CAR RESALE VALUE PREDICTION

1 INTRODUCTION

1.1 Overview

This document is to chronicle about the model builded so as to serve as a guide to the developers on one hand and a record to understand about the model. The project is to predict the amount or the value for car that are resaled. We have builded a model and have done the interface using the local host. This model has given good accuracy and predicts the resale value very well.

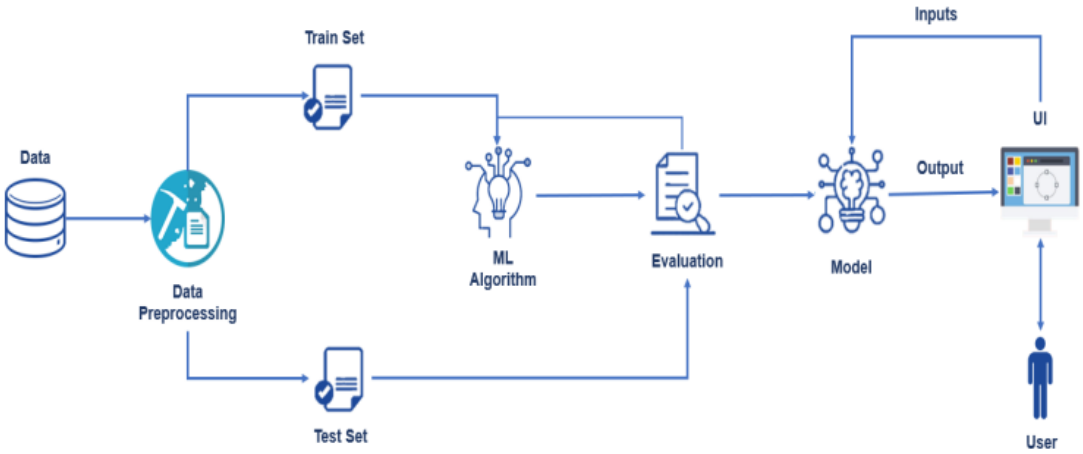
1.2 Purpose

The main purpose of this project is to predict the resale value for the cars in the market. 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.

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.

3 THEORITICAL ANALYSIS

3.1 Block diagram



3.2 Hardware / Software designing

The hardware requirements of this project include a system Processor 64-bit, four-core, 2.5 GHz minimum per core, RAM 4GB minimum,Harddisk 80 GB and a stable Ethernet Connection.

The software requirements includes UI. A user Interface Application will be accessed through a Browser Interface. The interface would be viewed best using 1024 x 768 and 800 x 600 pixels resolution setting. The software would be fully compatible with modern Web browsers like Google Chrome, Microsoft Edge and Mozilla Firefox.

For this project we need Anaconda (IDLE / Spyder / PyCharm)(Python 3.7) and other packages are also needed to be installed which include Tensorflow(this package is used as backend support to Keras),Keras(this package is used for building Neural Network layers),opencv(this package is used for image processing),Flask(To build a web application)

4 EXPERIMENTAL INVESTIGATIONS

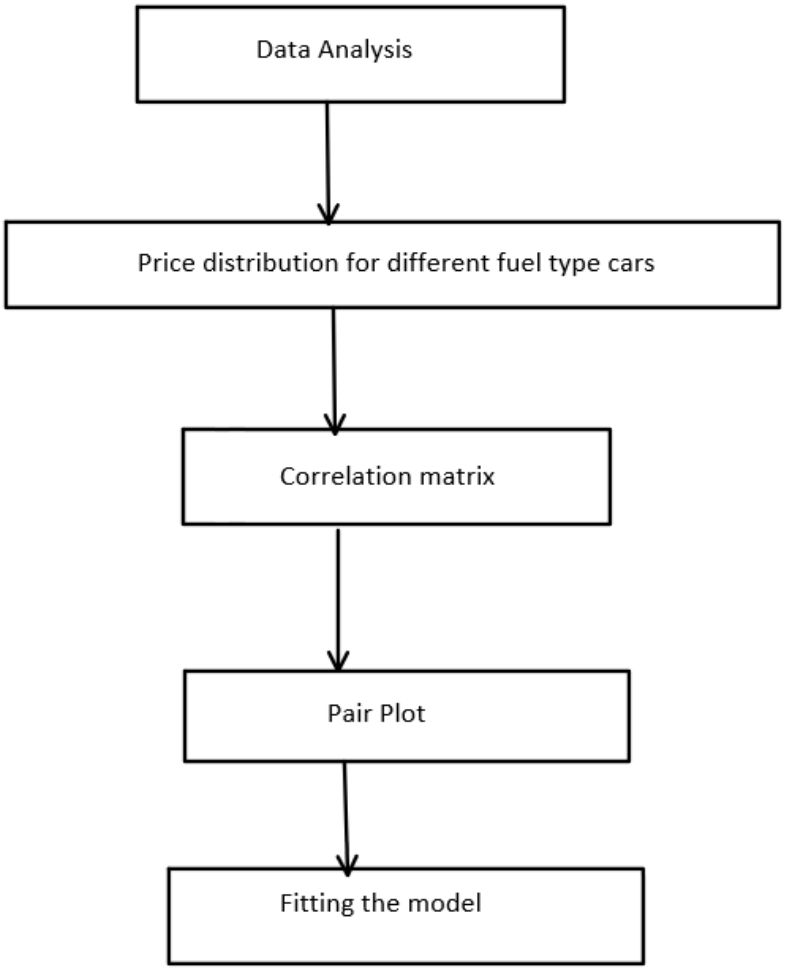
The main idea of making a car resale value prediction system is to get hands-on practice for python using Data Science. Car resale value prediction is the system to predict the amount of resale value based on the parameters provided by the user. User enters the details of the car into the form given and accordingly the car resale value is predicted.

