PROJECT REPORT

CAR RESALE VALUE PREDICTION

submitted by

PNT2022TMID37578

Ravindhran M - 312019205023

Shareeth Kumar G M - 312019205031

Siva Shunmugan R - 312019205034

Krishnakanth A- 312019205010

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 PROJECT OVERVIEW

A car price prediction has been a high-interest research area, as it requires noticeable effort and knowledge of the field expert. A considerable number of distinct attributes are examined for reliable and accurate prediction. To build a model for predicting the price of used cars, the applied three machine learning techniques are random forest, KN-N, and linear regression algorithm. Respective performances of different algorithms were then compared to find one that best suits the available data set. This ability to capture data, analyse it and use it to personalize a shopping experience or implement is the future of retail.

1.2 PURPOSE

Car resale value prediction helps the user to predict the resale value of the car depending upon various features like kilo-meters driven, fuel type, etc. This resale value prediction system is made for general purposes to just predict the amount that can be roughly acquired by the user. The most essential elements for the forecast are brand and model, period use of vehicle, mileage of the vehicle, gear type, and fuel type utilized in the vehicle just as fuel utilization per mile profoundly influence the cost of a vehicle because of continuous changes in the cost of fuel. In view of the different highlights and factors, and furthermore with the assistance of master information the vehicle value forecast has been done precisely.

CHAPTER 2
LITERATURE SURVEY

2.1 EXISTING PROBLEM

The prices of new cars in the industry is fixed by the manufacturer with some additional costs incurred by the Government in the form of taxes. So, customers buying a new car can be assured of the money they invest to be worthy. But due to the increased price of new cars and the incapability of customers to buy new cars due to the lack of funds, used cars sales are on a global increase. There is a need for a used car price prediction system to effectively determine the worthiness of the car using a variety of features. Even though there are websites that offers this service, their prediction method may not be the best. Besides, different models and systems may contribute on predicting power for a used car's actual market value. It is important to know their actual market value while both buying and selling.

2.2 REFERENCES

S. no	Author, Title, Publication	Tech Stack	Advantage	Disadvantage
01	Praful Rane, Deep Pandya, Dhawal Kotak, 'USED CAR PRICE PREDICTION', International Research Journal of Engineering and Technology (IRJET) Published on April 4, 2021	Random Forest Regression, Hyperparameters, Random forest, Extra-tree Regressor	It is estimated that a car's value depreciated to almost 50%. The calculation for the first year is about 20%. That means if you buy a used car launched a year ago, you can save up to 20% on its original cost.	Even though we curated a list of available cars under 50,000 but that wasn't for you to buy and use for your daily commute. Usually, these cars are lemons. They look fine on the outside but can land you huge repair costs while you use them.

02	Aditya Nikhade, Rohan Borde, Car Price Prediction using Machine Learning, International Advanced Research Journal in Science, Engineering and Technology Published on April 4, 2022	Linear Regression, Ridge Regression, and Lasso Regression	But the insurance premium for a new car is comparatively higher than an old car. If you search for a good vehicle in the used car market, the previous owner has already paid hefty premiums when the car was new.	However, banks and loan providers usually charge a higher rate of interest on a used car loan than they do on a new car loan. Their need to earn more anyhow so here's the catch.

03	Mukkesh Ganesh, Pattabirama Venkatasubbu, 'Used Cars Price Prediction using Supervised Learning Techniques', International Journal of Engineering and Advanced Technology Published in December 2019	ANOVA, Lasso Regression, Regression Tree, Tukey's Test	If you buy a used car from a brand authorized dealership, you get a warranty on the repair. Read this carefully. In my case, I bought a Maruti Suzuki car from a True Value dealership and the car was still under its original warranty from the factory.	The case with the used car market is a little different. There is no special month, they don't have to worry about the year-end too. Although the prices are obviously lower in the used car markets, you don't get that many "discounts".

04	Sameerchand Pudaruth 'Predicting the Price of Used Cars using Machine Learning Techniques' International Journal of Information & Computation Technology Published In 2014	Multiple linear regression analysis, k-nearest neighbors, naive Bayes and decision trees	This is the major difference between a new car and a used car. For the same cost as a new car, you can buy a used car from a couple of segments above. Imagine getting a D-segment car for the price of a top-end premium hatchback.	While you will be saving on lesser down payment and insurance costs, a used car comes with years of wear and tear in its parts and thousands of kilometers in its odo.

05	Abhishek Jha, Dr. Ramveer Singh Manish, Imran Saifi, Shipra Srivastava, 'Used car price prediction' International Journal of Advance Research, Ideas and Innovations in Technology(IJAR IIT) Published in July 19, 2021	Python, Flask, Pandas, Numpy, Seaborn, Sklearn, Matplotlib, Random Forest Regressor	While it's true that it's harder to get financing as easily as a new car, you get to pay a lot lesser in the case of used cars. The same goes for insurance too as you will be paying a lesser insurance bill.	When you buy a new car, it's made to order. You have the option of picking the color, the features, whether or not you want a sunroof, and more. When you buy a used car, you get what you pay for. If the car has a crappy radio, you'll have to deal with it or pay to have it replaced.

