CAR RESALE VALUE PREDICTION

TEAM ID: PNT2022TMID02177

KISHORE M M - 2116190701101

JAGADEEP N - 2116190701070

KANNAN M - 211619071086

KISHAN KUMAR E - 2116190701099

Project Report Format

1. **INTRODUCTION**
   - 1.1 Abstract
2. **LITERATURE SURVEY**
   - 2.1 Existing problem
   - 2.2 References
3. **IDEATION & PROPOSED SOLUTION**
   - 3.1 Empathy Map Canvas
   - 3.2 Ideation & Brainstorming
   - 3.3 Proposed Solution
   - 3.4 Problem Solution fit
4. **REQUIREMENT ANALYSIS**
   - 4.1 Functional requirement
   - 4.2 Non-Functional requirements
5. **PROJECT DESIGN**
   - 5.1 Data Flow Diagrams
   - 5.2 Solution & Technical Architecture
   - 5.3 User Stories
6. **PROJECT PLANNING & SCHEDULING**
   - 6.1 Sprint Planning & Estimation
   - 6.2 Sprint Delivery Schedule
   - 6.3 Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
   - 7.1 Feature
   - 7.2 Database Schema (if Applicable)
8. **TESTING**
   - 8.1 Test Cases
   - 8.2 User Acceptance Testing
9. **RESULTS**
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**

    Source Code

    GitHub & Project Demo Link

1. INTRODUCTION

ABSTRACT

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

A car price prediction has been a high interest research area, as it requires noticeable effort and knowledge of the field expert. Considerable number of distinct attributes are examined for the reliable and accurate prediction

2. LITERATURE SURVEY
2.1. EXISTING PROBLEM

Several studies and related works have been done previously to predict used car prices around the world using different methodologies and approaches, with varying results of accuracy from 50% to 90%. In (Pudaruth, 2014) the researcher proposed to predict used car prices in Mauritius, where he applied different machine learning techniques to achieve his results like decision tree, K-nearest neighbours, Multiple Regression and Naïve Bayes algorithms to predict the used cars prices, based on historical data gathered from the newspaper.

Achieved results ranged from accuracy of 60-70 percent, the author suggested using more sophisticated models and algorithms to make the evaluation, with the main weakness off the decision tree and naïve Bayes that it is required to discretize the price and classify it which accrue to more inaccuracies. Moreover, he suggested a larger set of data of data to train the models hence the data gathered was not sufficient.

(Monburinon, et al., 2018) Gathered data from a German e-commerce site that totalled to 304,133 rows and 11 attributes to predict the prices of used car using different techniques and measured their results using Mean Absolute Error (MEA) to compare their results. Same training dataset and testing dataset was given to each model. Highest results achieved was by using gradient boosted regression tree with a MAE of 0.28, and MEA of 0.35 and 0.55 for mean absolute error and multiple linear regression respectively. Authors suggested adjusting the parameters in future works to yield better results, as well as using one hot encoding instead of label encoding for more realistic data interpretations on categorical data.

(Gegic, Isakovic, Keco, Masetic, & Kevric, 2019) from the International Burch University in Sarajevo, used three different machine learning techniques to predict used car prices. Using data scrapped from a local Bosnian website for used cars totalled at 797 car samples after pre-processing, and proposed using these methods: Support Vector Machine, Random Forest and Artificial Neural network. Results have shown using only one machine learning algorithm achieved results less

than 50%, whereas after combing the algorithms with pre calcification of prices using Random Forest, results with accuracies up to 87.38% was recorded.

(Noor & Jan, 2017) were able to achieve high level of accuracy using Multiple linear regression models to predict the price of cars collected from used cars website in Pakistan called Pak Wheels that totalled to 1699 records after pre-processing, and where able to achieve accuracy of 98%, this was done after reducing the total amount of attributes using variable selection technique to include significant attributes only and to reduce the complexity of the model.

(K.Samruddhi & Kumar, 2020) Proposed using Supervised machine leaning model using K-Nearest Neighbour to predict used car prices from a data set obtained from Kaggle containing 14 different attributes, using this method accuracy reached up to 85% after different values of K as well as Changing the percent of training data to testing data, expectedly when increasing the percent of data that is tested better accuracy results are achieved. The model was also cross validated with 5 and 10 folds by using K fold method.

(Gongqi, Yansong, & Qiang, 2011) proposed using Artificial Neural Network (ANN) through a combined method of BP neural network and nonlinear curve fit and have achieved accurate value prediction with a feasible model.

(Listiani, 2009) used Support Vector Machines to evaluate leased cars prices, results have shown that SVM is far more accurate in large dataset with high dimensional data than Multiple linear regression. Whereas the computation Multiple linear regression can take several minutes and the SVM would take up to a day to compute the results. Multiple linear regression may be simple, but SVM is far more accurate. Moreover, the study includes Samples with up to 178 attributes which is far more than the proposed variable in our study, hence the use of multiple linear regression may be more suitable in our case.

(Kuiper, 2008) Collected data from General Motor of cars that are produced in 2005, where he as well used variable selection technique to include the most relevant attributes in his model to reduce the

complexity of the data. He proposed used Multivariate regression model that would be more suitable for values with numeric format.

In order to predict the price of used cars, researchers (Nabarun Pal, 2018) used a supervised learning method known as Random Forest. Kaggle's dataset was used as a basis for predicting used car prices. In order to determine the price impact of each feature, careful exploratory data analysis was performed. 500 Decision Trees were trained with Random Forests. It is most commonly used for classification, but they turned it into a regression model by transforming the problem into an equivalent regression problem. Using experimental results, it was found that training accuracy was 95.82%, and testing accuracy was 83.63%. By selecting the most correlated features, the model can accurately predict the car price.

In light of the number of works that have been done in this field, another group of researchers (Jian Da Wu, 2017) conducted research on this topic and tried to develop a system that consists of three components: a data acquisition system, a price forecasting algorithm, and a performance analysis. Due to its adaptive learning capability, a conventional artificial neural network (ANN) with a back-propagation network is compared to the proposed ANFIS. In the ANFIS, qualitative fuzzy logic approximation as well as adaptive neural network capabilities are included. Using ANFIS as an expert system in predicting used car prices showed better results in the experiment. Using GUI, the consumer can get accurate and convenient

information about used cars' purchasing prices, and experiments proved that the proposed system could provide accurate and convenient price forecasting.

