Car Resale Value prediction

Team ID: PNT2022TMID34636

Team Leader: Hari Sankar Raj S

Team Members: Annes Belmin S , Anand S , Amirtha Jeslin J

Project Report Format

1. **INTRODUCTION**
    a. Project Overview
    b. Purpose
2. **LITERATURE SURVEY**
    a. Existing problem
    b. References
    c. Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
    a. Empathy Map Canvas
    b. Ideation & Brainstorming
    c. Proposed Solution
    d. Problem Solution fit
4. **REQUIREMENT ANALYSIS**
    a. Functional requirement
    b. Non-Functional requirements
5. **PROJECT DESIGN**
    a. Data Flow Diagrams
    b. Solution & Technical Architecture
    c. User Stories
6. **PROJECT PLANNING & SCHEDULING**
    a. Sprint Planning & Estimation
    b. Sprint Delivery Schedule
7. **CODING & SOLUTIONING**
8. **TESTING**
    a. Test Cases
    b. User Acceptance Testing
9. **RESULTS**
    a. Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**

Source Code

GitHub & Project Demo Link

1.INTRODUCTION

In this project we have used different algorithms with different techniques for developing Car resale value prediction systems considering different features of the car. In a nutshell, car resale value prediction helps the user to predict the resale value of the car depending upon various features like kilometers driven, fuel type, etc.

1.1. PROJECT OVERVIEW

This resale value prediction system is made for general purpose to just predict the amount that can be roughly acquired by the user. We try to predict the amount of resale by best 70% accuracy so the user can get estimated value before he resales the car and doesn't make a deal in loss.

1.2. PURPOSE

The main idea of making a car resale value prediction system is to get hands-on practice for python using Data Science. Car resale value prediction is the system to predict the amount of resale value based on the parameters provided by the user. User enters the details of the car into the form given and accordingly the car resale value is predicted.Car resale value prediction system is made with the purpose of predicting the correct valuation of used cars that helps users to sell the car remotely with perfect valuation and without human intervention in the process to eliminate biased valuation

2. LITERATURE SURVEY

The first paper predicts Pre-owned cars or so-called used cars have capacious markets across the globe.Before acquiring a used car, the buyer should be able to decide whether the price affixed for the car is genuine. Several facets including mileage, year, model, make, run and many more are needed to be considered before getting a hold of any pre-owned car. Both the seller and the buyer should have a fair deal. This paper presents a system that has been implemented to predict a fair price for any pre-owned car.

The second paper shows how to develop a model that can anticipate fair used car pricing based on a variety of factors such as vehicle model, year of manufacture, fuel type, Price, km driven. In the used car market, this strategy can benefit vendors, purchasers, and car manufacturers. It can then produce a reasonably accurate price estimate based on the data that users provide. Machine learning and data science are used in the model-building process. The data was taken from classified ads for second hand autos. To attain the maximum accuracy, the researchers used a variety of regression approaches, including linear regression, polynomial regression, support vector regression, decision tree regression, and random forest regression.

The purpose of third paper is to investigate the application of supervised machine learning techniques to predict the price of used cars in Mauritius. The predictions are based on historical data collected from newspaper on daily basis. The predictions are then evaluated and compared to find those which provide the best performance. Predicting the resale value of car is not a simple task because the value of used cars depend on a number of factors. The most important ones are usually the age of the car, its make (and model), the origin of the car (the original country of the manufacturer), its mileage (the number of kilometers it has run) and its horsepower.

The fourth paper shows due to 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 which effectively determines the worthiness of the car using a variety of features. Existing System includes a process where a seller decides a price randomly and buyer has no idea about the car and it's value in the present day scenario.To overcome this problem we have developed a model which will be highly effective. Regression Algorithms are used because they provide us with continuous value as an output and not a categorized value.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 fifth paper on the development of machine learning models that can accurately predict the price of a used car based on its features, in order to make informed purchases. They have implemented and evaluated various learning methods on a dataset consisting of the sale prices of different makes and models across cities in the United States.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. Based on existing data, the aim is to use machine learning algorithms to develop models for predicting used car prices.

