PANIMALAR INSTITUTE OF TECHNOLOGY

391, Bangalore Trunk Road, Varadharajapuram,

Poonamallee, Chennai-600 123.

PROJECT REPORT

Project Name: Car Resale Value Prediction

TEAM ID : PNT2022TMID25799

TEAM MEMBERS:

- Emmanuel Stanes (Team Leader)

- Kirubakaran

- Jebason Chellakumar

- Mohammed Ameen

1. INTRODUCTION

1.1 Project Overview

- To understand the problem to classify if it is a regression or a classification kind of problem.
- To pre-process/clean the data using different data pre-processing techniques.

- Applying different algorithms according to the dataset

1.2 Purpose

To build a working web application using the Python Flask Framework and deploy our built model on it to meet user satisfaction.

2. LITERATURE SURVEY

2.1 Existing problem

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.
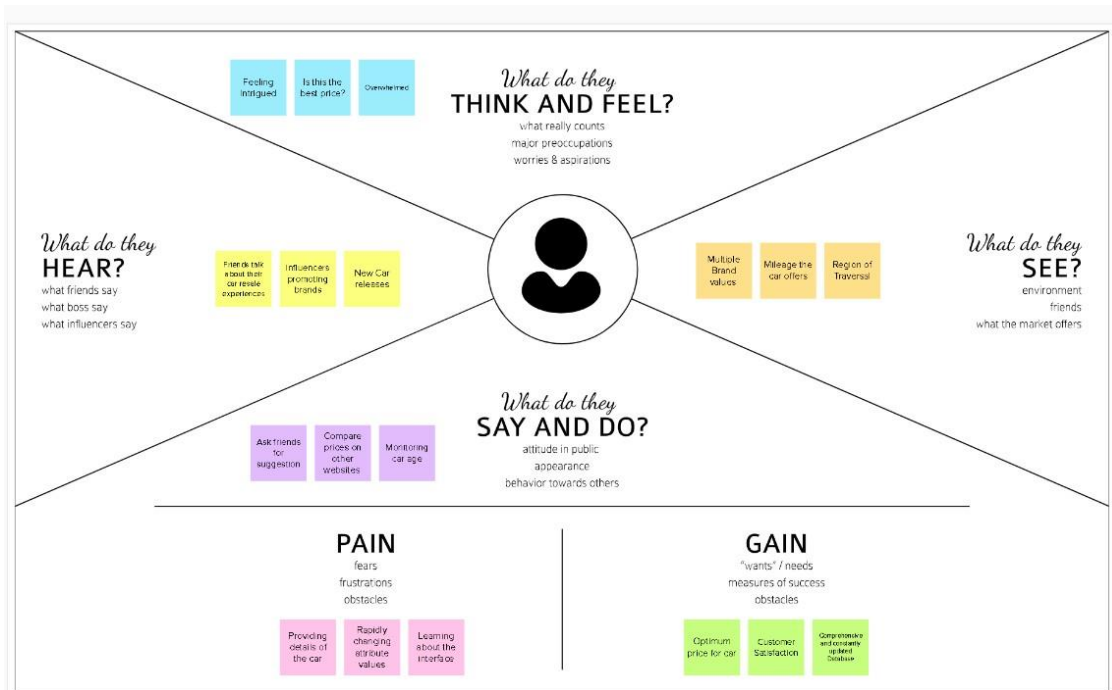
2.2 References

- Voß, S. (2013). Resale Price Prediction in the Used Car Market.
- Kiran, S., 2020. Prediction of resale value of the car using linear regression algorithm. *Int. J. Innov. Sci. Res. Technol*, *6*(7), pp.382-386.\
- Gegic, E., Isakovic, B., Keco, D., Masetic, Z. and Kevric, J., 2019. Car price prediction using machine learning techniques. *TEM Journal*, *8*(1), p.113.
- Ganesh, Mukkesh & Venkatasubbu, Pattabiraman. (2019). Used Cars Price Prediction using Supervised Learning Techniques. International Journal of Engineering and Advanced Technology. 9. 216-223. 10.35940/ijeat.A1042.1291S319.
- Pudaruth, S., 2014. Predicting the price of used cars using machine learning techniques. *Int. J. Inf. Comput. Technol*, *4*(7), pp.753-764.