Hence, from all literature review it is concluded that used cars price prediction is an important topic which is the area of many researchers nowadays. So far, the best achieved accuracy is 83.63% on kaggle’s dataset using random forest technique. The researchers have tested multiple regressors and final model is regression model using linear regression.

Method :

The topic such as this can be assessed with mathematical models derived from quantitative data. A multiple variable regression can analyze the data by assessing the role each independent variable plays in determining the dependent variable (in this case, resale value). Significance can also be assessed by observing the p-values for each variable. The use of a statistical model will aid in making a claim on this, and to identify some of the major contributors to resale value in automobiles.

Data Collection :

The data used for this regression will be quantitative in nature. The sources of data are what someone would expect for used car information. Four sources that are used include Kelly Blue Book, Edmunds, a government fuel economy resource, and Car and Driver. Kelly Blue Book and Edmunds will both serve as data sources, with each source providing different aspects of the independent variables used. With the cooperation of these sources, data regarding price of a car-including new and used-with the respective age, mileage, make, condition, miles per gallon, safety ratings, and hybrid technology information will be obtained. These variables will allow for a regression to be run and an equation to be estimated.

Expected Outcomes :

Before I can make predictions regarding the influence each variable will have on resale value, a review of prior research and literature is appropriate. This will allow me to make a more confident prediction as well as confirm which variables are needed to produce a strong equation that explains much of the variations in vehicle depreciation. An expected equation could look like this:

Resale Value (DV) = Intercept- B3(Age) - B4(Mileage) + Bl(Make) + B2(MPG) + B5(Hybrid Tech)

2.2. REFERENCES

[1] Pudaruth, Sameerchand. "Predicting the price of used cars using machine learning techniques." Int. J. Inf. Comput. Technol 4, no. 7 (2014): 753-764.

[2] Monburinon, Nitis, Prajak Chertchom, Thongchai Kaewkiriya,Suwat Rungpheung, Sabir Buya, and Pitchayakit Boonpou. "Prediction of prices for used car by using regression models." In2018 5th International Conference on Business and Industrial Research (ICBIR), pp. 115-119. IEEE, 2018.

[3] Gegic, Enis, Becir Isakovic, Dino Keco, Zerina Masetic, and JasminKevric. "Car price prediction using machine learning techniques." TEM Journal 8, no. 1 (2019): 113.

[4] Noor, Kanwal, and Sadaqat Jan. "Vehicle price prediction system using machine learning techniques." International Journal of Computer Applications 167, no. 9 (2017): 27-31.

[5] Kanwal Noor, 2017, Vehicle Price Prediction System using Machine Learning Techniques International Journal of Computer Applications. Volume 167 - Number 9

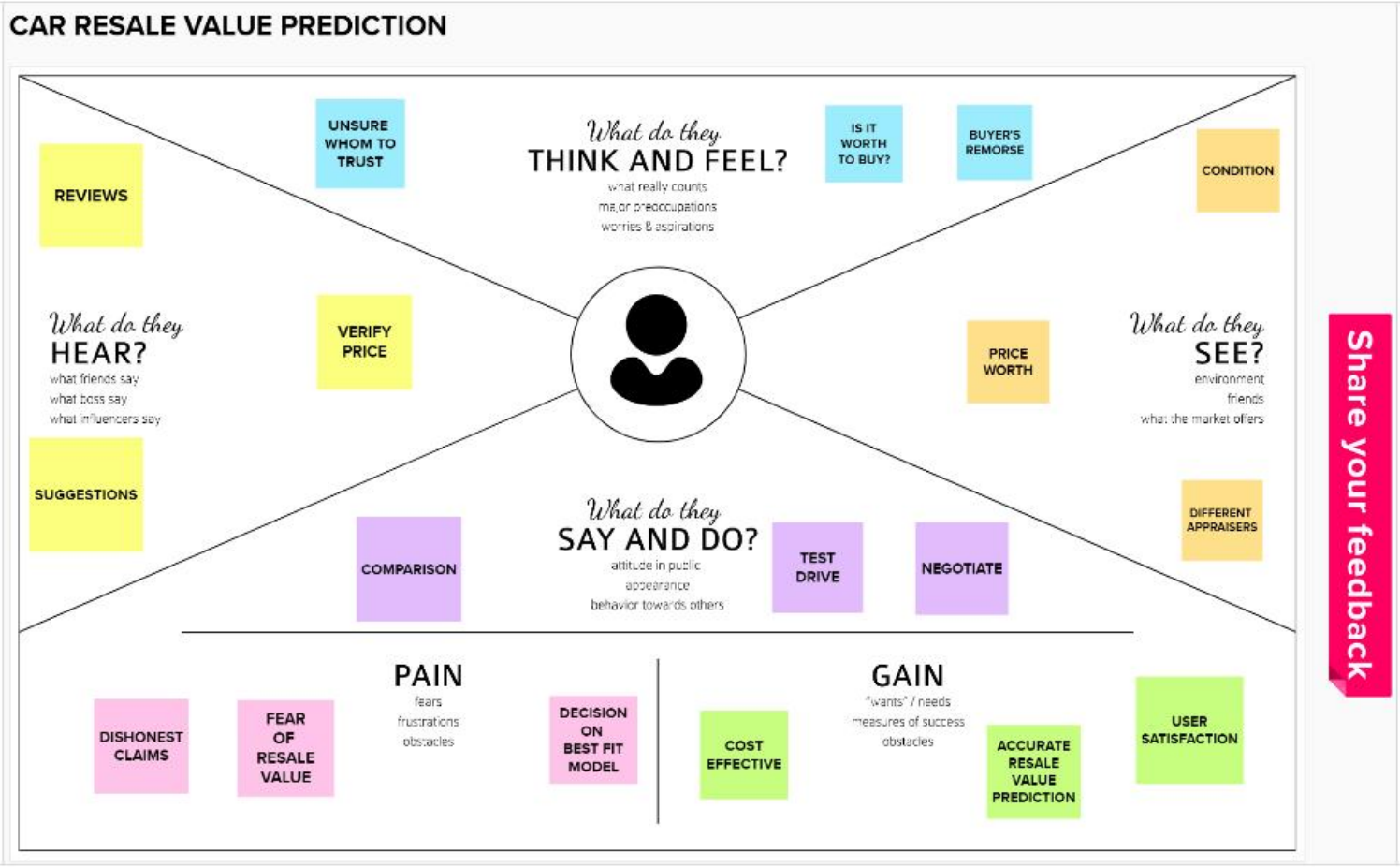
[6] Mariana Lusitania et al, (2009). Support vector regression analysis for price prediction in a vehicle leasing application

[7] Richardson, M. S. (2009). Determinants of used vehicle resale value.