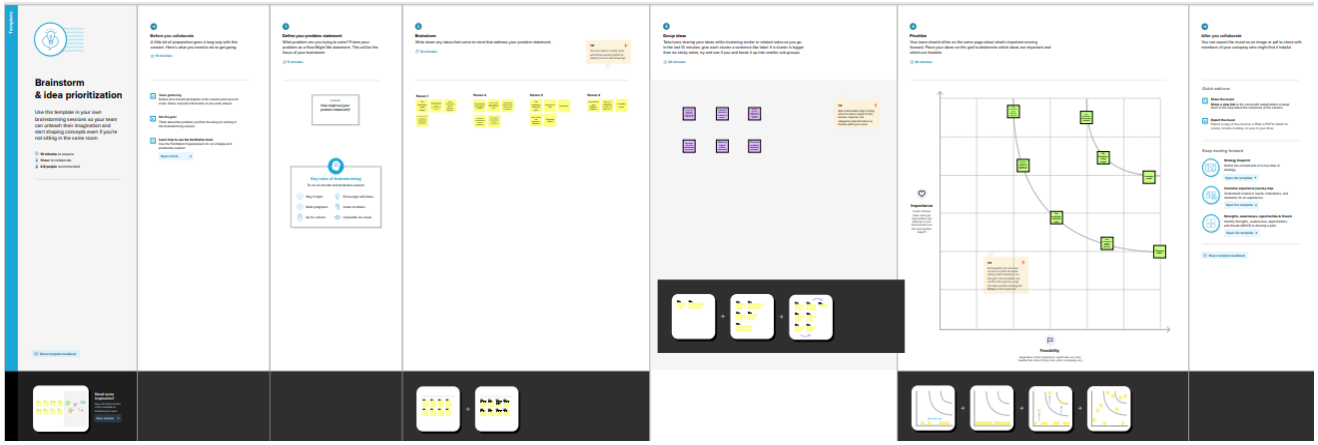
2.3 PROBLEM STATEMENT DEFINITION

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

CHAPTER 3
IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

3.2 IDEATION & BRAINSTORMING



3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To develop a webpage to predict the resale value of a car
2.	Idea / Solution description	To train the system with the dataset using a regression model
3.	Novelty / Uniqueness	By using the optimal regression model to predict the value in a less amount to time and predict its value
4.	Social Impact / Customer Satisfaction	The customer can get an idea about the resale value of their car .to have an idea whether to sell their vehicle or not based on their financial condition
5.	Business Model (Revenue Model)	The web based application has a friendly UI for the customer to enter their vehicles detail and the system predicts the value within few seconds
6.	Scalability of the Solution	The solution given by the trained system is efficient and is nearly accurate value of the vehicle.

3.4 PROBLEM SOLUTION FIT

Project Title: Car Resale value Prediction | **Project Design Phase-I - Solution Fit** | **Team ID:** PNT2022TMID52696

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

- Dealers
- Avid Buyers over the age of 18

6. CUSTOMER CONSTRAINTS CC

Customers are hesitant due to stigma of computer predicted values might not be accurtate.

5. AVAILABLE SOLUTIONS AS

Visit online websites to see how much other people with similar cars are selling their cars for.

By visiting dealerships and getting estimates.

Explore AS, differentiate

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

2. JOBS-TO-BE-DONE / PROBLEMS J&P

To build a supervised machine learning model that utilizes regression methods to accurately predict/anticipate the value of a Used car based on the following factors:

- Condition of the car
- Kilometers driven
- Life Span
- Damages
- No. of owners

9. PROBLEM ROOT CAUSE RC

The value proposed by dealers and other parties for a car may be untrustworthy and extremely low.

Users are unsure how much their can actually sell for or at a price which they can bid for.

7. BEHAVIOUR BE

Providing false claims on damages in and on the car.

To oversell non-existent features.

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

Identify strong TR & EM

3. TRIGGERS TR

Users may other sites to make a comparison which caters the decision process.

4. EMOTIONS: BEFORE / AFTER EM

Before:

The user might be concerned about the inaccurate prediction based on human assessment.

After:

without user intervention, the user may decide the attributes of the car on their own

10. YOUR SOLUTION SL

A machine learning model can be utilized to develop this system which can accurately predict the resale value of the car given a set of attributes of the car.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

Online:

Customers don't just look at the information provided by car brand websites but they also make a comparison study on pricings on various websites.

8.2 OFFLINE

Offline:

If an user is interested in buying a car. They would visit a lot of dealerships to get a quotation and do a comparison study.

Identify strong TR & EM

CHAPTER 4
REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning related applications. It can be installed on Windows, Linux, and macOS. Conda is an open-source, cross-platform, package management system. Anaconda comes with great tools like JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code.

For this project, we will be using **Jupyter notebook and Spyder**

To install Anaconda navigator and to know how to use Jupyter Notebook & Spyder using Anaconda watch the video

To build Machine learning models you must require the following packages

Sklearn : Scikit-learn is a library in Python that provides many unsupervised and supervised learning algorithms.

NumPy : NumPy is a Python package that stands for 'Numerical Python'. It is the core library for scientific computing, which contains a powerful n-dimensional array object

Pandas : pandas is a fast, powerful, flexible, and easy to use open-source data analysis and manipulation tool, built on top of the Python programming language.

Matplotlib : It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits

Flask : Web framework used for building Web applications.

1. Open anaconda prompt.
2. Type “pip install numpy” and click enter.

3. Type “pip install pandas” and click enter.
4. Type “pip install matplotlib” and click enter.
5. Type “pip install scikit-learn” and click enter.
6. Type “pip install Flask” and click enter.

