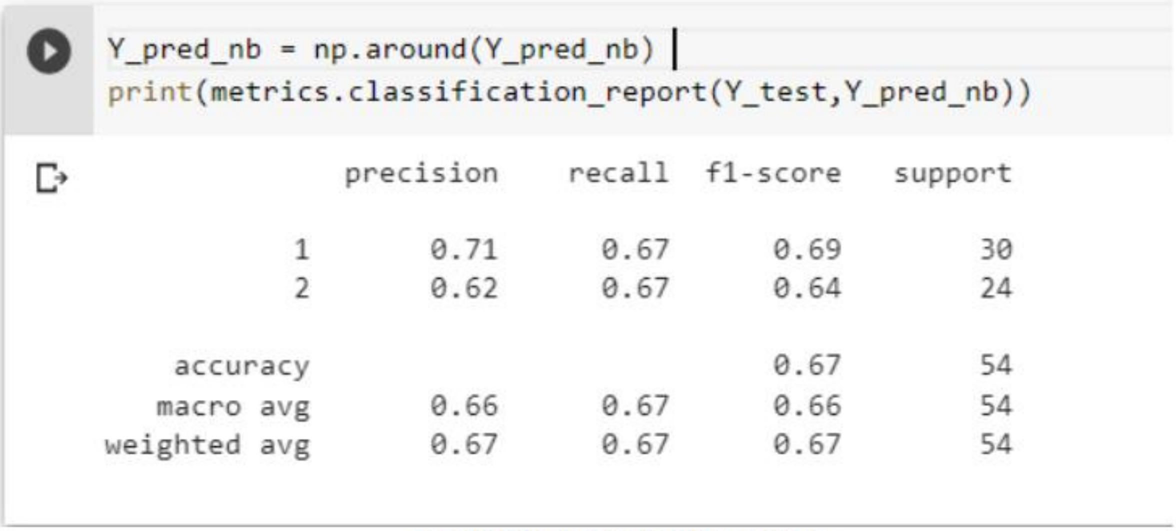
```
y_pred_nbe = nb_classifier.predict(X_test)
plt.figure(figsize=(10, 8)) CM=confusion_matrix(Y_test,y_pred_nbe) sns.heatmap(CM,
annot=True)
TN = CM[0][0]
FN = CM[1][0]
TP = CM[1][1]
FP = CM[0][1]
specificity = TN/(TN+FP)
loss_log = log_loss(Y_test, y_pred_nbe) acc= accuracy_score(Y_test, y_pred_nbe)
roc=roc_auc_score(Y_test, y_pred_nbe)
prec = precision_score(Y_test, y_pred_nbe) rec = recall_score(Y_test, y_pred_nbe)
f1 = f1_score(Y_test, y_pred_nbe)
mathew = matthews_corrcoef(Y_test, y_pred_nbe)
model_results =pd.DataFrame([['Naive Bayes ',acc, prec,rec,specificity, f1,roc,
loss_log,mathew]],
columns = ['Model', 'Accuracy','Precision', 'Sensitivity','Specificity', 'F1
Score','ROC','Log_Loss','mathew_corrcoef'])
model_results
```



	Model	Accuracy	Precision	Sensitivity	Specificity	F1 Score	ROC	Log_Loss	mathew_corrcoef
0	Naive Bayes	0.666667	0.714286	0.666667	0.666667	0.689655	0.666667	19.188653	0.331497

Naive Bayes Confusion Matrix

Y_pred_nb = np.around(Y_pred_nb)
print(metrics.classification_report(Y_test,Y_pred_nb))

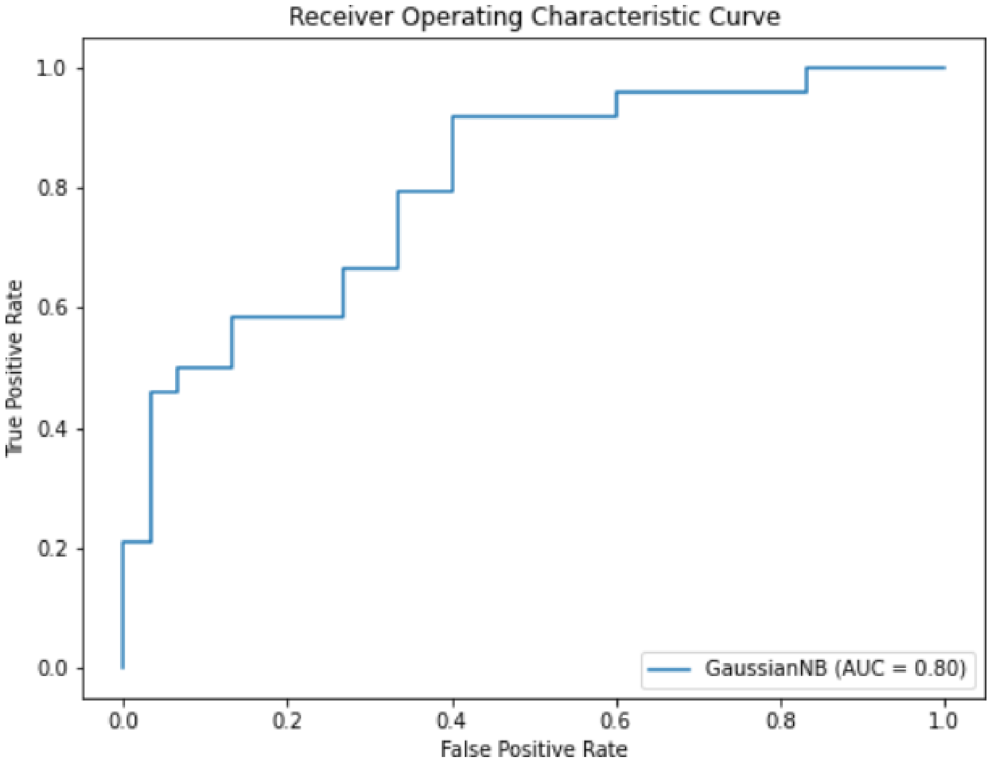


```
Y_pred_nb = np.around(Y_pred_nb)
print(metrics.classification_report(Y_test,Y_pred_nb))
```

```
              precision    recall  f1-score   support

           1       0.71      0.67      0.69        30
           2       0.62      0.67      0.64        24

    accuracy                           0.67        54
   macro avg       0.66      0.67      0.66        54
weighted avg       0.67      0.67      0.67        54
```