- Fuel Type
- Manufacturing Year
- Miles Driven
- Number of Historical Owners

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 labelled data where our target or dependent variable is the selling price of a car. All other features are independent variables. Maintenance Record Following are some regression algorithms that can be used for predicting the selling price.

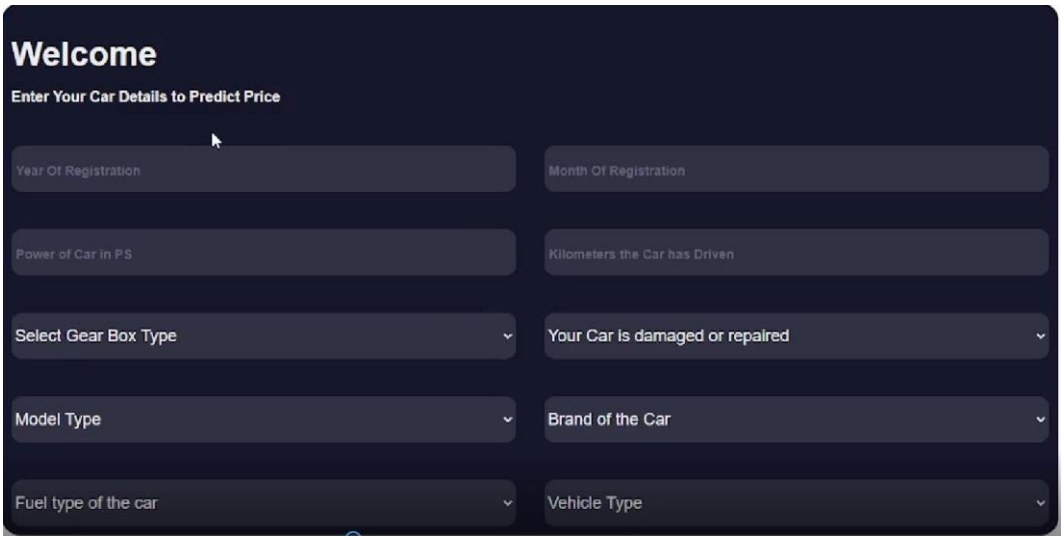
- Linear Regression
- Decision Tree Regressor
- Support Vector Regressor
- KNN Regressor
- Random Forest Regressor

5 FLOWCHART



6 RESULT

This image is the user interface that the user sees.



7 ADVANTAGES & DISADVANTAGES

Some of the advantages of the proposed model include:

i)The iceberg model helps to probe the underlying causes of events and patterns and the issues involved in any change process.

ii)This model predicts the value with great accuracy.

iii)With the help of this solution we are able to predict the value or the amount of the car in the market.

iv) It is centered on the concept of adaptability and focuses on making better efficiency of the values and the prices in the market in the future.

DISADVANTAGE:

We find no disadvantage in the model proposed but certain future enhancements can be made.

8 APPLICATIONS

i) It would be of great help to the future market and transportation team in all the countries.

ii) It could bring a whole new for the transportation industry to predict the price value of this kind.

9 CONCLUSION:

The prediction error rate of all the models was well under the accepted 5% of error. But, on further analysis, the mean error of the regression tree model was found to be more than the mean error rate of the multiple regression and lasso regression models. Even though for some seeds the regression tree has better accuracy, its error rates are higher for the rest. This has been confirmed by

performing an ANOVA. Also, the post-hoc test revealed that the error rates in multiple regression models and lasso regression models aren't significantly different from each other. To get even more accurate models, we can also choose more advanced machine learning algorithms such as random forests, an ensemble learning algorithm which creates multiple decision/regression trees, which brings down overfitting massively or Boosting, which tries to bias the overall model by weighing in the favour of good performers. More data from newer websites and different countries can also be scraped and this data can be used to retrain these models to check for reproducibility.

10 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.

11 BIBILOGRAPHY

The visited websites for the dataset collections and other informations include:

https://www.kaagle.com

Journals include:

https://www.ijrte.org

A. Source Code:

Code used to built the flask app

```
import pandas as pd
```

```
import numpy as np
from flask import Flask, render_template, Response, request
import pickle
from sklearn.preprocessing import LabelEncoder
import pickle

app = Flask(__name__, template_folder='templates')
filename = 'resale.pkl'
model_rand = pickle.load(open(filename, 'rb'))


@app.route('/')
def index():
    return render_template('index1.html')


@app.route('/intro', methods=['GET', 'POST'])
def intro():
    return render_template('intro.html')


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


@app.route('/y_predict', methods=['GET', 'POST'])
def y_predict():
    regyear = int(request.form['regyear'])
    powerps = float(request.form['powerps'])
    kms = float(request.form['kms'])
    regmonth = int(request.form.get('regmonth'))
    gearbox = request.form['gearbox']
    damage = request.form['dam']
    model = request.form.get('model_type')
    brand = request.form.get('brand')
    fuelType = request.form.get('fuel')
```

```
    vehicletype = request.form.get('vehicletype')
    new_row = {'yearOfRegistration': regyear, 'powerPS': powerps,
'kilometer': kms, 'monthOfRegistration': regmonth,
               'gearbox': gearbox, 'notRepairedDamage': damage, 'model':
model, 'brand': brand, 'fuelType': fuelType,
               'vehicleType': vehicletype}

    print(new_row)
    new_df = pd.DataFrame(
        columns=['vehicleType', 'yearOfRegistration', '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)
        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)
    return render_template('predict.html',
ypred="{:.2f}".format(y_prediction[0]))


if __name__ == '__main__':
    app.run(host='Localhost', debug=True, threaded=False)
```