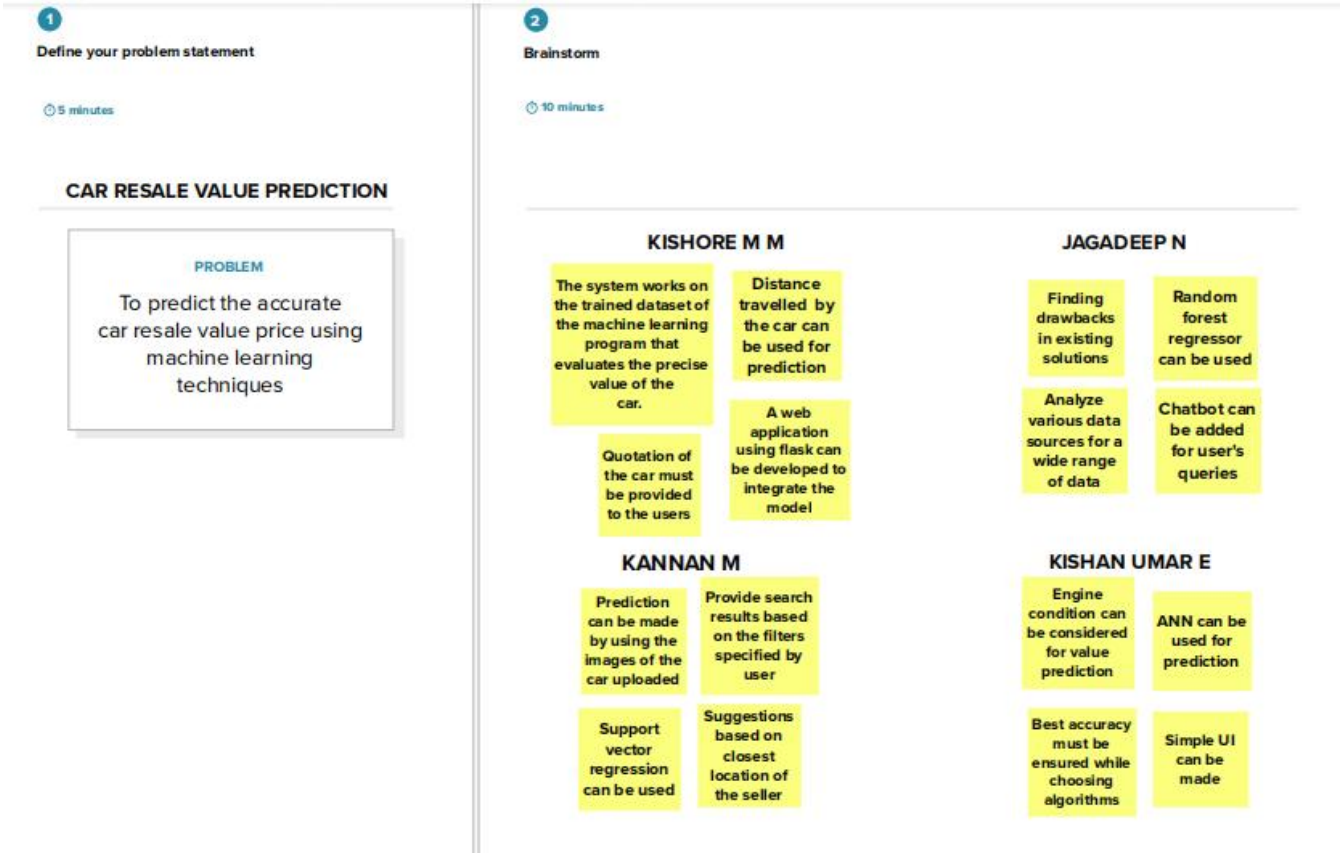
[8] Listiani, M. (2009). Support vector regression analysis for price prediction in a car leasing application (Doctoral dissertation, Master thesis, TU Hamburg-Harburg).