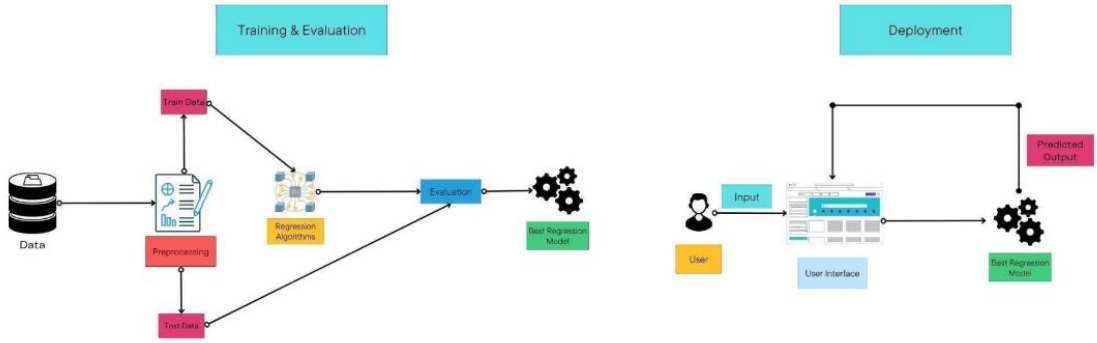
4.2 NON FUNCTIONAL REQUIREMENTS

NFR	NON-FUNCTIONAL REQUIREMENTS	DESCRIPTION
NFR-1	Usability	The application must be usable in all devices
NFR-2	Security	The application must protect user uploaded image
NFR-3	Reliability	The application must give an accurate result as much as possible
NFR-4	Performance	The application must be fast and quick to load up
NFR-5	Availability	The application must be available to use all the time
NFR-6	Scalability	The application must scale along with the user base

CHAPTER 5
PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

5.2 SOLUTION & TECHNICAL ARCHITECTURE



5.3 USER STORIES

Sprint	Functional Requirement (Epic)	User Story No.	User Story / Task	Story Points	Priority
Sprint1	Home Page	USN-1	As a user, I can view the home page of the web application.	20	Low
Sprint-2	Car resale value display	USN-2	As a user, I can be redirected to the data entry page	20	Medium
Sprint-3	Data Entry	USN-3	As a user, I can enter my car details in the required fields.	20	Medium

Sprint-4	Resale Value Prediction	USN-4	As a user, I expect the application to predict the resale value of my car.	20	Medium

CHAPTER 6
PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Dataset Reading and Preprocessing	USN-1	Cleaning the dataset end spliting to dependent and independentvariables	2	High	R.SIVASHUNMUGAN
Sprint-2	Building the Model	USN-2	Choosing the appropriate model for building and saving the modelas pickle file	1	High	G.M. SHAREETH KUMAR
Sprint-3	Application Building	USN-3	Using flask deploying the ML model	2	Medium	M.RAVINDHRAN
Sprint-4	Train the Model in ibm	USN-4	Finally train the model on IBM cloud and deploy the application	2	Medium	A. KRISHNAKANTH

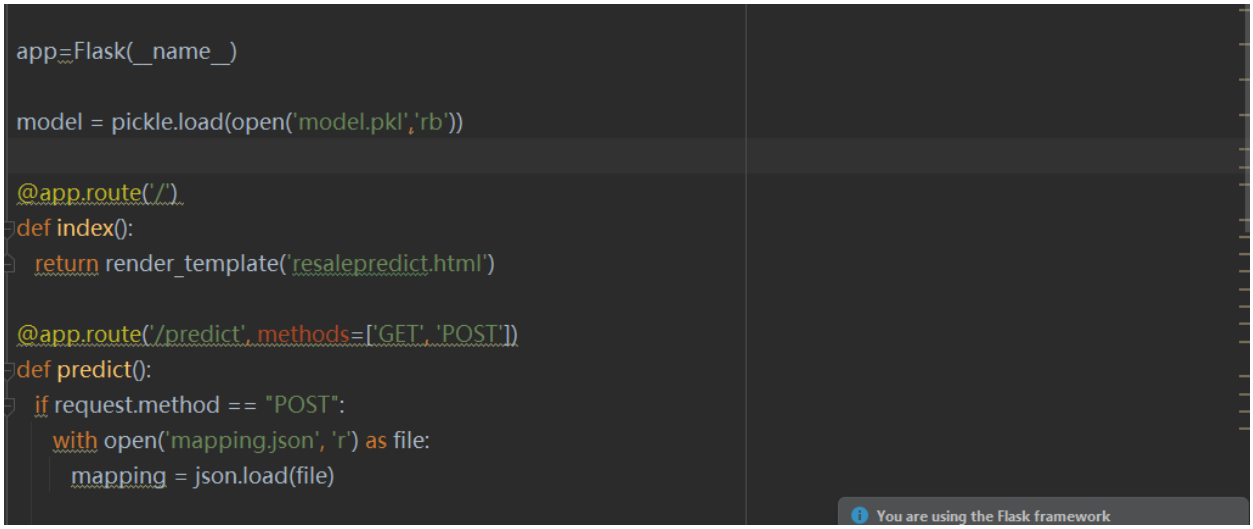
6.2 SPRINT DELIVERY SCHEDULE

SPRINT	TOTAL STORY POINTS	DURATION	SPRINT START DATE	SPRINT END DATE (PLANNED)	STORY POINTS COMPLETED (AS ON PLANNED DATE)	SPRINT RELEASE DATE (ACTUAL)
Sprint - I	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint - II	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint - III	20	6 Days	07 Oct 2022	12 Nov 2022	10	12 Nov 2022
Sprint - IV	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