2.3 Problem Statement Definition

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.
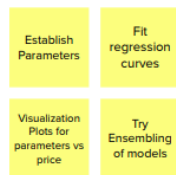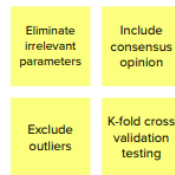
3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming

Scikit Learn package

Establish Parameters

Visualization Plots for parameters vs price

Eliminate irrelevant parameters

Fit regression curves

Exclude outliers

Backend in NoSQL(MongoDB)

Try Ensembling of models

Prerprocess data

Selenium python package for automation

K-fold cross validation testing

Use web-scraping to extract datasets

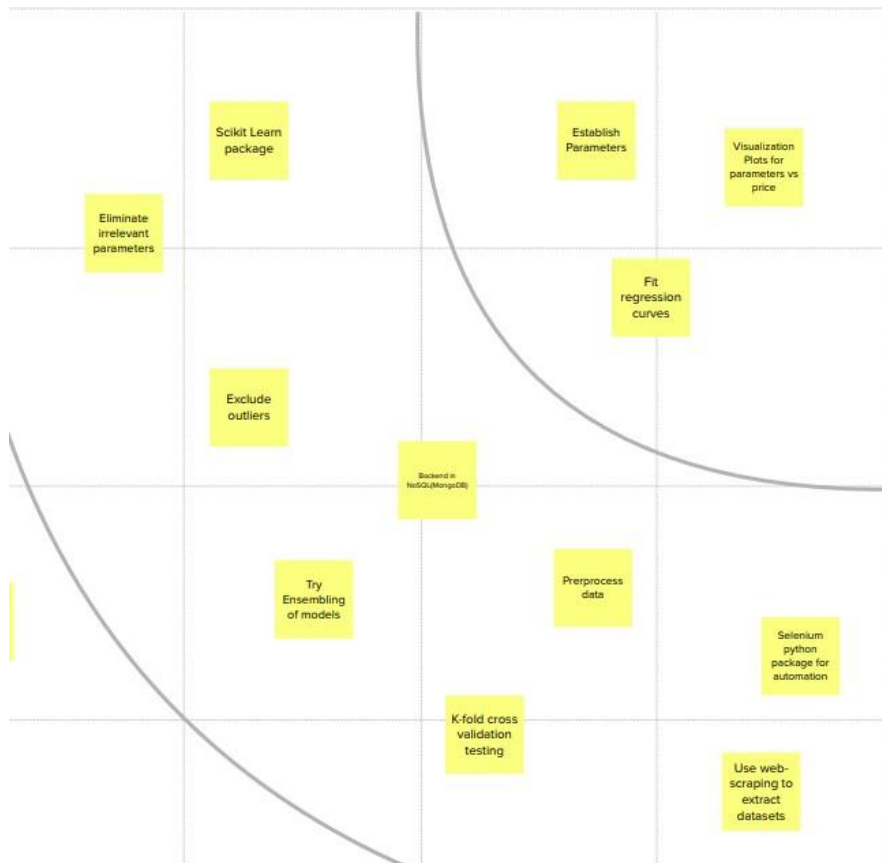## 3.3 Proposed Solution

The dataset has to be pre-processed in an appropriate way prior to Machine Learning model implementation. The stacking algorithm is employed to create an ensemble of strong learners. These learning algorithms can be finalized through experimentation. The model deployment is executed via the Flask platform and the model will be deployed via "pikl" files and on the IBM Cloud platform.