Naive Bayes Classification Report

plot_roc_curve(nb_classifier,X_test,Y_test)
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic Curve')



Naive Bayes ROC Curve

RESULTS

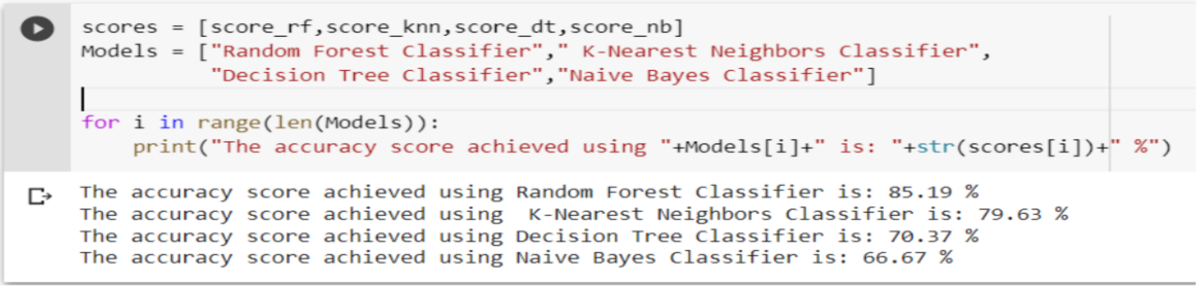
CHAPTER-9
RESULTS

9.1 Performance Metrics

Final Result:

Model	Accuracy	precision	Sensitivity	Specificity	F1 Score	ROC	Log_loss	Mathew_ correcoef
Random Forest	0.8519	0.8667	0.8667	0.8667	0.8667	0.85	19.1886	0.7
KNN	0.7963	0.7963	0.7879	0.8667	0.8254	0.7875	19.1886	0.5861
Decision Tree	0.7037	0.7692	0.6667	0.6667	0.7142	0.7083	19.1886	0.4144
Naive Bayes	0.6667	0.7143	0.6667	0.6667	0.6896	0.6667	19.1886	0.3315

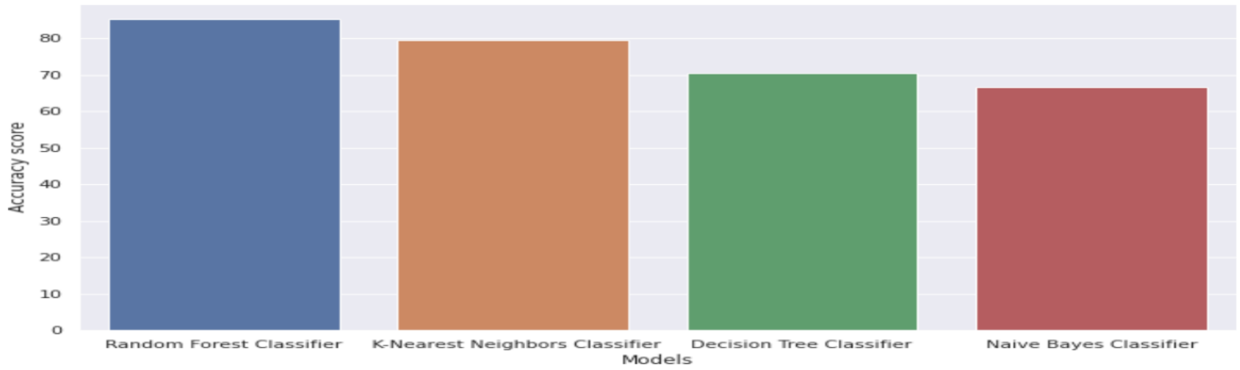
Final Accuracy Score:



```
scores = [score_rf,score_knn,score_dt,score_nb]
Models = ["Random Forest Classifier"," K-Nearest Neighbors Classifier",
          "Decision Tree Classifier","Naive Bayes Classifier"]

for i in range(len(Models)):
    print("The accuracy score achieved using "+Models[i]+" is: "+str(scores[i])+" %")
```

```
The accuracy score achieved using Random Forest Classifier is: 85.19 %
The accuracy score achieved using  K-Nearest Neighbors Classifier is: 79.63 %
The accuracy score achieved using Decision Tree Classifier is: 70.37 %
The accuracy score achieved using Naive Bayes Classifier is: 66.67 %
```

Accuracy Score Bar Graph:



CHAPTER-10
ADVANTAGES & DISADVANTAGES

10.1 ADVANTAGES

- Good at learning complex and non-linear relationships
- Highly explainable and easy to interpret
- Robust to outliers
- No feature scaling is required

10.2 DISADVANTAGES

- Consumes more time
- Requires high computational power

CHAPTER-11
CONCLUSION

11.1 CONCLUSION

- Determining whether the listed price of a used car is a challenging task, due to the many factors that drive a used vehicle’s price on the market.
- The focus of this project is developing machine learning models that can accurately predict the price of a used car based on its features, in order to make informed purchases.
- By performing ML models, we aim to get a better result or less error with max accuracy to predict the value of the used car. Initially, data cleaning is performed to remove the null values and outliers from the dataset then ML models are implemented to predict the price of cars.
- Next, with the help of data visualization features were explored deeply. The relation between the features is examined.
- From the report, it can be said that gradient regression regressor is the best model for the prediction for used car prices

CHAPTER-12

FUTURE SCOPE

12.1 FUTURE SCOPE

- In the future, more data will be collected using different web-scraping techniques, and deep learningclassifiers will be tested.
- Algorithms like Quantile Regression, ANN and SVM will be tested.Afterwards, the intelligent model will be integrated with web and mobile-based applications for public use.
- Moreover, after the data collection phase Semiconductor shortages have incurred after the pandemic which led to an increase in car prices, and greatly affected the secondhand market.
- Hence having a regular Datacollection and analysis is required periodically, ideally, we would be having a real time processing program.

CHAPTER-13
APPENDIX

13. APPENDIX

Python:

- Python is an interpreted, high-level, general purpose programming language created by Guido Van Rossum and first released in 1991, Python's design philosophy emphasizes code Readability with its notable use of significant White space.
- Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.
- Python is dynamically typed and garbage collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

Sklearn:

- Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python.
- It provides a selection of efficient tools for machine learning and statistical modelling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python.
- This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

Numpy:

- NumPy is a library for the python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
- The ancestor of NumPy, Numeric, was originally created by Jim with contributions from several other developers. In 2005, Travis created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications.
- NumPy is open-source software and has many contributors.

Librosa:

- Librosa is a Python package for music and audio analysis. Librosa is basically used when we work with audio data like in music generation (using LSTMs), Automatic Speech Recognition.
- It provides the building blocks necessary to create the music information retrieval systems. Librosa helps to visualize the audio signals and also do the feature extractions in it using different signal processing techniques.

Matplotlib:

- 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 statemachine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged.

Seaborn:

- Seaborn is a Python data visualization library based on matplotlib. It provides a highlevel interface for drawing attractive and informative statistical graphics.

- Seaborn is a library in Python predominantly used for making statistical graphics. Seaborn is a data visualization library built on top of matplotlib and closely integrated with pandas data structures in Python.
- Visualization is the central part of Seaborn which helps in exploration and understanding of data**.**

SciPy:

- SciPy contains modules for optimization, linearalgebra, integration, interpolation, special functions, FFT, signal and imageprocessing, ODE solvers and other tasks common in science and engineering.
- SciPy is also a family of conferences for users and developers of these tools: SciPy (in the United States), EuroSciPy (in Europe) and SciPy.in (in India).
- Enthought originated the SciPy conference in the United States and continues to sponsor many of the international conferences as well as host the SciPy website.
- SciPy is a scientific computation library that uses NumPy underneath. It provides more utility functions for optimization, stats and signal processing.

Visualizing and Predicting Heart Diseases with an Interactive Dashboardsource code:

```
from flask import Flask, render_template, request

import requests
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
API_KEY = "7FGNzBcGOMBIR-LTevrXYzj50iLFF2lJw2jYIkuus7wn"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token',
data={"apikey":
 API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

app = Flask(__name__)# interface between my server and my application wsgi


@app.route('/')#binds to an url
def index():
    return render_template('home.html')

@app.route('/predict')
def predict():
    return render_template('predict.html')
@app.route('/y_predict', methods= ['POST'])
def y_predict():
    regyear = int(request.form['Registrationyear'])
    powerps = int(request.form['PowerofcarinPS'])
    kms =int(request.form['KilometersDriven'])
    regmonth =int(request.form.get('Registrationmonth'))
    gearbox =(request.form['Geartype'])
    damage =request.form['cd']
    model = request.form.get('model')
    brand =request.form.get('brand')
    fuelType =request.form.get('fueltype')
    vehicletype = request.form.get('vechicletype')
    row = {'vehicleType': vehicletype,'yearOfRegistration':regyear,
           'gearbox': gearbox,'powerPS': powerps, 'model':model,'kilometer': kms,
           'monthOfRegistration': regmonth,'fuelType': fuelType,
           'brand': brand,'notRepairedDamage': damage}
    print(row)
    new_row = pd.DataFrame([row])
    new_df = pd.DataFrame(columns = ['vehicleType', 'yearOfRegistration',
'gearbox',
                           'powerPS','model', 'kilometer', 'monthOfRegistration', 'fuelType',
                           'brand','notRepairedDamage' ])
    new_df=pd.concat([new_df,new_row], ignore_index = True)
    new_df['monthOfRegistration']= new_df['monthOfRegistration'].astype(int)
    labels = ['gearbox', 'notRepairedDamage', 'model', 'brand', 'fuelType',
'vehicleType']
    mapper = {}
    for i in labels:
        mapper[i] = LabelEncoder()
        mapper[i].fit(new_df[i])
        mapper[i].classes_ = np.load(str('classes'+i+'.npy'),allow_pickle=True)
        tr = mapper[i].fit_transform(new_df[i])
        new_df.loc[:, i + '_labels'] = pd.Series (tr, index=new_df.index)
    labeled = new_df[ ['yearOfRegistration',
'powerPS','kilometer','monthOfRegistration']
        + [x+'_labels' for x in labels]]
    X = labeled.values
    X= X.tolist()
    print(X)
    payload_scoring = {"input_data": [{"field": ['f0', 'f1', 'f2',
                         'f3','f4', 'f5', 'f6', 'f7',
                         'f8','f9' ],
                                     "values":X }]}
    #payload_scoring = {"input_data": [{"fields": [array_of_input_fields], "values":
[array_of_values_to_be_scored, another_array_of_values_to_be_scored]}]}
    response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/2e6b4079-fdd3-4b9b-8427-
f309d0af6b20/predictions?version=2022-11-16', json=payload_scoring,
    headers={'Authorization': 'Bearer ' + mltoken})
    print("Scoring response")


    predictions=response_scoring.json()
    df_ev = np.exp(predictions['predictions'][0]['values'][0][0])
    print(df_ev)

    return render_template('predict.html',y = 'The resale value predicted is
{:.2f}$'.format(df_ev))

if __name__=='__main__':
    app.run(debug= False)
```