2.1. EXISTING PROBLEM

Online used car trading platforms have developed rapidly, but they still face many problems. In practice, institutions and individuals differ in how they screen the characteristic variables of used car prices and predict used car prices. Under such conditions, it is easy to lead to the unsound development of the market, and it is difficult to establish a unified evaluation system, which causes great difficulties in the transaction of used cars. In terms of theory, traditional used car price evaluation methods rely too much on the subjective judgment of evaluators, which can no longer meet the needs of online transactions in the used car market. Therefore, it is necessary to establish an efficient, reasonable, fair, and accurate used car price evaluation system. This paper analyzes the factors affecting the price of used cars from three aspects— used car parameters, vehicle condition factors, and transaction factors—and establishes a used car price evaluation system including 12 characteristic variables. In addition, in view of the fact that the prediction accuracy of high-end used cars is lower than that of low- end used cars, it is suggested that when pricing high-end used cars, you need to check other configuration information in order to make a more reasonable judgment.

2.2. REFERENCE

1. SAMEERCHAND PUDARUTH, Computer Science and Engineering Department, University of Mauritius, Reduit, Mauritius.
2. Praful Rane, Deep Pandya, Dhawal Kotak, Information Technology Engineering, Padmabhushan Vasantdada Patil Pratisthan College of Engineering, Maharashtra India.
3. Kshitij Kumbar, Pranav Gadre and Varun Nayak.
4. Chetna Longania, Sai Prasad Potharaju, Sandhya Deore, Dept of Computer Engineering, Sanjivani College of Engineering, Kopargaon, Maharashtra, India.

2.3. PROBLEM STATEMENT DEFINITION

The main aim of this project is to predict the price of used cars using the various Machine Learning (ML) models. This can enable the customers to make decisions based on different inputs or factors namely

• Brand or Type of the car one prefers like Ford, Hyundai

• Model of the car namely Ford Figo, Hyundai Creta

• Location like Delhi, Chennai, Mumbai

• Year of manufacturing like 2020, 2021

• Type of fuel namely Petrol, Diesel

• Price range or Budget

• Type of transmission which the customer prefers like Automatic or Manual

• Mileage

3. IDEATION & PROPOSED SOLUTION

Prediction helps companies to make plan and procedure for success. Resale of cars occupy a major part in sales economy. It regards several factors such as age of car, service reports, fuel type, brand, vehicle type, kilometers travelled, mileage, gearbox, number of owners, battery conditions and considering engine condition and insurance. The prediction using these factors suggests the final cost.The overall solution is to predict the resale value price and display it to the customers. There are two phases: first is to collect datasets, pre process the datasets, testing and training the models. Second is to create webpage to display the solution as a customized GUI. The user will be asked to enter the data like model, color, kilometers travelled, fuel type. If the user clicks the predict button the predicted value will be displayed.

3.1. EMPATHY MAP CANVAS

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviors and attitudes. It is a useful tool to helps teams better understand their users.
Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

3.2. IDEATION & BRAIN STORMING

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 amount of creative solutions.

3.3. PROPOSED SOLUTION

Sale prediction helps to attain a greater efficiency for customers. It helps in improving pricing, advertising and product development. It helps customers to know about the details of car before buying. Both the seller and buyer will be benefited. This model should provide accurate price. This should contain details like images of interior and exterior of car, service details, kilometer travelled details, etc. The user should be able to contact with the historical owner through the website.

3.4. PROBLEM SOLUTION FIT

This system is built by using the Machine Learning and Regression model. By using this system we can predict the resale value of the car at any time any where.