3.4 Problem Solution fit
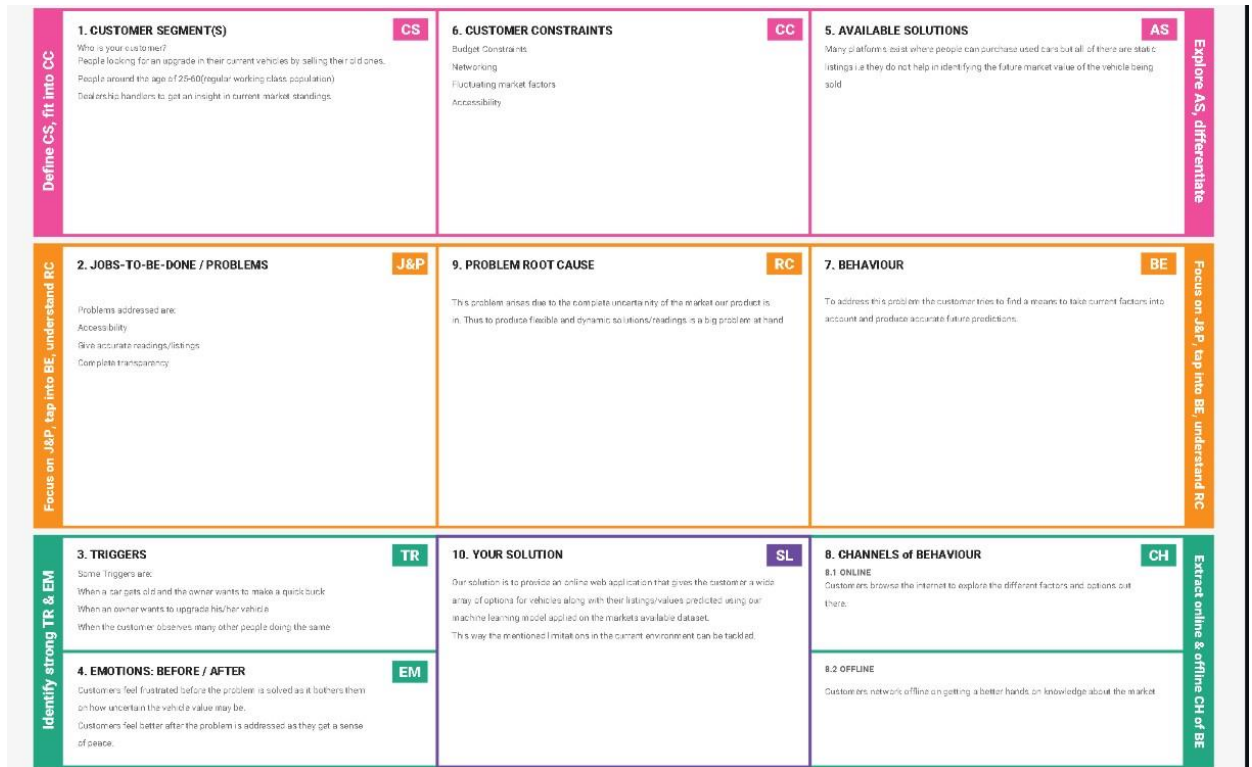
4. REQUIREMENT ANALYSIS

4.1 Functional requirement

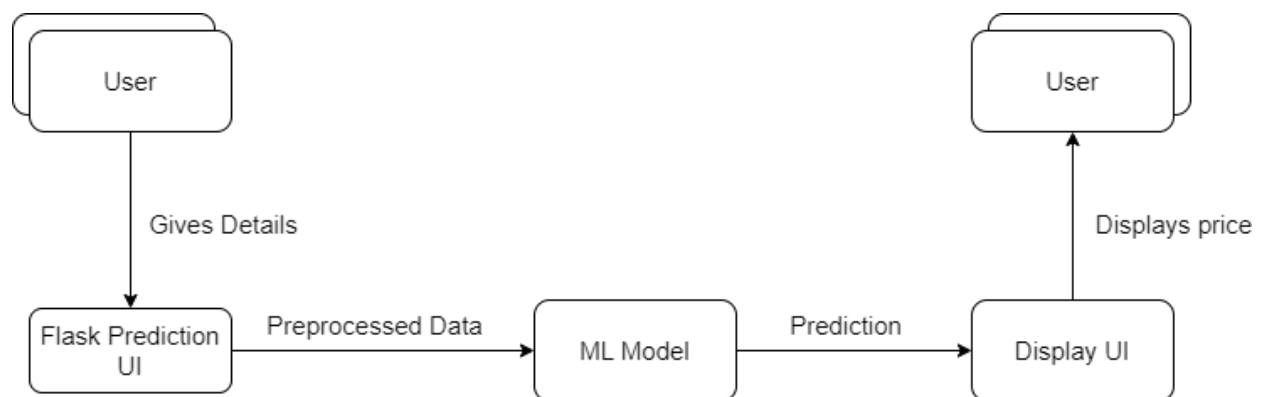| FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration and Login	Registration through Form Registration through Email Login through Email
FR-2	User should be able to input car details	Car information like date of purchase, price, damages incurred, etc are entered by the user
FR-3	User should be able to view past predictions	User can view the previous predictions the model has made on different cars and categorise according to the brand, type of car, date of purchase, etc.

| FR-4	System predicts car resale vale	Taking input features given by user, system should be able to predict car price by forwarding the prediction request to the ML model.
FR-5	Admin should be notified of any errors in the system	Any error that occurs like the model taking a long time to evaluate resale price should be notified to the admin so that the problem might be fixed.

4.2 Non-Functional requirements

NFR-1	**Usability**	Effective User Interface with descriptions for each feature and proper layout that ensures each user finds it easy to access and interact with the system.
NFR-2	**Security**	Account creation for each user with a mandatory password strength check while creating the account.
NFR-3	**Reliability**	Chance of critical failure should be less than or equal to 2%.
NFR-4	**Performance**	The system must provide a webpage rendering images and texts upon receiving a request within a time of 8 seconds over a standard internet connection.
NFR-5	**Availability**	The website should be available to users 24x7. Any issues or errors will be addressed within the next 24 hours.
NFR-6	**Scalability**	The system must be scalable enough to support 1,00,000 requests at the same time without crashing.