CHAPTER 7

CODING &SOLUTIONING

```
import json
from flask import Flask, request, render_template
import pickle
import pickle
```



```
app=Flask(__name__)

model = pickle.load(open('model.pkl','rb'))

@app.route('/')
def index():
  return render_template('resalepredict.html')

@app.route('/predict', methods=['GET', 'POST'])
def predict():
  if request.method == "POST":
    with open('mapping.json', 'r') as file:
      mapping = json.load(file)
```

```
    vehicle_type = mapping['vehicleType'][request.form.get('vehicletype')]
    years_old = 2022 - int(request.form['regyear'])
    gearbox = mapping['gearbox'][request.form.get('gearbox')]
    powerps = float(request.form['powerps'])
    kms = float(request.form['kms'])
    fuelType = mapping['fuelType'][request.form.get('fuel')]
    brand = mapping['brand'][request.form.get('brand')]
    damage = mapping['notRepairedDamage'][request.form.get('damage').lower()]

    data = [[vehicle_type, years_old, gearbox, powerps, kms, fuelType, brand, damage]]
    pred = model.predict(data)
    print(pred)
return render_template('output.html', pred='The resale value predicted is ${:.2f}'.format(pred[0]))
```

CHAPTER 8

TESTING

8.1 TEST CASES

Test case ID	Feature Type	Component	Test Scenario	Expected Result	Actual Result	Status
HP_TC_001	UI	Home Page	Verify UI elements in the Home Page	The Home page must be displayed properly	Working as expected	PASS
HP_TC_002	UI	Home Page	Check if the UI elements are displayed properly in different screen sizes	The Home page must be displayed properly in all sizes	The UI is not displayed properly in screen size 2560 x 1801 and 768 x 630	FAIL

HP_TC_003	Functional	Home Page	Check if user can upload their file	The input image should be uploaded to the application successfully	Working as expected	PASS
HP_TC_004	Functional	Home Page	Check if user cannot upload unsupported files	The application should not allow user to select a non image file	User is able to upload any file	FAIL
HP_TC_005	Functional	Home Page	Check if the page redirects to the result page once the input is given	The page should redirect to the results page	Working as expected	PASS

BE_TC_001	Functional	Backend	Check if all the routes are working properly	All the routes should properly work	Working as expected	PASS
M_TC_001	Functional	Model	Check if the model can handle various image sizes	The model should rescale the image and predict the results	Working as expected	PASS
M_TC_002	Functional	Model	Check if the model predicts the digit	The model should predict the number	Working as expected	PASS
M_TC_003	Functional	Model	Check if the model can handle complex input image	The model should predict the number in the complex image	The model fails to identify the digit since the model is not built to handle such data	FAIL

RP_TC_001	UI	Result Page	Verify UI elements in the Result Page	The Result page must be displayed properly	Working as expected	PASS
RP_TC_002	UI	Result Page	Check if the input image is displayed properly	The input image should be displayed properly	The size of the input image exceeds the display container	FAIL
RP_TC_003	UI	Result Page	Check if the result is displayed properly	The result should be displayed properly	Working as expected	PASS
RP_TC_004	UI	Result Page	Check if the other predictions are displayed properly	The other predictions should be displayed properly	Working as expected	PASS

8.2 USER ACCEPTANCE TESTING

8.2.1 DEFECT ANALYSIS

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Total
By Design	1	0	1	0	2
Duplicate	0	0	0	0	0
External	0	0	2	0	2
Fixed	4	1	0	1	6
Not Reproduced	0	0	0	1	1
Skipped	0	0	0	1	1

Won't Fix	1	0	1	0	2
Total	6	1	4	3	14

8.2.2 TEST CASE ANALYSIS

Section	Total Cases	Not Tested	Fail	Pass
Client Application	10	0	3	7
Security	2	0	1	1
Performance	3	0	1	2
Exception Reporting	2	0	0	2

CHAPTER 9
RESULTS

9.1 PERFORMANCE METRICS

RMSE: Root Mean Squared Error is the metric used by us to evaluate our model. It is most commonly used to evaluate regression models. Root Mean Square Error as the name suggests is calculated as the root of the mean squared errors of the predicted values. The formula is given below:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \|y(i) - \hat{y}(i)\|^2}{N}},$$

Where zfi is the predicted value, zoi is the actual value and N is the sample size We have used the sklearn.metrics package to calculate the RMSE.

Our ensembled model had a RMSE value of 3545.68 and the individual model error rates are given below:

{
'mae': 1325.112086905962,
'mse': 9577053.62710202,
'rmse': 3094.6815065692977,
'rmsle': 8.03744027403009,
'r2': 0.8661221626879432,
'adj r2 score': 0.8661152969113608
}

The model is tested with the various damaged car images which is not used during the training and validation of the model which also shows that the model works with the accuracy of about 98% in the overall performance

CHAPTER 10
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.

CHAPTER 11
CONCLUSION

With the increased prices of new cars and the financial incapability of the customers to buy them, Used Car sales are on a global increase. Therefore, there is an urgent need for a Used Car Price Prediction system that effectively determines the worthiness of the car using a variety of features. The proposed system will help to determine the accurate price of a used car price prediction.