Web App Codefor ipynb file to flask framework view Visualizing and Predicting HeartDiseases with an Interactive Dashboard:

app_flask.py

```
import pandas
from flask import Flask, render_template, request

import pandas as pd
import numpy as np


from sklearn.preprocessing import LabelEncoder
```

```
app =Flask(__name__)
import pickle
filename = 'resale_model.sav'
model_rand= pickle.load(open(filename, 'rb'))
@app.route('/')
def index():
   return render_template('home.html')

@app.route('/predict')
def predict():
   return render_template('predict.html')
@app.route('/y_predict', methods= ['POST'])
def y_predict():
   regyear = int(request.form['Registrationyear'])
   powerps = int(request.form['PowerofcarinPS'])
   kms =int(request.form['KilometersDriven'])
   regmonth =int(request.form.get('Registrationmonth'))
   gearbox =(request.form['Geartype'])
   damage =request.form['cd']
   model = request.form.get('model')
   brand =request.form.get('brand')
   fuelType =request.form.get('fueltype')
   vehicletype = request.form.get('vechicletype')
   row = {'vehicleType': vehicletype,'yearOfRegistration':regyear,
        'gearbox': gearbox,'powerPS': powerps, 'model':model,'kilometer': kms,
        'monthOfRegistration': regmonth,'fuelType': fuelType,
        'brand': brand,'notRepairedDamage': damage}
   print(row)
   new_row = pd.DataFrame([row])
   new_df = pd.DataFrame(columns = ['vehicleType', 'yearOfRegistration',
'gearbox',
                    'powerPS','model', 'kilometer', 'monthOfRegistration', 'fuelType',
                    'brand','notRepairedDamage' ])
   new_df=pd.concat([new_df,new_row], ignore_index = True)
   new_df['monthOfRegistration']= new_df['monthOfRegistration'].astype(int)
   labels = ['gearbox', 'notRepairedDamage', 'model', 'brand', 'fuelType',
'vehicleType']
   mapper = {}
   for i in labels:
       mapper[i] = LabelEncoder()
       mapper[i].fit(new_df[i])
       mapper[i].classes_ = np.load(str('classes'+i+'.npy'),allow_pickle=True)
       tr = mapper[i].fit_transform(new_df[i])
       new_df.loc[:, i + '_labels'] = pd.Series (tr, index=new_df.index)
   labeled = new_df[ ['yearOfRegistration',
'powerPS','kilometer','monthOfRegistration']
       + [x+'_labels' for x in labels]]
   X = labeled.values
   print(X)
   y_prediction = model_rand.predict(X)
   print(y_prediction)
   df_ev = np.exp(y_prediction)
   print("df_ev: {} ".format(df_ev))
   return render_template('predict.html',y = 'The resale value predicted is
{:.2f}$'.format(df_ev[0]))

if __name__=='__main__':
   app.run(debug= False)
```

Home.html:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
.container {
  position: relative;
  font-family: Arial;
}

.text-block {
  position: absolute;
  bottom: 20px;
  right: 20px;
  background-color: black;
  color: white;
  padding-left: 20px;
  padding-right: 20px;
}
```

```
.container .btn {
  position: absolute;
  top: 50%;
  left: 12.5%;
  display:flex;
  transform: translate(-50%, -50%);
  -ms-transform: translate(-50%, -50%);
  background-color: #f1f1f1;
  color: black;
  font-size: 16px;
  padding: 16px 30px;
  border: none;
  cursor: pointer;
  border-radius: 5px;
  text-align: center;
  text-decoration: none;
}
.container .btn:hover {
  background-color: black;
  color: white;
}
.myflex{
  display: flex;
  align-items: center;
  background-color: #21222A;
  color: white;
  justify-content: space-between;
  font-size: x-large;
}
.myflex > h3{
  margin-right: auto;
  font-weight: 700;
  font-size: 30px;
}
.myflex >h4{
  margin-right: 20px;
  font-weight: 600;
}
.logo{
  width: 60px;
  margin-right: 20px;
 height: 90px;