5. PROJECT DESIGN

5.1 Data Flow Diagrams

5.2 Solution Technical Architecture
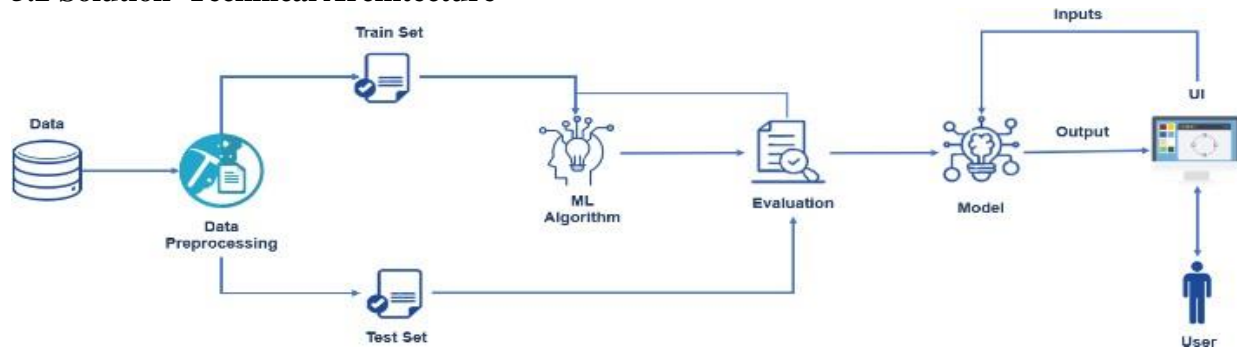
5.3 User Stories

User Type	Functional Requirements	User Story Number	User Story	Acceptance Criteria	Priority	Release
Customer	Entering Car Details	USN-1	The user enters the necessary car details	All the mandatory fields are filled	High	Sprint 1
Customer	Viewing valuation	USN-2	The user's car's value is predicted by model	The value is displayed	High	Sprint 1
Admin	Updating model	USN-3	The admin can update the ML model after modifying it.	The ML model is properly loaded into the app.	High	Sprint 2

6 PROJECT PLANNING & SCHEDULING

6.2 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.
Sprint-1	Login	USN-2	As a user, I can log into the application by entering email & password
Sprint-2	Model Development		Gather dataset and perform pre-processing and cleaning
Sprint-2			Training and testing of model using data at hand

Sprint-2			Integrating model in application
Sprint-3	Input for prediction	USN-3	As a user, I should be able to give inputs to the application that will be sent to the model for prediction
Sprint-3	Prediction	USN-4	As a user, I will be able to view the predicted valuation for given vehicle
Sprint-4	Previous predictions	USN-5	As a user, I will be able to view previous predictions made by the model
Sprint-4	Update model	USN-6	As an admin, I will be able to update the model according to the fluctuating environment

7. ADVANTAGES & DISADVANTAGES

Advantages:

• The model is fairly accurate and is able to give a good prediction of what the actual resale value might be.

• The model is very quick in calculating the predictions.

• Errors of one model will be reduced by the ensembling with other models.

• It is easier for us to upload a better trained version of the model onto the cloud.
Disadvantages:

• The datasets available and the dataset the model has trained on do not give sufficient information to the model as it does not have very useful information.

• Attributes such as Fuel Mileage and Popularity of model has not been taken into account which can give a better idea about the resale value

8. CONCLUSION

Thus with our project we can help both working resale dealers in the market and general users who want to sell their used cars by providing them with accurate resale value prediction.

9. FUTURE SCOPE

Can help with shaping the future of car resale market as our project opens up an insight into how the factors can be taken into for predicting the value of used cars.