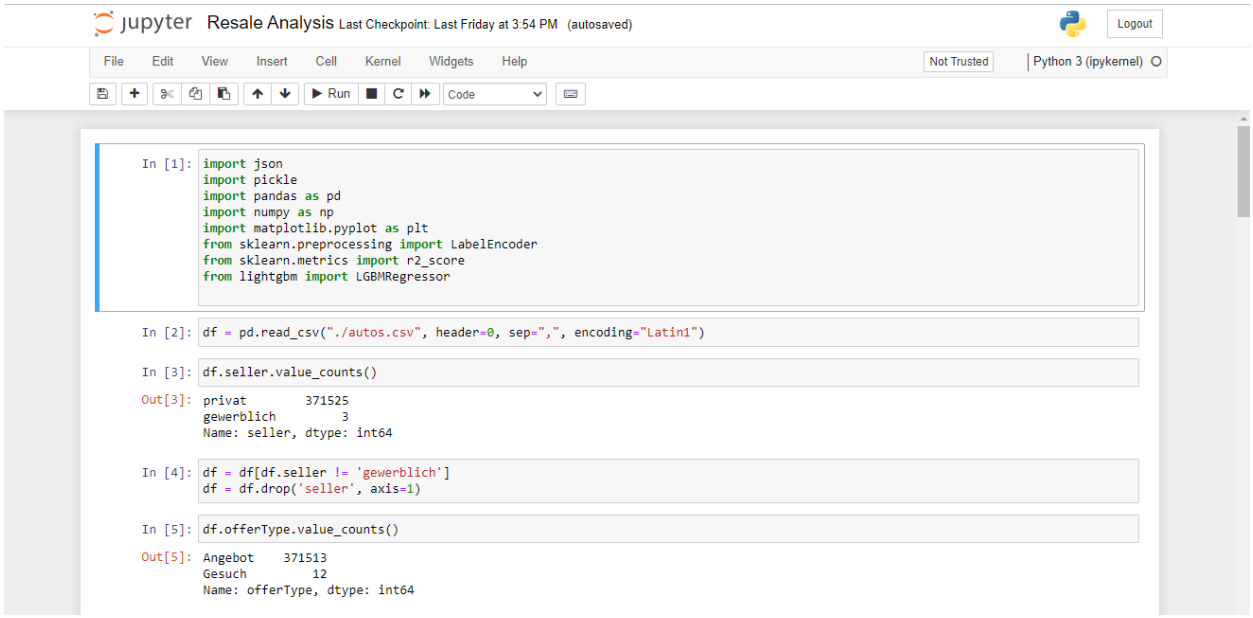
CHAPTER 12
FUTURE SCOPE

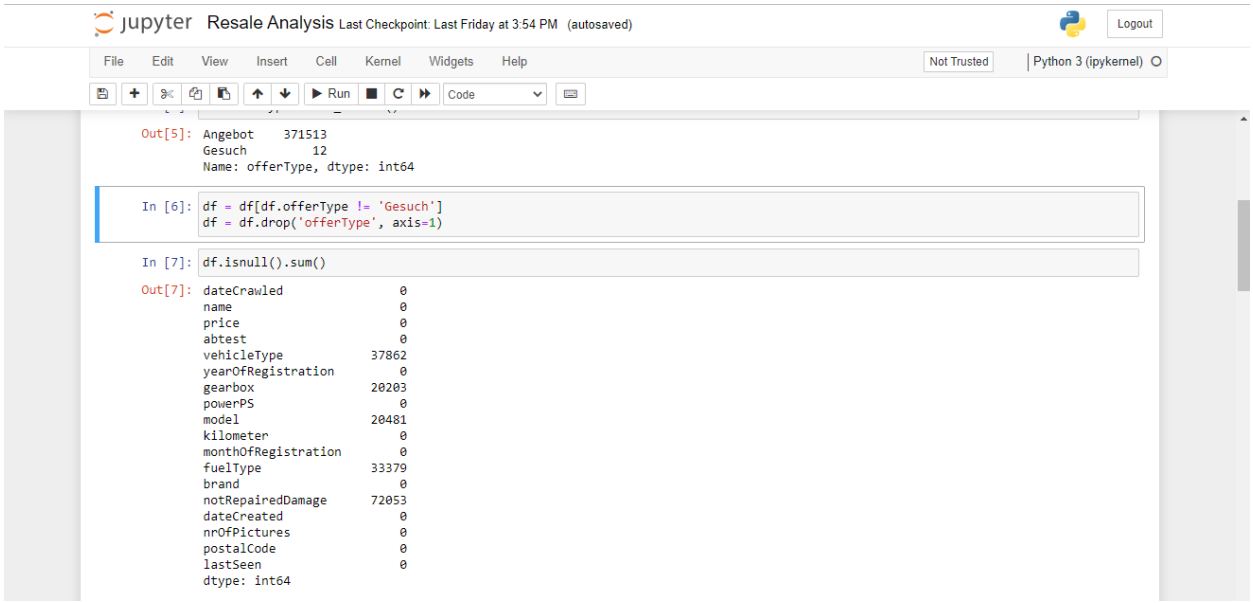
In the future this machine learning model may bind with various websites which can provide real-time data for price prediction. Also, we may add large historical data on car prices which can help to improve the accuracy of the machine learning model. We can build an android app as a user interface for interacting with users. For better performance, we plan to judiciously design deep learning network structures, use adaptive learning rates, and train on clusters of data rather than the whole dataset.

APPENDIX

SOURCE CODE

MODEL CREATION





jupyter Resale Analysis Last Checkpoint: Last Friday at 3:54 PM (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Run Code

In [8]:
```
df['yearsOld'] = 2022 - df['yearOfRegistration']
```

In [9]:
```
df = df[['price', 'vehicleType', 'yearsOld', 'gearbox', 'powerPS', 'kilometer', 'fuelType', 'brand', 'notRepairedDamage']]
```

In [10]:
```
df=df.dropna()
```

In [11]:
```
df = df.drop_duplicates([ 'price', 'vehicleType','gearbox', 'powerPS', 'kilometer', 'fuelType', 'brand','notRepairedDamage'])
```

In [12]:
```
df.gearbox.replace(('manuell', 'automatik'), ('manual', 'automatic'), inplace=True)
df.fuelType.replace(('benzin', 'andere', 'elektro'), ('petrol', 'others', 'electric'), inplace=True)
df.vehicleType.replace(('kleinwagen', 'cabrio', 'kombi', 'andere'),
                       ('compact', 'convertible', 'combination', 'others'), inplace=True)
df.notRepairedDamage.replace(('ja', 'nein'), ('Yes', 'No'),inplace=True)
```