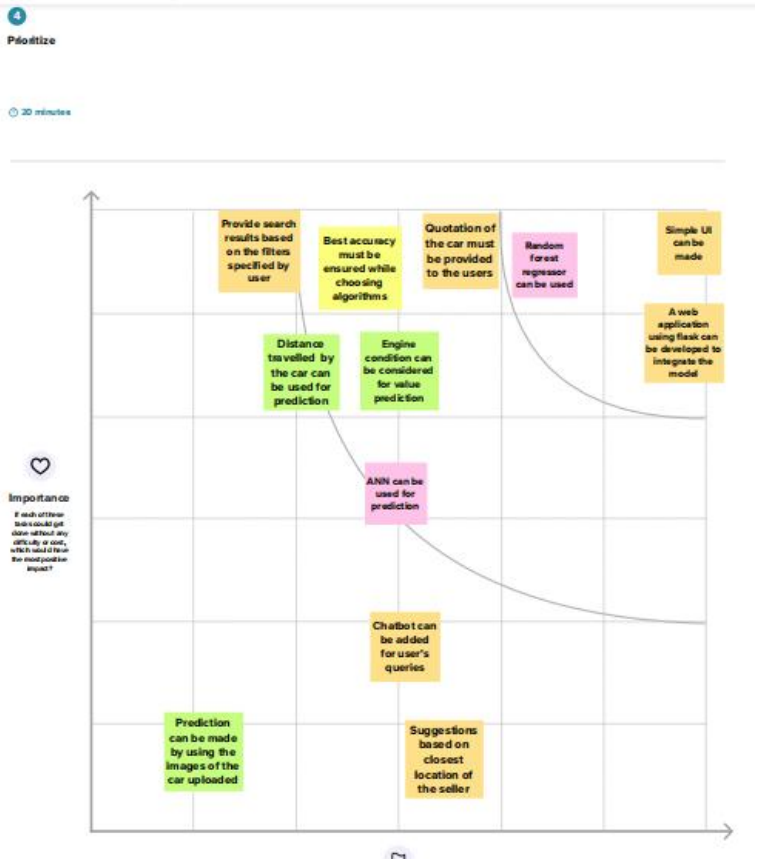
3. IDEATION PHASE

3.1. EMPATHY MAP



3.2. BRAINSTROMING





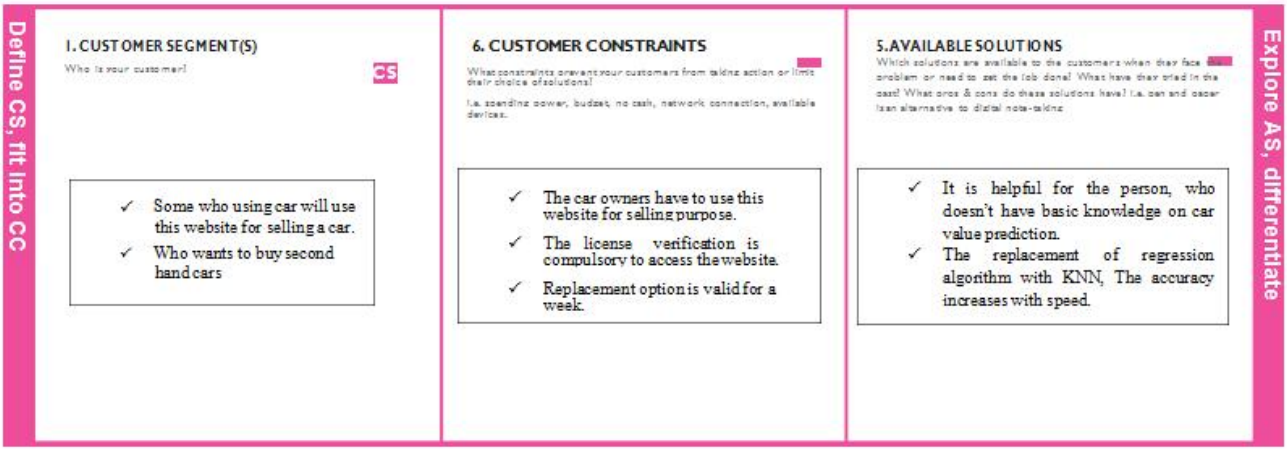
3.3. PROPOSED SOLUTION

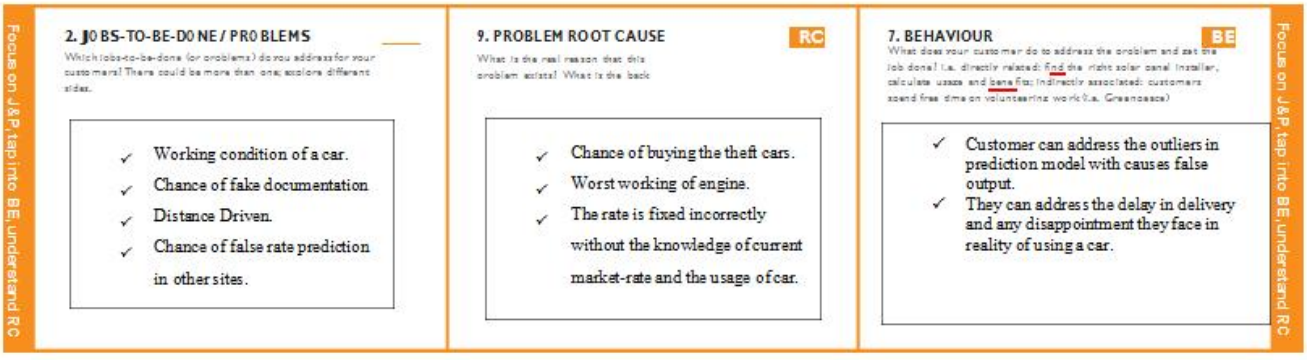
Proposed Solution Template:

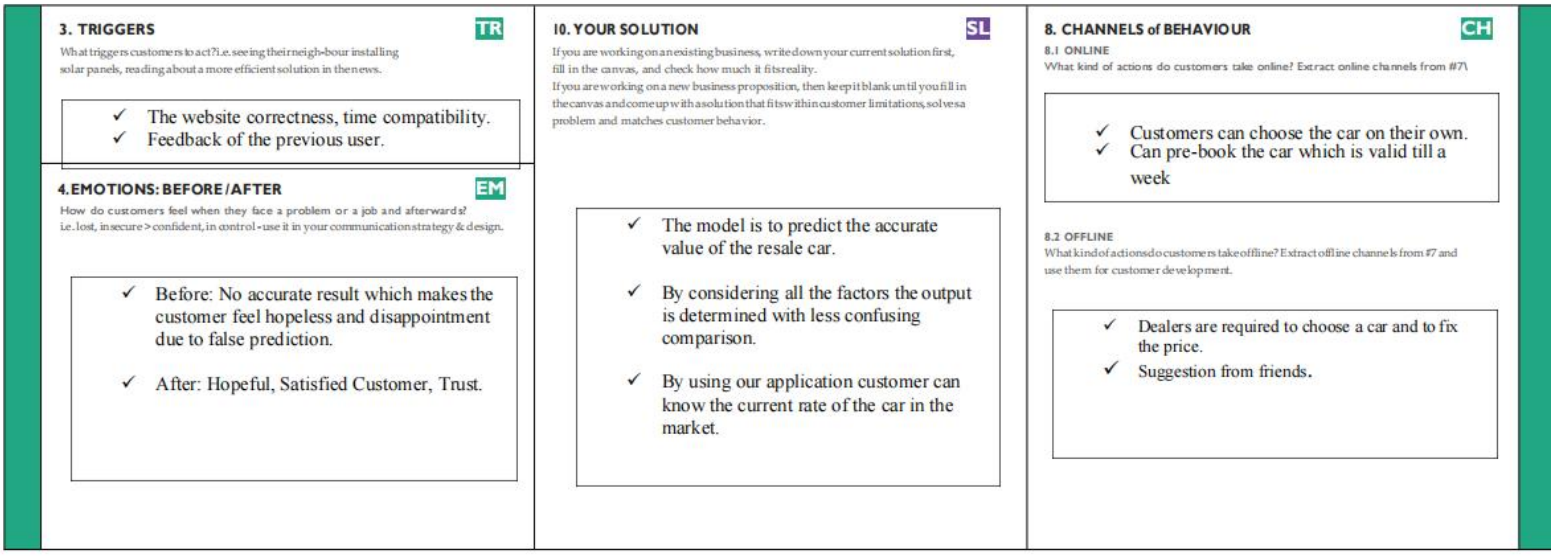
Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The Statement is to built a system to predict the resale value of the car using the regression algorithm
2.	Idea / Solution description	To forecast the car's resale value by taking into account the key elements that influence a car's resale value. Building a regression model that would give the closest estimate of the car's resale value.Using a variety of regression algorithms,The most accurate algorithm will be if adopted as a remedy, it will then be integrated into the user's web-based application receives information on the status of his goods
3.	Novelty / Uniqueness	The goal of this thesis is to analyse the behaviour of various different machine learning models for used automobile price prediction. Machine learning's understanding as it relates to automobile valuations and other comparable price prediction issues will grow as a result of this.
4.	Social Impact / Customer Satisfaction	It can be the ideal platform for purchasing or selling used vehicles. The ideal worth of the car is forecasted , preventing individuals from being let down by resale car price predictions.The customer’s requirement is taken into consideration and the ultimate aim would be to bring customer satisfaction by accurate prediction.
5.	Business Model (Revenue Model)	A revenue model is a plan that outlines how a new company will make money from its regular business operations and how it will cover its operational costs and expenses.The price is predicted not only by using the condition and previous status of the car, but also it considers the current market value of the particular model.
6.	Scalability of the Solution	The model is trained by setting vast population. The sample data include all possible model and features of the car. So, it can predict the rate for all type of car in world wide.

3.4. PROBLEM SOLUTION FIT







4. Solution Requirements (Functional& Non-functional)

4.1. Functional Requirements

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Website
FR-2	User Confirmation	Confirmation via website
FR-3	Car Registration	Registering the car details
FR-4	Value Prediction	Predicting the car resale value

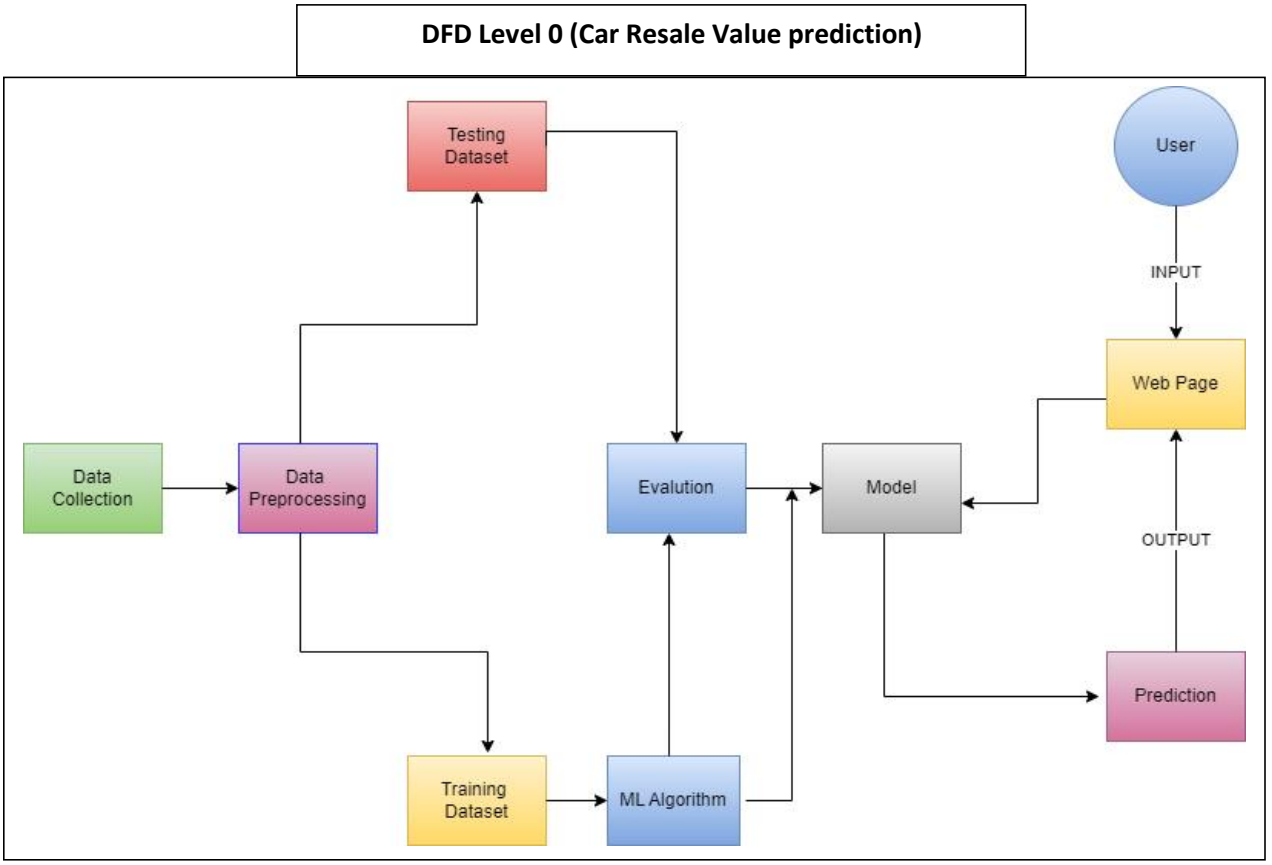
4.2. Non-functional Requirements

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Predicting the resale value
NFR-2	**Security**	Providing security to the website
NFR-3	**Reliability**	Providing high reliability by predicting values for different types of cars
NFR-4	**Performance**	Providing high performance by using some machine learning techniques
NFR-5	**Availability**	It is used for all types of cars
NFR-6	**Scalability**	Predicting values for different types of cars