4. REQUIREMENT ANALYSIS

The car manufacturing has been increasing swiftly over the years during past decade, with about 92 million cars that were manufactured in 2019. This provides a big boost for the market of old and used cars which is now coming up as a progressively growing industry. The recent entries of various websites and web-portals have fulfilled the requirements of customers up to some extent as they now know the present trends and scenario to get the market value of any old vehicle present in the market. Machine Learning has a lot of applications in real world scenario but one of the most known application is the use of Machine Learning in resolving the prediction problems. The project being discussed here is very much based upon one among such applications. Employing various Machine Learning Algorithms, we will try and build a statistical model based upon given data and features set to estimate the prices of used cars.

4.1. FUNCTIONAL REQUIREMENT

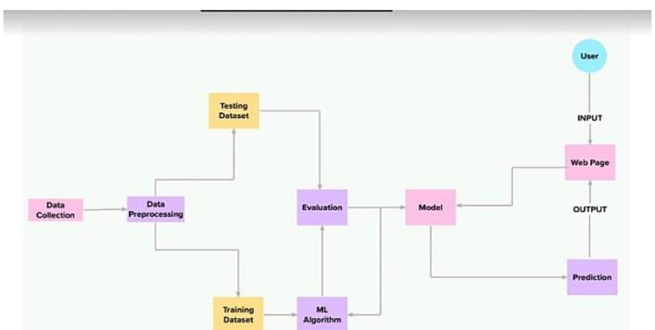
- ✓ User Registration
- ✓ User Confirmation
- ✓ Car registration
- ✓ Value Prediction

4.2. NON-FUNCTIONAL REQUIREMENT

- ✓ Usability
- ✓ Security ✓
- Reliability ✓
- Availability✓
- Scalability

5. PROJECT DESIGN

5.1. DATA FLOW DIAGRAMS



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, behavior, 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.

5.3. USER STORIES

In this topic the type of user, functional requirement, User story number, User story or task, Acceptance criteria, Priority, Release including data are collected together.

6. PROJECT PLANNING & SCHEDULING

'Project planning' is all about choosing and designing effective policies and methodologies to attain project objectives. While 'Project scheduling' is a procedure of assigning tasks to get them completed by allocating appropriate resources within an estimated budget and time-frame.

6.1. SPRINT PLANNING AND ESTIMATION

In agile development, the product owner is tasked with prioritizing the backlog — the ordered list of work that contains short descriptions of all desired features and fixes for a product. Product owners capture requirements from the business, but they don’t always understand the details of implementation. So good estimation can give the product owner new insight into the level of effort for each work item, which then feeds back into their assessment of each item's relative priority.

6.2. SPRINT DELIVERY SCHEDULE

Sprint planning and scheduling revolves around a product backlog, which is a list of available requests for iteration and development. A product backlog typically contains a variety of requests or user stories from stakeholders like customers, partners, and team members. During sprint scheduling, the product owner and agile development team analyze the product backlog and determine the number of requests they can deliver. Product backlogs continuously evolve as stakeholders generate new requests. That being the case, it’s critical to keep a close watch on the backlog to avoid letting it get too large and unmanageable.

It’s necessary to set a firm time frame for each sprint. Sprints tend to be uniform in duration throughout each phase of an agile development project. What you want to avoid is having projects with short, medium, and long sprints. To this end, the best approach is to figure out a time frame that works for everyone and then assign individual tasks accordingly.

Part of managing sprints involves knowing when to be firm about deadlines and when to extend time frames to accommodate sudden changes or unexpected issues. Keep in mind that if you consistently break deadlines, it can impact the time frame of an agile project, setting a launch back weeks or even months beyond its target date. When you boil it down, agile development is all about working quickly.