10. Appendix

HTML Code:

```html
<!DOCTYPE html>
<html lang="en">

<head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- Bootstrap CSS -->
<link            rel="stylesheet"            href="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/css/bootstrap.min.css"
integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
crossorigin="anonymous">
<link rel="image" href="car.jpeg">
<title>Car Resale Value</title>
<style>
body {
background-image: url('car.jpeg');
background-repeat: no-repeat;
background-attachment: fixed;
background-size: cover;
}
#colour{
color:black;
}
#space{

justified-content:left;
}
.box{
position:relative;
width:400px;
height:100px;
background:white;
border:1px solid black;
border-top-left-radius:20px;
border-bottom-right-radius:20px;
}
.box1{
position:relative;
width:500px;
height:800px;
background:white;
border:1px solid black;
border-top-right-radius:20px;
```

```
border-bottom-left-radius:20px;
}
.box2{
position:relative;
width:300px;
height:300px;
background:white;
border:1px solid black;
border-top-right-radius:20px;
border-bottom-left-radius:20px;

}


</style>


</head>

<body class="bg-warning" >
<form action="#" method="post">

<div id=colour>

<center>
<div class=box>
<h1 style="color:black">CAR RESALE VALUE PREDICTION</h1><br>
</div></center><br>

<div class=box1>
<div id=space>
<br>

<h2>Car company :</h2>
<select        class="selectpicker        form-control"        id="company"        name="company"        required="1"
onchange="load_car_models(this.id, 'car_model')">
{% for company in companies %}
<option value="{{ company}}">{{ company }}</option>
{% endfor %}
</select>
<br><br>

<h2>Car Model : </h2>
<select class="selectpicker form-control" id="car_model" name="car_model" required="1">
<option value=""></option>
</select>
<br><br>
```

```html
<h2>Year of Purchase: </h2>
<select class="selectpicker form-control" id="year" name="year" required="1">
{% for year in years %}
<option value="{{ year }}">{{ year }}</option>
{% endfor %}
</select>

<br><br>

<h2>Distance Driven (Kms):</h2> <input class="form-control" id="kio_driven" name="kilo_driven" type="text"
placeholder="Enter no. of Kilometers driven">

<br><br>

<h2> Fuel type: </h2>
<select class="selectpicker form-control" id="fuel_type" name="fuel_type" required="1">
{% for fuel_type in fuel_types %}
<option value="{{ fuel_type }}">{{ fuel_type }}</option>
{% endfor %}
</select></h2>
<br><br>


<button class="btn btn-primary form-control" onclick="send_data()" >Calculate</button>



</div></div></div>
</form>
<br>
<div class=box2 style = "position:relative; left:900px; top:-335px; ">

<div class="col-12">
<h3><span id="prediction" ></span></h3>
</div></div>
<script>

function load_car_models(company_id, car_model_id)
{
var company=document.getElementById(company_id);
var car_model = document.getElementById(car_model_id);

car_model.value="";
car_model.innerHTML="";
{% for company in companies %}

if(company.value == "{{company}}")
{
```

```
{% for model in car_models %}
{% if company in model %}

var newOption = document.createElement("option");
newOption.value="{{ model}}";
newOption.innerHTML="{{ model }}";
car_model.options.add(newOption);

{% endif %}
{% endfor %}
}

{% endfor %}
}
function form_handler(event)
{
event.preventDefault();
}
function send_data()
{
document.querySelector('form').addEventListener('submit', form_handler);

var fd= new FormData(document.querySelector('form'));
var xhr= new XMLHttpRequest();
xhr.open('post','/predict',true);
document.getElementById("prediction").innerHTML="Please Wait..!";

xhr.onreadystatechange=function()
{
if(xhr.readyState == XMLHttpRequest.DONE)
{
document.getElementById("prediction").innerHTML="Your car can be sold For RS."+xhr.responseText;
}
}
xhr.onload=function(){};
xhr.send(fd);
}
</script>
<script                 src="https://code.jquery.com/jquery-3.3.1.slim.min.js"                 integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
<script         src="https://cdn.jsdelivr.net/npm/popper.js@1.14.3/dist/umd/popper.min.js"         integrity="sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIPm49"
crossorigin="anonymous"></script>
<script         src="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/js/bootstrap.min.js"         integrity="sha384-
ChfqqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
crossorigin="anonymous"></script>
```

```
</body>
```

Backend.py

```python
from flask import Flask,render_template, request
import pandas as pd
import pickle
import numpy as np


app=Flask(__name__,template_folder='template')


model=pickle.load(open("test.pkl",'rb'))
car=pd.read_csv("Cleaned car.csv")
@app.route("/")
def index():
    companies = sorted(car['company'].unique())
    car_models = sorted(car['name'].unique())
    year = sorted(car['year'].unique(), reverse=True)
    fuel_type=car['fuel_type'].unique()
    companies.insert(0,"Select Company")
    return render_template('index.html', companies=companies, car_models=car_models, years=year, fuel_types=fuel_type)

@app.route('/predict',methods=['POST'])
def predict():
    company= request.form.get('company')
    car_model=request.form.get('car_model')
    year=int(request.form.get('year'))
    fuel_type=request.form.get('fuel_type')
    kms_driven=int(request.form.get('kilo_driven'))

    prediction    =    model.predict(pd.DataFrame([[car_model,    company,    year,    kms_driven,    fuel_type]],
columns=['name','company', 'year','kms_driven','fuel_type']))

    return str(np.round(prediction[0],2))

if __name__=="__main__":
    app.run(debug=True)
```