

PROJECT DEVELOPMENT PHASE

DELIVERY OF SPRINT-4

DATE	10 November 2022
TEAM ID	PNT2022TMID43633
PROJECT NAME	Car Resale Value Prediction
MAXIMUM MARK	4 Marks

- **Output of the Python Flask Code**

```
main-app x
"C:\Users\arun\PycharmProjects\Car_Resale_Value_Model IBM\venv\Scripts\python.exe" "C:\Users\arun\PycharmProjects\Car_Resale_Value_Model IBM\main-app.py"
C:\Users\arun\PycharmProjects\Car_Resale_Value_Model IBM\venv\lib\site-packages\sklearn\base.py:329: UserWarning: Trying to unpickle estimator DecisionTreeRe
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
warnings.warn(
C:\Users\arun\PycharmProjects\Car_Resale_Value_Model IBM\venv\lib\site-packages\sklearn\base.py:329: UserWarning: Trying to unpickle estimator RandomForestRe
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
warnings.warn(
* Serving Flask app 'main-app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
C:\Users\arun\PycharmProjects\Car_Resale_Value_Model IBM\venv\lib\site-packages\sklearn\base.py:329: UserWarning: Trying to unpickle estimator DecisionTreeRe
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
warnings.warn(
C:\Users\arun\PycharmProjects\Car_Resale_Value_Model IBM\venv\lib\site-packages\sklearn\base.py:329: UserWarning: Trying to unpickle estimator RandomForestRe
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
warnings.warn(
* Debugger is active!
* Debugger PIN: 909-165-664
```

- **Application User Interface**

Car Resale Value Prediction

Please fill the following details of your car !!!

Registration Year

Enter Year of Registration

Registration Month

Enter Month of Registration

Power of the car in PS

Enter Power of the car in PS

Kilometers Driven

Enter Kilometers Driven

Your car is Damaged or Repaired

Yes

Brand of the car

Audi

Vehicle Type

Bus

Gear Box Type

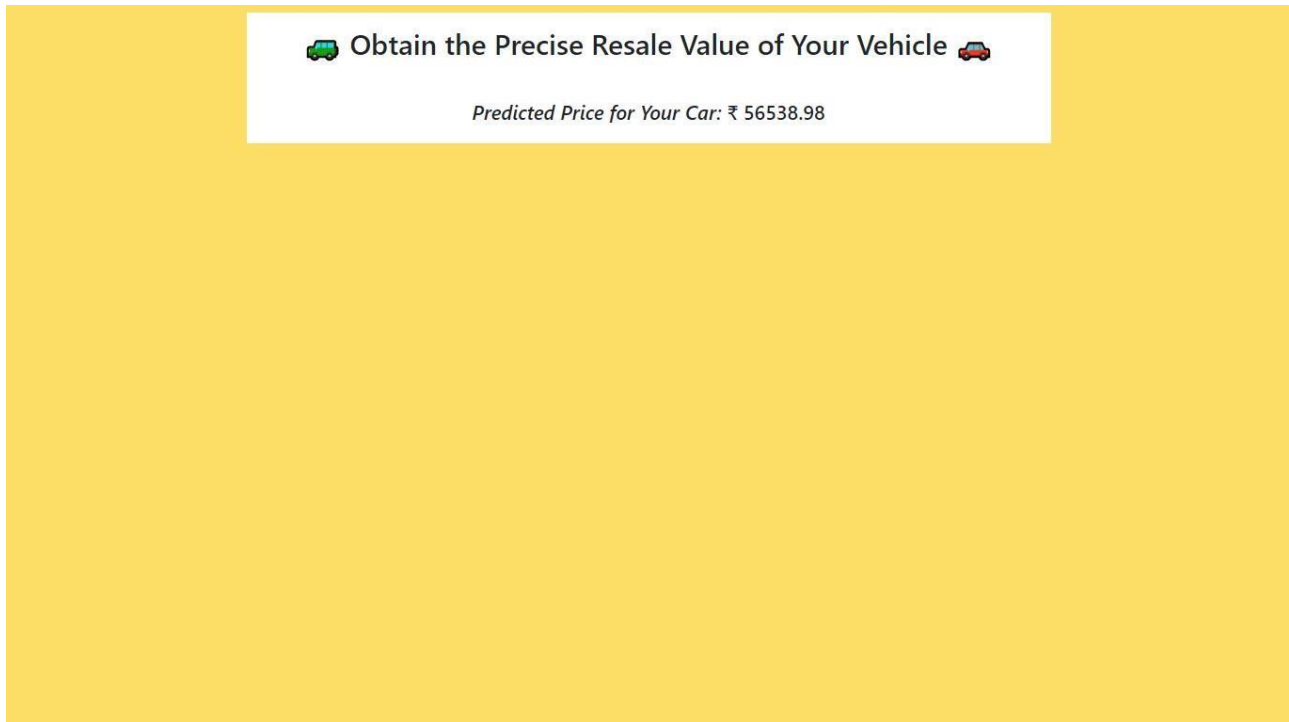
Manual

Fuel Type

CNG

Predict

- **Output Page**



- **Training on IBM CLOUD**

```
In [27]: import pandas as pd
import numpy as np
import matplotlib as plt
from sklearn.preprocessing import LabelEncoder
import pickle

In [28]: import os, types
import pandas as pd
from boto3.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
                              ibm_api_key_id='2Xcapalh0rE5A6fyhLSmRKvLyegsLUgW_Xun2H4mPzHg',
                              ibm_auth_endpoint='https://iam.cloud.ibm.com/oidc/token',
                              config=Config(signature_version='oauth'),
                              endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'carresalevaluepredictionibm-donotdelete-pr-cdchbq0a85f2oq'
object_key = 'autos_preprocessed.csv'

body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )

df = pd.read_csv(body)
df.head()
```

Out[28]:

Unnamed: 0	price	vehicleType	yearOfRegistration	gearbox	powerPS	model	kilometer	monthOfRegistration	fuelType	brand	notRepairedDamage
0	1	18300	coupe	2011	manual	190 not-declared	125000	5	diesel	audi	Yes

```
np.save(str('classes'+1+'.npy'),mapper[1].classes_)
print(i,":",mapper[i])
new_df.loc[:,i+'_labels'] = pd.Series(tr,index = new_df.index)
labeled = new_df[ ['price','yearOfRegistration','powerPS','kilometer','monthOfRegistration'] + [x+"_labels" for x in labels]]

gearbox ; LabelEncoder()
notRepairedDamage ; LabelEncoder()
brand ; LabelEncoder()
fuelType ; LabelEncoder()
vehicleType ; LabelEncoder()
```

In [35]: print(labeled.columns)

```
Index(['price', 'yearOfRegistration', 'powerPS', 'kilometer',
       'monthOfRegistration', 'gearbox_labels', 'notRepairedDamage_labels',
       'brand_labels', 'fuelType_labels', 'vehicleType_labels'],
      dtype='object')
```

In [36]: Y = labeled.iloc[:,0].values
X = labeled.iloc[:,1:].values

In [37]: Y = Y.reshape(-1,1)
from sklearn.model_selection import cross_val_score , train_test_split
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.3,random_state=3)

In [38]: from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score
regressor = RandomForestRegressor(n_estimators = 1000,max_depth=10,random_state=34)

In [39]: regressor.fit(X_train,np.ravel(Y_train,order='C'))

Out[39]: RandomForestRegressor(max_depth=10, n_estimators=1000, random_state=34)

In [40]: y_pred = regressor.predict(X_test)
print(r2_score(Y_test,y_pred))

0.8282067869983389