7.CODING & SOLUTIONING

```
#import libraries
import pandas as pd
import numpy as np
import matplotlib as plt
from sklearn.preprocessing import LabelEncoder import
pickle

# read the dataset

df = pd.read_csv("Data/autos.csv", header=0, sep=',', encoding='Latin1',) df

# clean the dataset

df=df.drop('offerType',axis=1) df=df[(df.powerPS > 50)
& (df.powerPS < 900)]
df = df[(df.yearOfRegistration >= 1950) & (df.yearOfRegistration < 2017)]
df.drop(['name', 'abtest', 'dateCrawled', 'nrOfPictures', 'lastSeen',
          'postalCode','dateCreated'], axis='columns',inplace=True) new_df =
df.copy()
new_df = new_df.drop_duplicates ([ 'price', 'vehicleType',
'yearOfRegistration'
                                        ,'gearbox', 'powerPS', 'model',
'kilometer', 'monthOfRegistration', 'fuelType'
                                        ,'notRepairedDamage'])
new_df.gearbox.replace(('manuell', 'automatik'), ('manual', 'automatic'), inplace=True)
new_df.fuelType.replace(('benzin', 'andere', 'elektro'), ('petrol', 'others',
'electric'), inplace=True) new_df.vehicleType.replace(('kleinwagen', 'cabrio',
'komb, 'andere'),
                                ('small car', 'convertible', 'combination',
'others'), inplace=True) new_df.notRepairedDamage.replace(('ja',
'nein'), ('Yes', 'No'),inplace=True)
new_df = new_df[(new_df.price >= 100) & (new_df.price <= 150000)]
new_df['notRepairedDamage'].fillna(value='not-declared', inplace=True) new_df[
'fuelType'].fillna(value='not-declared', inplace=True)
new_df[ 'gearbox'].fillna(value='not-declared', inplace=True) new_df[
'vehicleType'].fillna (value='not-declared', inplace=True)
new_df['model'].fillna(value='not-declared',inplace=True)
```

```
new_df.to_csv("autos_preprocessed.csv")

#import libraries
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import pickle
from lightgbm import LGBMRegressor #read

preprocessed data

data = pd.read_csv("autos_preprocessed.csv")

#metrics evaluation

def find_scores(Y_actual, Y_pred, X_train): scores
    = dict()
    mae = mean_absolute_error(Y_actual, Y_pred) mse =
    mean_squared_error(Y_actual, Y_pred) rmse =
    np.sqrt(mse)
    rmsle = np.log(rmse)
    r2 = r2_score(Y_actual, Y_pred) n, k
    = X_train.shape
    adj_r2_score = 1 - ((1-r2)*(n-1)/(n-k-1))

    scores['mae']=mae
    scores['mse']=mse
    scores['rmse']=rmse
    scores['rmsle']=rmsle
    scores['r2']=r2
    scores['adj_r2_score']=adj_r2_score return

    scores

#testing abd training

X = labeled.iloc[:,1:].values
Y = labeled.iloc[:,0].values.reshape(-1,1)
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.4,
random_state=42)
```

```
model = LGBMRegressor(boosting_type="gbdt",learning_rate=0.07,metric="rmse",n_esti
mators=300,objective="root_mean_squared_error",random_state=42,reg_sqrt=Tr ue)

model.fit(X_train, Y_train) Y_pred =

model.predict(X_test)

find_scores(Y_test, Y_pred, X_train) #save

the model

pickle.dump(model, open('resale_model.sav', 'wb')) BUILD A

FLASK APP

# Import Libraries
import pandas as pd
import numpy as np
from flask import Flask, render_template, Response, request import
pickle
from sklearn.preprocessing import LabelEncoder app =

Flask(_name_)#initiate flask app

def load_model(file='resale_model.sav'):#load the saved model return
	pickle.load(open(file, 'rb'))

@app.route('/')
def index():#main page
	return render_template('car.html')

@app.route('/predict_page')
def predict_page():#predicting page return
	render_template('value.html')

@app.route('/predict',    methods=['GET','POST'])    def
predict():
	reg_year = int(request.args.get('regyear')) powerps
	=       float(request.args.get('powerps'))       kms=
	float(request.args.get('kms'))
	reg_month = int(request.args.get('regmonth'))
```

```
        gearbox = request.args.get('geartype') damage
        = request.args.get('damage') model =
        request.args.get('model') brand =
        request.args.get('brand')
        fuel_type = request.args.get('fuelType') veh_type
        = request.args.get('vehicletype')

        new_row    =    {'yearOfReg':reg_year,    'powerPS':powerps,    'kilometer':kms,
                            'monthOfRegistration':reg_month,          'gearbox':gearbox,
                            'notRepairedDamage':damage,
                            'model':model,      'brand':brand,      'fuelType':fuel_type,
                            'vehicletype':veh_type}

        print(new_row)

        new_df = pd.DataFrame(columns=['vehicletype','yearOfReg','gearbox',

'powerPS','model','kilometer','monthOfRegistration','fuelType',
              'brand','notRepairedDamage'])
        new_df = new_df.append(new_row, ignore_index=True) labels =
['gearbox','notRepairedDamage','model','brand','fuelType','vehicletype'] mapper = {}

        for i in labels:
              mapper[i] = LabelEncoder()
              mapper[i].classes = np.load(str('classes'+i+'.npy'),
allow_pickle=True)
              transform = mapper[i].fit_transform(new_df[i])
              new_df.loc[:,i+'_labels'] = pd.Series(transform,
index=new_df.index)
        labeled =
new_df[['yearOfReg','powerPS','kilometer','monthOfRegistration'] + [x+'_labels'
for x in labels]]

        X = labeled.values.tolist()
        print('¥n¥n', X)
        predict = reg_model.predict(X)


        #predict = predictions['predictions'][0]['values'][0][0]
        print("Final prediction :",predict)
```