}
.para1{
 display: flex;
  align-items: center;
 justify-content: space-between;
margin:10px;
text-align: justify;
tab-size: 8;
}
.para2{
display: flex;
  align-items: center;
 justify-content: space-between;
margin:10px;
text-align: justify;
tab-size: 8;
}
.para1 >h3{

  font-family: "Times New Roman", Times, serif;
  font-weight: normal;

}
.para2>h3
{
 font-family: "Times New Roman", Times, serif;
  font-weight: normal;
}

</style>
</head>
<body>

<header>
            <nav class='myflex'>
             <img src="{{url_for('static', filename='images/i1.png')}}" class="logo"
alt="logo"><br>
```

```
            <h3>Car Price Predictor!</h3>
            <h4>Home</h4>
           </nav>
  </header>
<div class="container">
  <img src="{{url_for('static', filename='images/unnamed.jpg')}}" alt="Nature"
style="width:100%;">
  <a href="predict.html" class="btn">Want to know the resale value of your
car?</a>
  <div class="text-block">
    <h4>Hurray</h4>
    <p>A Car For All Budgets!!!</p>
  </div>
</div>
<div class="para">

<div class="para1">



       <h3 ><i>


The growing world of e-commerce is not just restricted to buying electronics and
clothings but everything that you expect in a general store.
 Keeping the general store perspective aside and looking at the bigger picture,
every day there are thousands or perhaps millions of deals happening in the digital
marketplace.
One of the most booming markets in the digital space is that of the automobile
industry wherein the buying and selling of used cars take place.
The rise of e-commerce facilities and the practical aspect of unaffordability due to
inflation have created a niche market for used vehicles.
The only difference here is that you do not have to walk up to the dealer or
individual sellers to get a used car price quote, instead you get the used car
valuation at the comfort of your home within 10 seconds.
However, buyers and sellers face a major stumbling block when it comes to their
used car valuation or say their second-hand car price.</i></h3>


</div>
```

```
<div class="para2">


<h3><i>


Traditionally, you would go to a showroom and get your vehicle inspected before learning about the price,
but now you do not need to do that anymore. With technologically advanced websites which is a well-known used car valuation tool,
you can simply check your pre-owned car price online in a hassle-free manner. You can check used Audi car price, used Hyundai car price,
second-hand Honda car price, and so on. As a seller, you will always look to make the most out of the deal, and as a buyer,
you are not willing to spend an extra penny on the deal for Used car price. The difference in thoughts and expectations often keeps buyers
from buying and sellers from selling the product. However, you need not hear from anyone else. Simply visit the website and with the help of a used car pricing
 calculator, you can get the right amount range for your used car.</i></h3>
</div>