In [13]:
```
df = df.sample(frac=0.093, random_state=42)
df
```

Out[13]:

	price	vehicleType	yearsOld	gearbox	powerPS	kilometer	fuelType	brand	notRepairedDamage
141976	12900	limousine	13	manual	120	60000	petrol	audi	No
293573	34000	limousine	12	automatic	0	150000	diesel	bmw	No
299514	1400	combination	19	automatic	136	150000	petrol	peugeot	No
76260	16800	combination	12	manual	184	150000	diesel	bmw	No
330662	3350	limousine	19	manual	73	5000	petrol	citroen	No
...	...	...	...	...	...	...	...	...	...

jupyter Resale Analysis Last Checkpoint: Last Friday at 3:54 PM (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Run Code

330662	3350	limousine	19	manual	73	5000	petrol	citroen	No
...	...	...	...	...	...	...	...	...	...
178696	4700	compact	11	manual	60	150000	petrol	volkswagen	No
123311	2500	combination	17	manual	105	150000	lpg	volkswagen	No
134075	17900	convertible	11	automatic	160	50000	petrol	volkswagen	No
192738	1300	combination	24	manual	0	150000	petrol	audi	Yes
357398	600	others	28	automatic	125	150000	petrol	honda	No

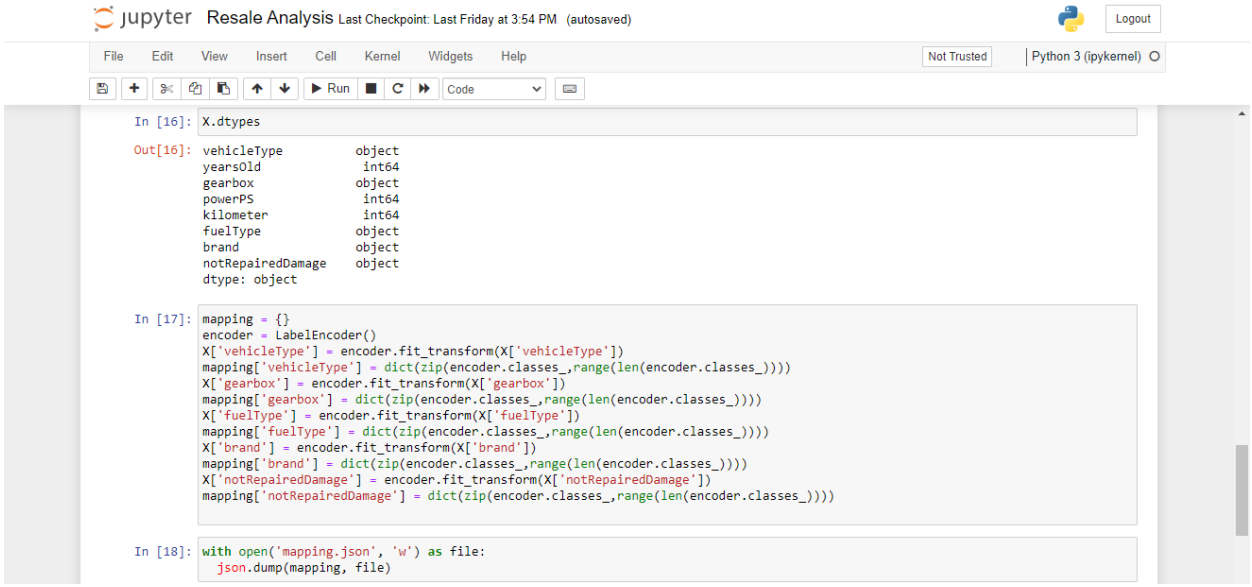
18215 rows × 9 columns

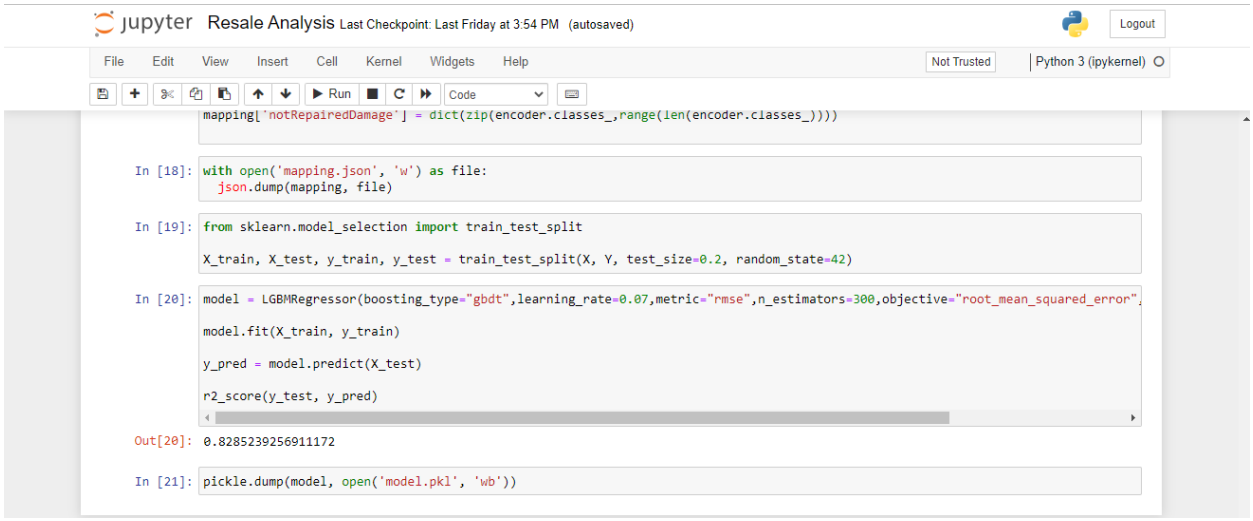
In [14]:
```
df.nunique(axis=0)
```

Out[14]:
```
price               1816
vehicleType            8
yearsOld              72
gearbox                2
powerPS              362
kilometer             13
fuelType               7
brand                 40
notRepairedDamage      2
dtype: int64
```