5. Project Design Phase-I

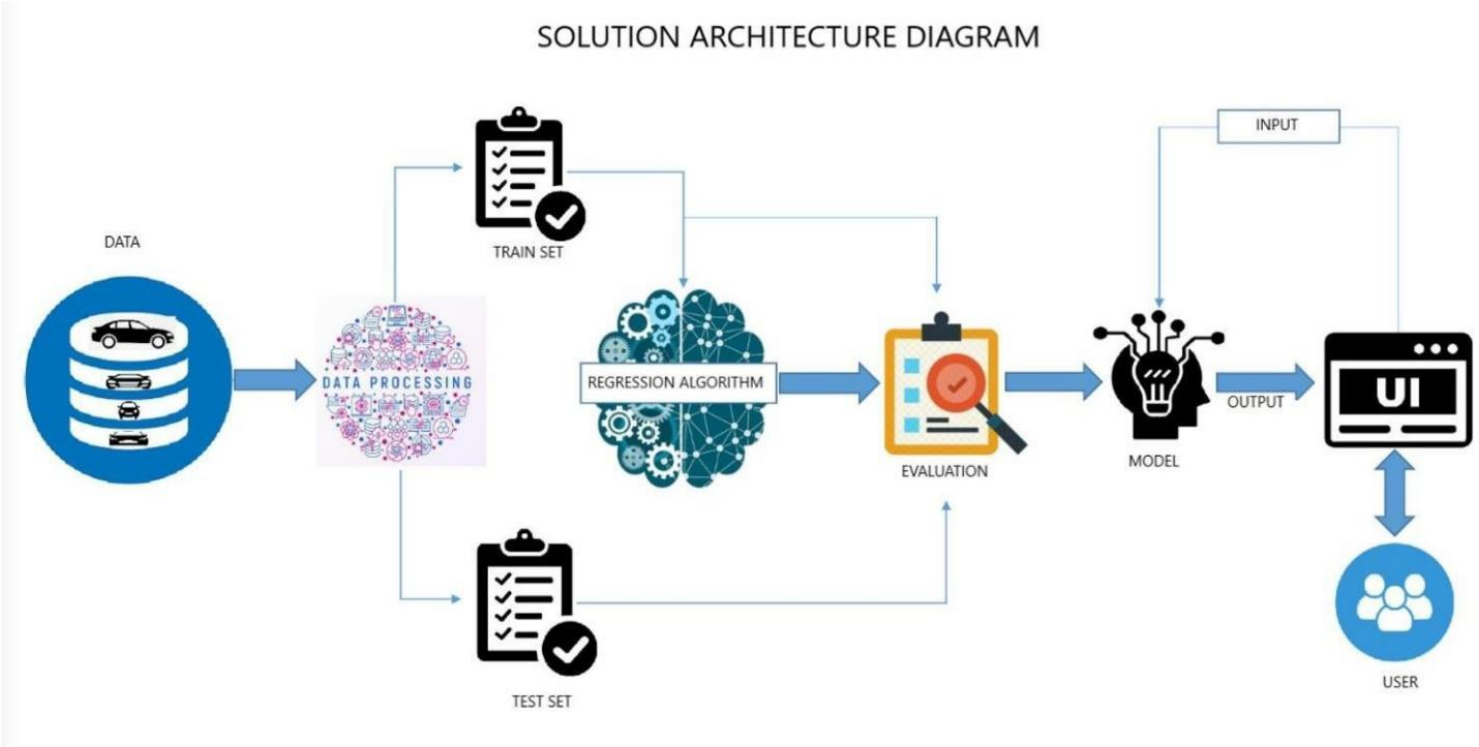
5.1. Data Flow Diagram & User Stories



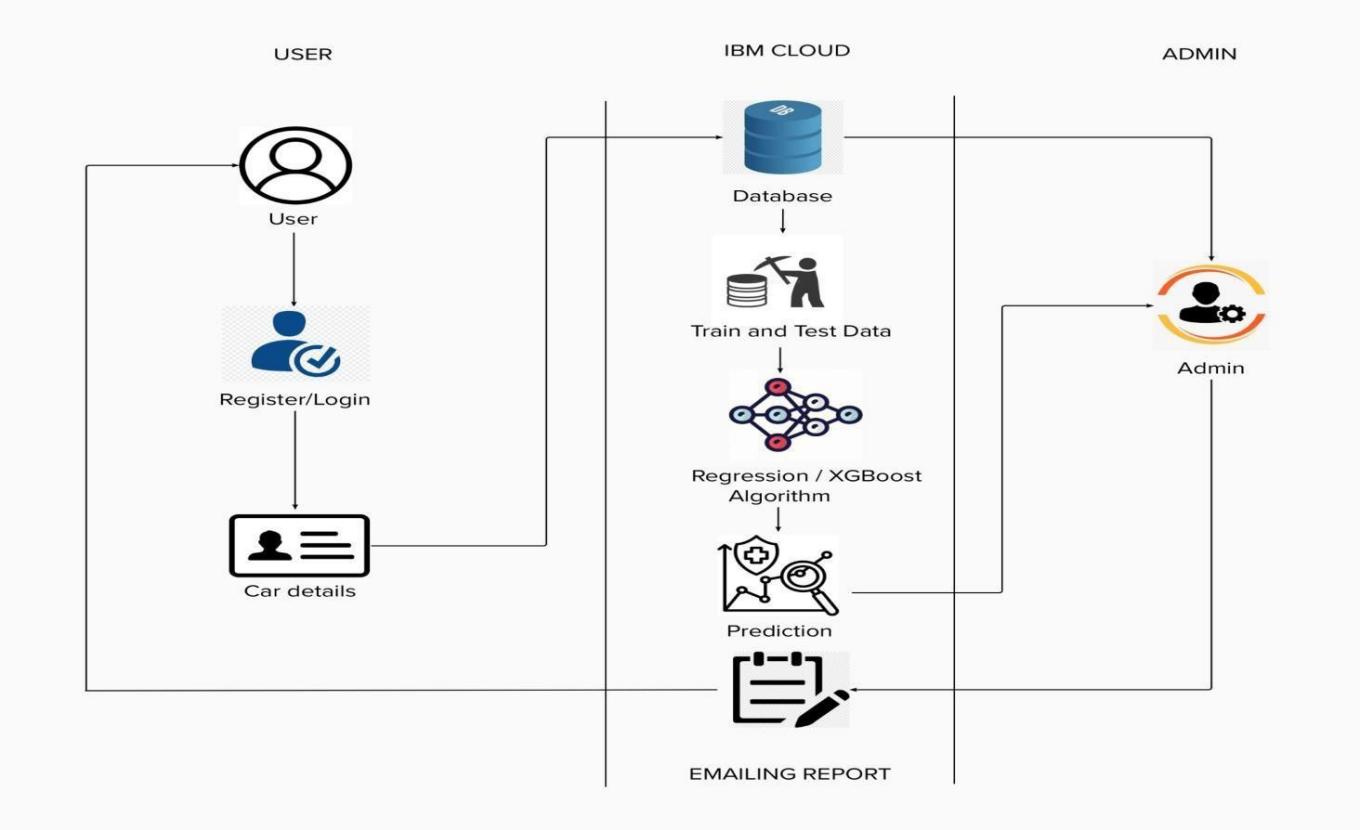
User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the car details application by entering my email, password, and confirming my password.	I can access my dashboard and view the car details	High	Sprint-1
		USN-2	As a user, I will receive car resale value in the application.	I can receive my car resale value in the application	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register & access the dashboard with Gmail Login	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can access my dashboard and view the car details	High	Sprint-1
	Dashboard	USN-1-5	Show the details of different varieties of used cars.	I can know the resale value of a car	High	Sprint-1