</div>
</body>
</html>
```

predict.html :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Prediction Form</title>
    <link href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700" rel="stylesheet">
    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.5.0/css/all.css" integrity="sha384-B4dIYHKNBt8Bc12p+WXckhzcICo0wtJAoU8YZTY5qE0Id1GSseTk6S+L3BlXeVIU" crossorigin="anonymous">
    <style>
```

```
html, body {
min-height: 100%;
}
body, div, form, input, select, textarea, label {
padding: 0;
margin: 0;
outline: none;
font-family: Roboto, Arial, sans-serif;
font-size: 14px;
color: #666;
line-height: 22px;
}
h1 {
position: absolute;
margin: 0;
font-size: 40px;
color: #fff;
z-index: 2;
line-height: 83px;
}
.testbox {
display: flex;
justify-content: center;
align-items: center;
height: inherit;
padding: 20px;
}
form {
width: 100%;
padding: 20px;
border-radius: 6px;
background: #fff;
box-shadow: 0 0 8px  #cc7a00;
}
.banner {
position: relative;
height: 300px;
background-image: url({{ url_for('static', filename='images/i1.png') }})
background-size: cover;
display: flex;
justify-content: center;
align-items: center;
text-align: center;
}
.banner::after {
content: "";
background-color: rgba(0, 0, 0, 0.2);
position: absolute;
width: 100%;
height: 100%;
}
input, select, textarea {
margin-bottom: 10px;
border: 1px solid #ccc;
border-radius: 3px;
}
input {
width: calc(100% - 10px);
padding: 5px;
}
input[type="date"] {
padding: 4px 5px;
}
textarea {
width: calc(100% - 12px);
padding: 5px;
}
.item:hover p, .item:hover i, .question:hover p, .question label:hover, input:hover::placeholder {
color: #cc7a00;
}
.item input:hover, .item select:hover, .item textarea:hover {
border: 1px solid transparent;
box-shadow: 0 0 3px 0 #cc7a00;
color: #cc7a00;
}
.item {
position: relative;
margin: 10px 0;
}
```

```
.item span {
color: red;
}
input[type="date"]::-webkit-inner-spin-button {
display: none;
}
.item i, input[type="date"]::-webkit-calendar-picker-indicator {
position: absolute;
font-size: 20px;
color: #cc7a00;
}
.item i {
right: 1%;
top: 30px;
z-index: 1;
}
input[type=radio], input[type=checkbox]  {
display: none;
}
label.radio {
position: relative;
display: inline-block;
margin: 5px 20px 15px 0;
cursor: pointer;
}
.question span {
margin-left: 30px;
}
.question-answer label {
display: block;
}
label.radio:before {
content: "";
position: absolute;
left: 0;
width: 17px;
height: 17px;
border-radius: 50%;
border: 2px solid #ccc;
}
```

```
input[type=radio]:checked + label:before, label.radio:hover:before {
border: 2px solid #cc7a00;
}
label.radio:after {
content: "";
position: absolute;
top: 6px;
left: 5px;
width: 8px;
height: 4px;
border: 3px solid #cc7a00;
border-top: none;
border-right: none;
transform: rotate(-45deg);
opacity: 0;
}
input[type=radio]:checked + label:after {
opacity: 1;
}
.btn-block {
margin-top: 10px;
text-align: center;
}
button {
width: 150px;
padding: 10px;
border: none;
border-radius: 5px;
background: #cc7a00;
font-size: 16px;
color: #fff;
cursor: pointer;
}
button:hover {
background: #ff9800;
}
@media (min-width: 568px) {
.name-item, .city-item {
display: flex;
flex-wrap: wrap;
```

```
    justify-content: space-between;
    }
    .name-item input, .name-item div {
    width: calc(50% - 20px);
    }
    .name-item div input {
    width:97%;}
    .name-item div label {
    display:block;
    padding-bottom:5px;
    }
    }


footer {

text-align: center;

padding: 3px;

background-color: DarkSalmon;

color: white;

}
  </style>
 </head>
 <body>
  <div class="testbox">
   <form action="/y_predict" method="post">
    <div class="banner">
     <h1>Prediction Form</h1>
    </div>
    <p>Enter car details</p>
    <div class="item">
     <label for="Registration year">Registration year<span>*</span></label>
     <input id="Registration_year" type="text" name="Registrationyear"
required/>
    </div>
    <div class="item">
```

```
          <label for="Registration month">Registration
month<span>*</span></label>
          <input id="Registration_month" type="text" name="Registrationmonth"
required/>
        </div>
        <div class="item">
          <label for="Power of car in PS">Power of car in PS<span>*</span></label>
          <input id="Power_of_car_in_PS" type="text" name="PowerofcarinPS"
required/>
        </div>
        <div class="item">
          <label for="Kilometers Driven">Kilometers Driven<span>*</span></label>
          <input id="Kilometers_Driven" type="text" name="KilometersDriven"
required/>
        </div>

<div class="question">
          <label>Gear Box Type</label>
          <div class="question-answer">
            <div>
              <input type="radio" value="manual" id="radio_1" name="Geartype"/>
              <label for="radio_1" class="radio"><span>Manual</span></label>
            </div>
            <div>
              <input  type="radio" value="automatic" id="radio_2" name="Geartype"/>
              <label for="radio_2" class="radio"><span>Automatic</span></label>
            </div>



<div>
              <input  type="radio" value="not-declared" id="radio_3"
name="Geartype"/>
              <label for="radio_3" class="radio"><span>Not-declared</span></label>
            </div>
          </div>
        </div>
       <div class="question">
          <label>Car damaged/reparied</label>
          <div class="question-answer">
            <div>
```

```
              <input type="radio" value="Yes" id="d_radio_1" name="cd"/>
              <label for="d_radio_1" class="radio"><span>Yes</span></label>
            </div>
            <div>
              <input  type="radio" value="No" id="d_radio_2" name="cd"/>
              <label for="d_radio_2" class="radio"><span>No</span></label>
            </div>


<div>
              <input  type="radio" value="not-declared" id="d_radio_3" name="cd"/>
              <label for="d_radio_3" class="radio"><span>Not-declared</span></label>
            </div>
          </div>
        </div>
        <div class="item">
          <p>Model Type</p>
          <select id="model" name="model">
            <option selected value="" disabled selected></option>
            <option value="grand" >grand</option>
            <option value="golf">golf</option>
            <option value="fabia">fabia</option>
            <option value="3er">3er</option>
            <option value="2_reihe">2_reihe</option>
            <option value="c_max">c_max</option>
            <option value="3_reihe">3_reihe</option>
            <option value="passat">passat</option>
            <option value="navara">navara</option>
            <option value="polo">polo</option>
            <option value="twingo">twingo</option>
            <option value="a_klasse">a_klasse</option>
            <option value="scirocco">scirocco</option>