In [15]:
```
Y = df.price
X = df.drop("price", axis=1)
```

In [16]:
```
X.dtypes
```


jupyter Resale Analysis Last Checkpoint: Last Friday at 3:54 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [16]:
X.dtypes
Out[16]:
vehicleType object
yearsOld int64
gearbox object
powerPS int64
kilometer int64
fuelType object
brand object
notRepairedDamage object
dtype: object
In [17]:
mapping = {}
encoder = LabelEncoder()
X['vehicleType'] = encoder.fit_transform(X['vehicleType'])
mapping['vehicleType'] = dict(zip(encoder.classes_,range(len(encoder.classes_))))
X['gearbox'] = encoder.fit_transform(X['gearbox'])
mapping['gearbox'] = dict(zip(encoder.classes_,range(len(encoder.classes_))))
X['fuelType'] = encoder.fit_transform(X['fuelType'])
mapping['fuelType'] = dict(zip(encoder.classes_,range(len(encoder.classes_))))
X['brand'] = encoder.fit_transform(X['brand'])
mapping['brand'] = dict(zip(encoder.classes_,range(len(encoder.classes_))))
X['notRepairedDamage'] = encoder.fit_transform(X['notRepairedDamage'])
mapping['notRepairedDamage'] = dict(zip(encoder.classes_,range(len(encoder.classes_))))
In [18]:
with open('mapping.json', 'w') as file:
json.dump(mapping, file)


jupyter Resale Analysis Last Checkpoint: Last Friday at 3:54 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
mapping['notRepairedDamage'] = dict(zip(encoder.classes_,range(len(encoder.classes_))))
In [18]:
with open('mapping.json', 'w') as file:
json.dump(mapping, file)
In [19]:
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=42)
In [20]:
model = LGBMRegressor(boosting_type="gbdt",learning_rate=0.07,metric="rmse",n_estimators=300,objective="root_mean_squared_error",
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
r2_score(y_test, y_pred)
Out[20]:
0.8285239256911172
In [21]:
pickle.dump(model, open('model.pkl', 'wb'))

FLASK APP

```
app=Flask(__name__)

model = pickle.load(open('model.pkl','rb'))

@app.route('/')
def index():
  return render_template('resalepredict.html')

@app.route('/predict', methods=['GET', 'POST'])
def predict():
  if request.method == "POST":
    with open('mapping.json', 'r') as file:
      mapping = json.load(file)
```

You are using the Flask framework

RECOGNIZER

```
import json
from flask import Flask, request, render_template
import pickle
import pickle
```

20 3

```
app=Flask(__name__)

model = pickle.load(open('model.pkl','rb'))

@app.route('/')
def index():
  return render_template('resalepredict.html')

@app.route('/predict', methods=['GET', 'POST'])
def predict():
  if request.method == "POST":
    with open('mapping.json', 'r') as file:
      mapping = json.load(file)
```

You are using the Flask framework

```
    vehicle_type = mapping['vehicleType'][request.form.get('vehicletype')]
    years_old = 2022 - int(request.form['regyear'])
    gearbox = mapping['gearbox'][request.form.get('gearbox')]
    powerps = float(request.form['powerps'])
    kms = float(request.form['kms'])
    fuelType = mapping['fuelType'][request.form.get('fuel')]
    brand = mapping['brand'][request.form.get('brand')]
    damage = mapping['notRepairedDamage'][request.form.get('damage').lower()]

    data = [[vehicle_type, years_old, gearbox, powerps, kms, fuelType, brand, damage]]
    pred = model.predict(data)
    print(pred)
  return render_template('output.html', pred='The resale value predicted is ${:.2f}'.format(pred[0]))
```

HOME PAGE (HTML)

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

  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="{{url_for('static',filename='css/resaleintro.css')}}">
    <title>Document</title>
  </head>

  <body>

    <div class="main">
      <div id="bg"></div>
      <div id="head">
        <h1>Car Resale Value</h1>
      </div>
```

```
      <form action="/predict" method="post">
        <div id="row">
          <h3>Brand</h3>
          <select name="brand" required="required">
            <option value="" selected disabled hidden>choose</option>
            <option value="audi">Audi</option>
            <option value="bmw">BMW</option>
            <option value="jaguar">Jaguar</option>
            <option value="jeep">Jeep</option>
            <option value="honda">Honda</option>
            <option value="hyundai">Hyundai</option>
            <option value="chevrolet">Chevrolet</option>
            <option value="land_rover">Land_rover</option>
            <option value="nissan">Nissan</option>
            <option value="toyota">Toyota</option>
            <option value="volvo">Volvo</option>
            <option value="suzuki">Suzuki</option>
          </select>
        </div>
```

```
<div id="row">
  <h3>Kilometers Drived</h3>
  <input name="kms" type="number" placeholder="eg: 9999" required="required">
</div>

<div id="row">
  <h3> Fuel type</h3>
  <select name="fuel" required="required">
    <option value="" selected disabled hidden>choose</option>
    <option value="petrol">Petrol</option>
    <option value="diesel">Diesel</option>
    <option value="cng">CNG</option>
    <option value="lgp">LPG</option>
    <option value="electric">Electric</option>
  </select>
</div>
```