5.2. SOLUTION ARCHITECTURE



5.2. Technology Architecture



Technical Architecture:

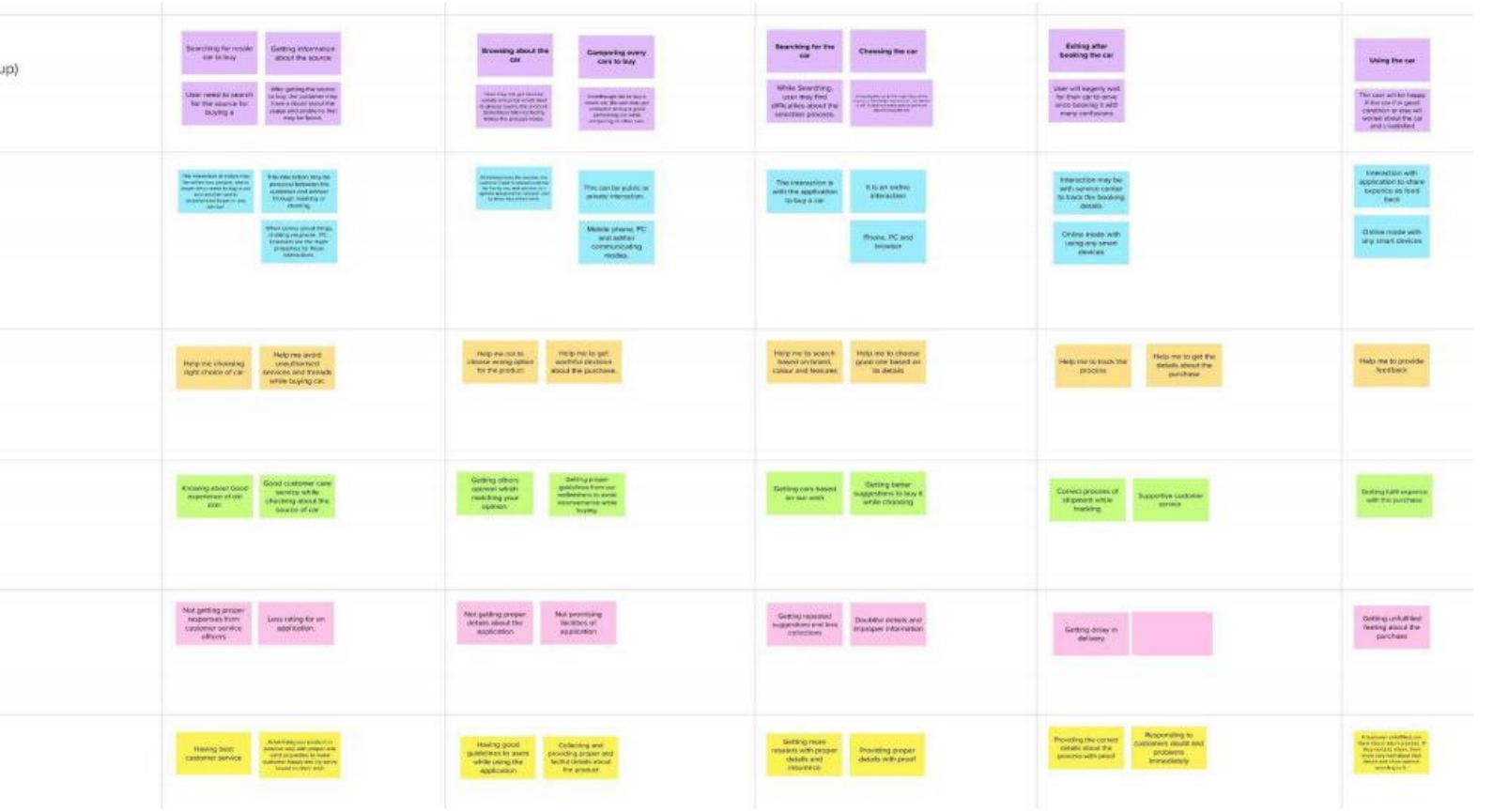
Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Using a Web UI, the user engages with theapplication	Python flask
2.	Dataset	The dataset containing the price and specificationof the cars details is used for training the model	Python libraries like numpy, pandas,etc
3.	Cloud Database	The dataset is stored in the IBM cloud	IBM cloud
4.	Machine Learning Model	The machine learning algorithm are used to predictthe used car rates	Regression model