            <option value="5er">5er</option>
            <option value="andere">andere</option>
            <option value="civic">civic</option>
            <option value="punto">punto</option>
            <option value="e_klasse">e_klasse</option>
            <option value="clio">clio</option>
            <option value="kadett">kadett</option>
```

```
<option value="one">one</option>
<option value="fortwo">fortwo</option>
<option value="1er">1er</option>
<option value="b_klasse">b_klasse</option>
<option value="a8">a8</option>
<option value="jetta">jetta</option>
<option value="c_klasse">c_klasse</option>
<option value="micra">micra</option>
<option value="vito">vito</option>
<option value="sprinter">sprinter</option>
<option value="astra">astra</option>
<option value="156">156</option>
<option value="escort">escort</option>
<option value="forester">forester</option>
<option value="xc_reihe">xc_reihe</option>
<option value="fiesta">fiesta</option>
<option value="scenic">scenic</option>
<option value="ka">ka</option>
<option value="a1">a1</option>
<option value="transporter">transporter</option>
<option value="focus">focus</option>
<option value="a4">a4</option>
<option value="tt">tt</option>
<option value="a6">a6</option>
<option value="jazz">jazz</option>
<option value="omega">omega</option>
<option value="slk">slk</option>
<option value="7er">7er</option>
<option value="combo">combo</option>
<option value="corsa">corsa</option>
<option value="80">80</option>
<option value="147">147</option>
<option value="glk">glk</option>
<option value="z_reihe">z_reihe</option>
<option value="sorento">sorento</option>
<option value="ibiza">ibiza</option>
<option value="mustang">mustang</option>
<option value="eos">eos</option>
<option value="touran">touran</option>
<option value="getz">getz</option>
```

```
<option value="insignia">insignia</option>
<option value="almera">almera</option>
<option value="megane">megane</option>
<option value="a3">a3</option>
<option value="r19">r19</option>
<option value="caddy">caddy</option>
<option value="mondeo">mondeo</option>
<option value="cordoba">cordoba</option>
<option value="colt">colt</option>
<option value="impreza">impreza</option>
<option value="vectra">vectra</option>
<option value="lupo">lupo</option>
<option value="berlingo">berlingo</option>
<option value="m_klasse">m_klasse</option>
<option value="tiguan">tiguan</option>
<option value="6_reihe">6_reihe</option>
<option value="c4">c4</option>
<option value="panda">panda</option>
<option value="up">up</option>
<option value="i_reihe">i_reihe</option>
<option value="ceed">ceed</option>
<option value="kangoo">kangoo</option>
<option value="5_reihe">5_reihe</option>
<option value="yeti">yeti</option>
<option value="octavia">octavia</option>
<option value="zafira">zafira</option>
<option value="mii">mii</option>
<option value="rx_reihe">rx_reihe</option>
<option value="6er">6er</option>
<option value="modus">modus</option>
<option value="fox">fox</option>
<option value="matiz">matiz</option>
<option value="beetle">beetle</option>
<option value="rio">rio</option>
<option value="touareg">touareg</option>
<option value="logan">logan</option>
<option value="spider">spider</option>
<option value="cuore">cuore</option>
<option value="s_max">s_max</option>
<option value="a2">a2</option>
```

```
<option value="x_reihe">x_reihe</option>
<option value="a5">a5</option>
<option value="galaxy">galaxy</option>
<option value="c3">c3</option>
<option value="viano">viano</option>
<option value="s_klasse">s_klasse</option>
<option value="1_reihe">1_reihe</option>
<option value="sharan">sharan</option>
<option value="avensis">avensis</option>
<option value="sl">sl</option>
<option value="roomster">roomster</option>
<option value="q5">q5</option>
<option value="santa">santa</option>
<option value="leon">leon</option>
<option value="cooper">cooper</option>
<option value="4_reihe">4_reihe</option>
<option value="sportage">sportage</option>
<option value="laguna">laguna</option>
<option value="ptcruiser">ptcruiser</option>
<option value="clk">clk</option>
<option value="primera">primera</option>
<option value="espace">espace</option>
<option value="exeo">exeo</option>
<option value="159">159</option>
<option value="transit">transit</option>
<option value="juke">juke</option>
<option value="v40">v40</option>
<option value="carisma">carisma</option>
<option value="accord">accord</option>
<option value="corolla">corolla</option>
<option value="lanos">lanos</option>
<option value="phaeton">phaeton</option>
<option value="boxster">boxster</option>
<option value="verso">verso</option>
<option value="rav">rav</option>
<option value="kuga" >kuga</option>
<option value="qashqai">qashqai</option>
<option value="swift">swift</option>
<option value="picanto">picanto</option>
<option value="superb" >superb</option>
```

```
<option value="stilo">stilo</option>
<option value="911">911</option>
<option value="m_reihe">m_reihe</option>
<option value="roadster">roadster</option>
<option value="ypsilon" >ypsilon</option>
<option value="galant">galant</option>
<option value="justy">justy</option>
<option value="90">90</option>
<option value="sirion" >sirion</option>
<option value="signum">signum</option>
<option value="crossfire">crossfire</option>
<option value="agila">agila</option>
<option value="duster">duster</option>
<option value="v50" >v50</option>
<option value="mx_reihe">mx_reihe</option>
<option value="meriva">meriva</option>
<option value="discovery">discovery</option>
<option value="c_reihe" >c_reihe</option>
<option value="v_klasse">v_klasse</option>
<option value="yaris">yaris</option>
<option value="c5">c5</option>
<option value="aygo">aygo</option>
<option value="seicento">seicento</option>
<option value="cc">cc</option>
<option value="carnival">carnival</option>
<option value="fusion">fusion</option>
<option value="bora" >bora</option>
<option value="cl">cl</option>
<option value="tigra">tigra</option>
<option value="300c">300c</option>
<option value="500">500</option>
<option value="100">100</option>
<option value="q3">q3</option>
<option value="cr_reihe">cr_reihe</option>
<option value="spark">spark</option>
<option value="x_type">x_type</option>
<option value="ducato">ducato</option>
<option value="s_type">s_type</option>
<option value="x_trail">x_trail</option>
<option value="toledo">toledo</option>
```

```
<option value="altea">altea</option>
<option value="voyager">voyager</option>
<option value="calibra">calibra</option>
<option value="v70">v70</option>
<option value="bravo">bravo</option>
<option value="range_rover">range_rover</option>
<option value="forfour">forfour</option>
<option value="tucson">tucson</option>
<option value="q7">q7</option>