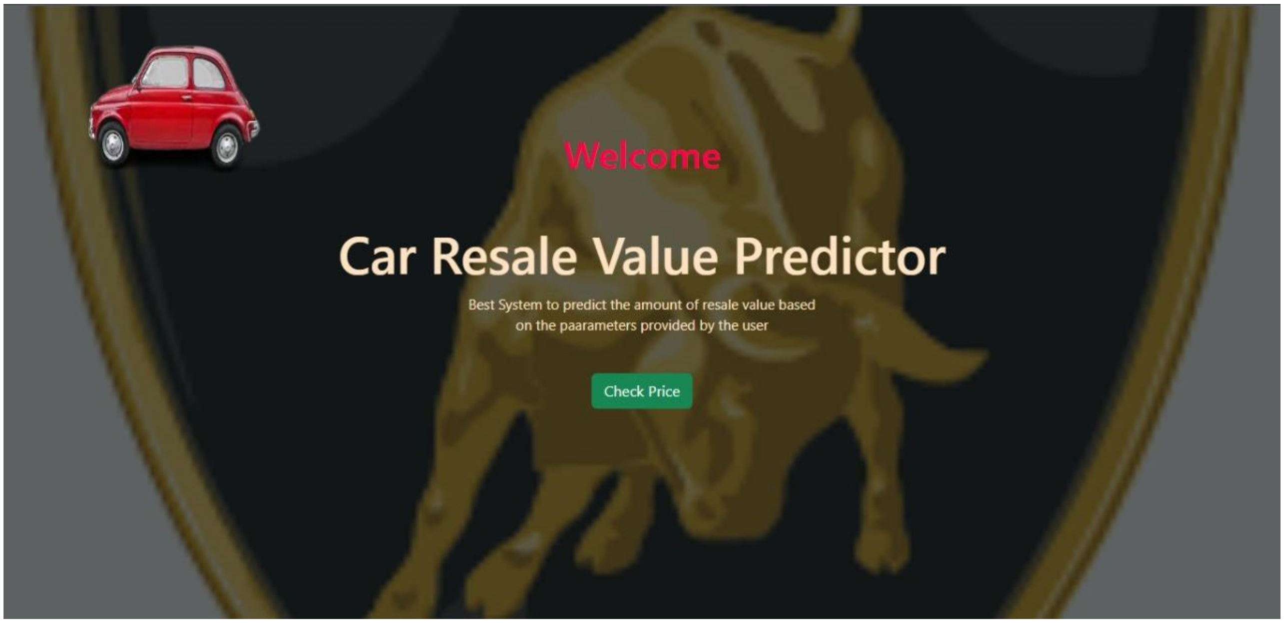
```
        return render_template('predict.html',predict=predict)

if_name_=='_main__':
        reg_model = load_model()#load the saved model
        app.run(debug=True)
```