```
<div id="row">
  <h3>Vehicle Type</h3>
  <select name="vehicletype" required="required">
    <option value="" selected disabled hidden>choose</option>
    <option value="bus">Bus</option>
    <option value="combination">Combination</option>
    <option value="compact">Compact</option>
    <option value="coupe">Coupe</option>
    <option value="limousine">Limousine</option>
    <option value="suv">SUV</option>
  </select>
</div>

<div id="row">
  <h3>Year</h3>
  <input name="regyear" type="number" placeholder="eg: 2001" required>
</div>
```

```
<div id="row">
  <h3>Transmission Type</h3>
  <select name="gearbox" required="required">
    <option value="" selected disabled hidden>choose</option>
    <option value="manual">Manual</option>
    <option value="automatic">Automatic</option>
  </select>
</div>

<div id="row">
  <h3>Engine (cc)</h3>
  <input name="powerps" type="number" placeholder="eg: 1000" required>
</div>

<div id="row">
  <h3>Damage</h3>
  <input name="damage" placeholder="Yes or NO" required >

</div>
```

```
        <div id="row">
          <h3>Engine (cc)</h3>
          <input name="powerps" type="number" placeholder="eg: 1000" required>
        </div>

        <div id="row">
          <h3>Damage</h3>
          <input name="damage" placeholder="Yes or NO" required >

        </div>


        <button id="sub" type="submit ">Submit</button>
      </form>
    </div>
  </body>
</html>
```

HOME PAGE (CSS)

resaleintro.css - Notepad

File Edit Format View Help

```
* {
  padding: 0;
  margin: 0;
  font-family: "Times New Roman";
}

.main {
  height: 100vh;
  width: 100vw;
  display: flex;
  background-color: rgb(0, 0, 0);
  justify-content: center;
  align-items: center;
  flex-direction: column;
}

.main #bg{
  width: 100%;
  height: 100vh;
  position:absolute;
  background-color: rgb(0, 0, 0);
  background-image: url('../image/car2.jpg');
  background-repeat: no-repeat;
  background-size: cover;
  opacity: 0.3;
}

.main #head {
  position: absolute;
  top: 0;
  width: 100%;
  background-image: linear-gradient(to right,rgb(0, 0, 3),rgb(1, 1, 65),rgb(2, 2, 77));

  color: rgb(255, 255, 255);
}
```

```
.main form {
  z-index: 100;
}
.main #head h1 {
  padding: 7px 25px;
  font-size: 2rem;
}
.main h3 {
  color: white;
  font-weight: 600;
  width: 25rem;
}
.main #row {
  display: flex;
  margin-top: 1.5rem;
}
.main #row input {
  color: white;
  background-color: rgba(0, 0, 0, 0.4);
  margin-top: -7px;
  padding: 2px 10px;
  font-size: 15px;
  outline: none;
  border: 2px solid;
  height: 1.4rem;
  border-radius: 10px;
}
.main #row input:required {
  border: 2px solid rgb(0, 255, 128);
}
.main #row input:required:invalid {
  border: 2px solid rgb(255, 0, 0);
}
```

```
.main #row select {
  color: white;
  background-color: rgba(0, 0, 0, 0.4);
  padding: 5px;
  border-radius: 10px;
  outline: none;
}
.main #row select option {
  background-color: rgb(248, 77, 77);
  color: white;
}
.main #row select:required {
  border: 2px solid rgb(0, 255, 128);
}
.main #row select:required:invalid {
  border: 2px solid rgb(255, 0, 0);
}
.main #sub {
  font-size: 1rem;
  background-color: black;
  color: white;
  font-weight: 700;
  letter-spacing: 2px;
  padding: 10px 30px;
  position: relative;
  margin-top: 5rem ;
  margin-left: 35%;
  border-radius: 10px;
  outline: none;
  border: 1px solid rgb(0, 119, 255);
}


.main #sub:hover {
  background-color: rgb(4, 2, 107);
  color: rgb(255, 255, 255);
  border: 1px solid rgb(254, 255, 255);
}
```

PREDICT PAGE (HTML)

```
<html>
  <head>
    <title>predicted value</title>
    <style>
      * {
        padding: 0;
        margin: 0;
      }

      body {
        height: 100vh;
        width: 100vw;
        background-color: black;
      }

      #bg {
        height: 100%;
        width: 100%;
```

```
        background-image: url('static/image/car.jpg');
        background-size: cover;
        opacity: 0.5;
      }

      h2 {
        font-size: 1.7rem;
        color: #ffffff;
        height: 6rem;
        letter-spacing: 2px;
        width:24rem ;
        position: absolute;
        top: 35%;
        left:3rem;
        line-height: 4rem;
      }
```

```
#head {
  position: absolute;
  top: 0;
  width: 100%;
  background-image: linear-gradient(to right,rgb(0, 0, 3),rgb(1, 1, 63),rgb(4, 4, 117));

  color: rgb(255, 255, 255);
 }


#head h1 {
  padding: 7px 25px;
  font-size: 2rem;
}
```

```
  </style>

</head>
<body>
  <div id="bg"></div>
  <div id="head">
    <h1>Car Resale Value</h1>
  </div>
  <h2>{{ pred }}</h2>
</body>
</html>
```

GITHUB

https://github.com/IBM-EPBL/IBM-Project-1891-1658419826.git

PROJECT DEMO

https://www.dropbox.com/s/tjr2ijts3jjmhbg/Project%20Demo.mp4?dl=0