<option value="c1">c1</option>
<option value="citigo">citigo</option>
<option value="jimny">jimny</option>
<option value="cx_reihe">cx_reihe</option>
<option value="cayenne">cayenne</option>
<option value="wrangler">wrangler</option>
<option value="lybra">lybra</option>
<option value="range_rover_sport">range_rover_sport</option>
<option value="lancer">lancer</option>
<option value="freelander">freelander</option>
<option value="captiva">captiva</option>
<option value="range_rove_evoque">range_rover_evoque</option>
<option value="sandero">sandero</option>
<option value="note">note</option>
<option value="antara">antara</option>
<option value="900">900</option>
<option value="defender">defender</option>
<option value="cherokee">cherokee</option>
<option value="clubman">clubman</option>
<option value="arosa">arosa</option>
<option value="legacy">legacy</option>
<option value="pajero">pajero</option>
<option value="auris">auris</option>
<option value="c2">c2</option>
<option value="niva">niva</option>
<option value="s60">s60</option>
<option value="nubira">nubira</option>
<option value="vivaro">vivaro</option>
<option value="g_klasse">g_klasse</option>
<option value="lodgy">lodgy</option>
<option value="850">850</option>
```

```html
            <option value="serie_2">serie_2</option>
            <option value="charade">charade</option>
            <option value="croma">croma</option>
            <option value="outlander">outlander</option>
            <option value="g1">g1</option>
             <option value="kaefer">kaefer</option>
            <option value="doblo">doblo</option>
             <option value="musa">musa</option>
            <option value="amarok">amarok</option>
            <option value="9000">9000</option>
            <option value="kalos">kalos</option>
            <option value="v60">v60</option>
            <option value="200">200</option>
            <option value="145">145</option>
            <option value="b_max">b_max</option>
            <option value="delta">delta</option>
            <option value="aveo">aveo</option>
            <option value="rangerover">rangerover</option>
            <option value="move">move</option>
            <option value="materia">materia</option>
            <option value="terios">terios</option>
            <option value="kalina">kalina</option>
            <option value="elefantino">elefantino</option>
            <option value="i3">i3</option>
            <option value="samara">samara</option>
            <option value="kappa">kappa</option>
            <option value="serie_3">serie_3</option>
            <option value="discovery_sport">discovery_sport</option>
            <option value="not-declared">not-declared</option>
          </select>
        </div>

    <div class="item">
          <p>Brand of the car</p>
          <select id="brand" name="brand">
            <option selected value="" disabled selected></option>
            <option value="seat" >seat</option>
            <option value="lancia">lancia</option>
            <option value="porsche">porsche</option>
            <option value="citroen">citroen</option>
            <option value="toyota" >toyota</option>
            <option value="chevrolet">chevrolet</option>
            <option value="dacia">dacia</option>
            <option value="suzuki">suzuki</option>
            <option value="chrysler">chrysler</option>
            <option value="daihatsu" >daihatsu</option>
            <option value="jaguar">jaguar</option>
            <option value="daewoo">daewoo</option>
            <option value="rover">rover</option>
            <option value="sonstige_autos" >sonstige_autos</option>
            <option value="saab">saab</option>
            <option value="land_rover">land_rover</option>
            <option value="lada">lada</option>
            <option value="trabant">trabant</option>
            <option value="audi" >audi</option>
            <option value="jeep">jeep</option>
            <option value="volkswagen">volkswagen</option>
            <option value="skoda">skoda</option>
            <option value="bmw" >bmw</option>
            <option value="peugeot">peugeot</option>
            <option value="ford">ford</option>
            <option value="mazda">mazda</option>
            <option value="nissan" >nissan</option>
            <option value="renault">renault</option>
            <option value="mercedes_benz">mercedes_benz</option>
            <option value="honda">honda</option>
            <option value="fiat">fiat</option>
            <option value="opel" >opel</option>
            <option value="mini">mini</option>
            <option value="smart">smart</option>
            <option value="hyundai">hyundai</option>
            <option value="alfa_romeo" >alfa_romeo</option>
            <option value="subaru">subaru</option>
            <option value="volvo">volvo</option>
            <option value="mitsubishi">mitsubishi</option>
            <option value="kia">kia</option>
          </select>
        </div>

<div class="item">
```

```
          <p>Fuel type of the car</p>
          <select id="fuel_type" name="fueltype">
            <option selected value="" disabled selected></option>
            <option value="diesel" >diesel</option>
            <option value="petrol">petrol</option>
            <option value="lpg">lpg</option>
            <option value="hybrid">hybrid</option>
            <option value="cng">cng</option>
            <option value="electric">electric</option>
            <option value="others">others</option>
            <option value="not-declared">not-declared</option>
          </select>
        </div>

<div class="item">
          <p>Vechicle Type</p>
          <select id="vechicle_type" name="vechicletype">
            <option selected value="" disabled selected></option>
            <option value="coupe" >coupe</option>
            <option value="suv">suv</option>
            <option value="small car">small car</option>
            <option value="limousine">limousine</option>
            <option value="convertible" >convertible</option>
            <option value="bus">bus</option>
            <option value="combination">combination</option>
            <option value="others">others</option>
            <option value="not-declared">not-declared</option>
          </select>
        </div>
        <div class="btn-block">
          <button type="submit" >PREDICT</button>
        </div>
      </form>

    </div>
<footer>
              <h4> <b>{{y}}</b></h4>
            </footer>
  </body>
</html>
```

GITHUB LINK:

https://github.com/IB0M-EPBL/IBM-Project-45640-1660731393

PROJRCT DEMO LINK:

https://drive.google.com/file/d/16078mlsdwRrWMjA9ySkjBaM_zLoGmOTp/view?usp=drivesdk