8.TESTING

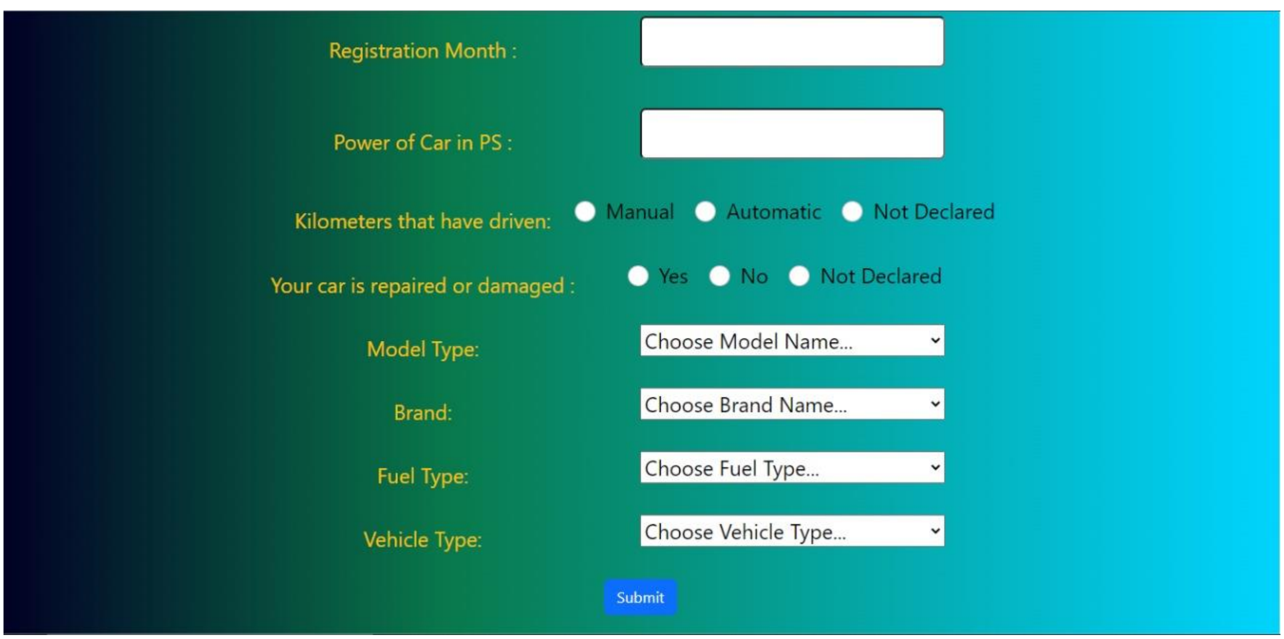
8.1. TEST CASES

1. User Login and Registration test
2. Database Update test
3. Prediction test

8.2. USER ACCEPTANCE TESTING




Get the Accurate Resale Value of Your Car
Registration Year :
Registration Month :
Power of Car in PS :
Kilometers that have driven:
Manual
Automatic
Not Declared
Your car is repaired or damaged :
Yes
No
Not Declared
Model Type:
Choose Model Name...
Brand:
Choose Brand Name...
Choose Fuel Type...


Registration Month :
Power of Car in PS :
Kilometers that have driven:
Manual
Automatic
Not Declared
Your car is repaired or damaged :
Yes
No
Not Declared
Model Type:
Choose Model Name...
Brand:
Choose Brand Name...
Fuel Type:
Choose Fuel Type...
Vehicle Type:
Choose Vehicle Type...
Submit

10.ADVANTAGES & DISADVANTAGES

MERITS:

- ✓ Highly effective
- ✓ Time efficiency

- ✓ Less power consumption
- ✓ More information obtained

DEMERITS:

- ✓ Not accurate
- ✓ Not effective

10. CONCLUSION

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 which effectively determines the worthiness of the car using a variety of features. The proposed system will help to determine the accurate price of used car price prediction. This project compares 2 different algorithms for machine learning : Decision tree, Random forest.

11. FUTURE SCOPE

In future this machine learning model may bind with various website which can provide real time data for price prediction. Also we may add large historical data of car price which can help to improve accuracy of the machine learning model. We can build an android app as user interface for interacting with user. 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.

12.APPENDIX

app.py

```
# Import Libraries
import pandas as pd
import numpy as np
from flask import Flask, render_template, Response, request import
pickle
from sklearn.preprocessing import LabelEncoder
```

```
app = Flask(_name_)#initiate flask app

def load_model(file='resale_model.sav'):#load the saved model return
      pickle.load(open(file, 'rb'))

@app.route('/')
def index():#main page
      return render_template('car.html')

@app.route('/predict_page')
def predict_page():#predicting page return
      render_template('value.html')

@app.route('/predict',    methods=['GET','POST'])    def
predict():
      reg_year = int(request.args.get('regyear')) powerps
      =      float(request.args.get('powerps'))      kms=
      float(request.args.get('kms'))
      reg_month = int(request.args.get('regmonth'))

      gearbox = request.args.get('geartype') damage
      = request.args.get('damage') model =
      request.args.get('model') brand =
      request.args.get('brand')
      fuel_type = request.args.get('fuelType') veh_type
      = request.args.get('vehicletype')

      new_row    =    {'yearOfReg':reg_year,    'powerPS':powerps,    'kilometer':kms,
                         'monthOfRegistration':reg_month,          'gearbox':gearbox,
                         'notRepairedDamage':damage,
                         'model':model,      'brand':brand,      'fuelType':fuel_type,
                         'vehicletype':veh_type}

      print(new_row)

      new_df = pd.DataFrame(columns=['vehicletype','yearOfReg','gearbox',

'powerPS','model','kilometer','monthOfRegistration','fuelType',
             'brand','notRepairedDamage'])
      new_df = new_df.append(new_row, ignore_index=True) labels =
['gearbox','notRepairedDamage','model','brand','fuelType','vehicletype'] mapper = {}
```

```
        for i in labels:
                mapper[i] = LabelEncoder()
                mapper[i].classes = np.load(str('classes'+i+'.npy'),
allow_pickle=True)
                transform = mapper[i].fit_transform(new_df[i])
                new_df.loc[:,i+'_labels'] = pd.Series(transform,
index=new_df.index)
        labeled =
new_df[['yearOfReg','powerPS','kilometer','monthOfRegistration'] + [x+'_labels'
for x in labels]]

        X = labeled.values.tolist()
        print('¥n¥n', X)
        predict = reg_model.predict(X)


        #predict = predictions['predictions'][0]['values'][0][0]
        print("Final prediction :",predict)

        return render_template('predict.html',predict=predict)  if_name_

=='_main__':
        reg_model = load_model()#load the saved model
        app.run(debug=True)
```

SOURCECODE:
https://github.com/IBM-EPBL/IBM-Project-49591-1660827990/tree/main/Amirtha Jeslin/Application Building

GITHUB :
https://github.com/IBM-EPBL/IBM-Project-49591-1660827990

PROJECT DEMO LINK:
https://drive.google.com/drive/folders/1qlN7GQDXHwIj0c1pYmhcJVdC8y5TPafY?usp=share_link