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Open- source framework is used for building webapplication	Python Flask , Python , IBM Cloud
2.	Security Implementations	Authentication process Implementation	Encryptions
3.	Scalable Architecture	Scalability of architecture consist of 3 tiers like: Web server, Application server, Database server	Web server- HTML,CSS, Java script Application server- Python flask Database server- IBM cloud
4.	Availability	The user can access through web browser	IBM cloud hosting
5.	Performance	Multiple users can access the web application	IBM load balance

5.3. CUSTOMER JOURNEY



6. Project Planning Phase
6.1. SPRINT DELIVERY PHASE

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 splitting to dependent andindependent variables	2	High	KISHAN KUMAR E KANNAN M
Sprint-2	Building the Model	USN-2	Choosing the appropriate model for building andsavingthe models pickle file	1	High	KISHORE M M JAGADEEP N
Sprint-3	Application Building	USN-3	Using flask deploying the ML model	2	Medium	KISHORE M M KANNAN M
Sprint-4	Train the Model in ibm	USN-4	Finally train the model on IBM cloud and deploy theapplication	2	Medium	JAGADEEP N KISHAN KUMAR E

Sprint	Total Story Points	Duration	Sprint StartDate	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	5 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	5 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	5 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	5 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

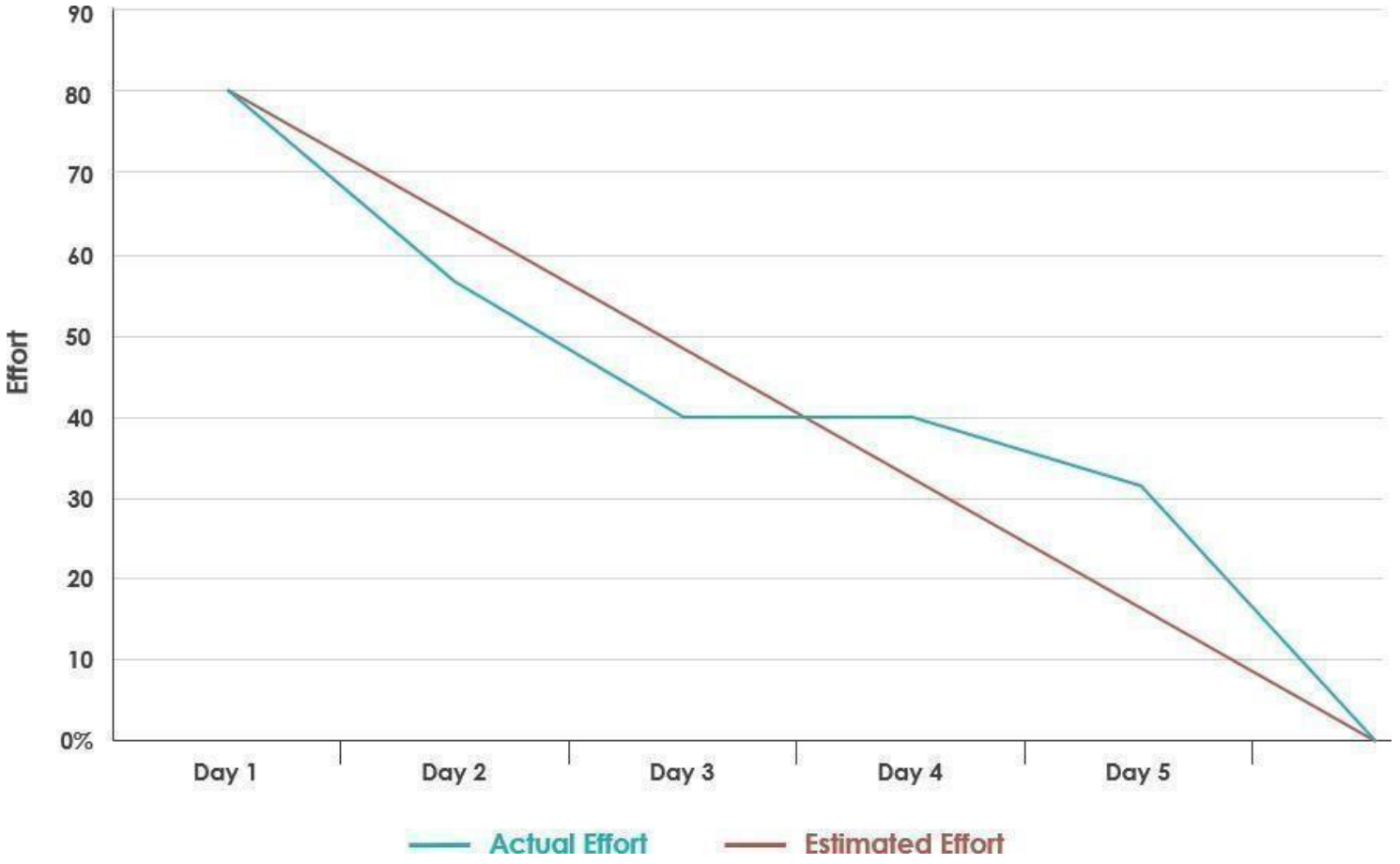
Velocity:

We have a 5-day sprint duration, and the velocity of the team is 15 (points per sprint). The team's average velocity (AV) per iteration unit (storypoints per day)

Actual Velocity = Sprint Duration/Velocity = 15/5 = 3

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies suchas Scrum. However, burn down charts can be applied to any project containing measurable progress over time



6.2. MILESTONE AND ACTIVITY LIST

TITLE	DESCRIPTION	ASSIGNED TO
Literature Survey & Information Gathering	Literature survey on the selected project & gatheringinformation by referring the,technical papers, research publications etc	KISHORE M M
Prepare Empathy Map	Prepare Empathy Map Canvasto capture the user Pains & Gains, Prepare list of problemstatements	JAGADEEP N
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas basedon the feasibility & importance.	KISHORE M M JAGADEEP N KISHAN KUMAR E KANNAN M
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact,scalability of solution, etc.	KISHORE M M KANNAN M KISHAN KUMAR E
Problem Solution Fit	Prepare problem - solution fit document.	KISHORE M M KANNAN M JAGADEEP N
Solution Architecture	Prepare solution architecture document.	JAGADEEP N KISHAN KUMAR E

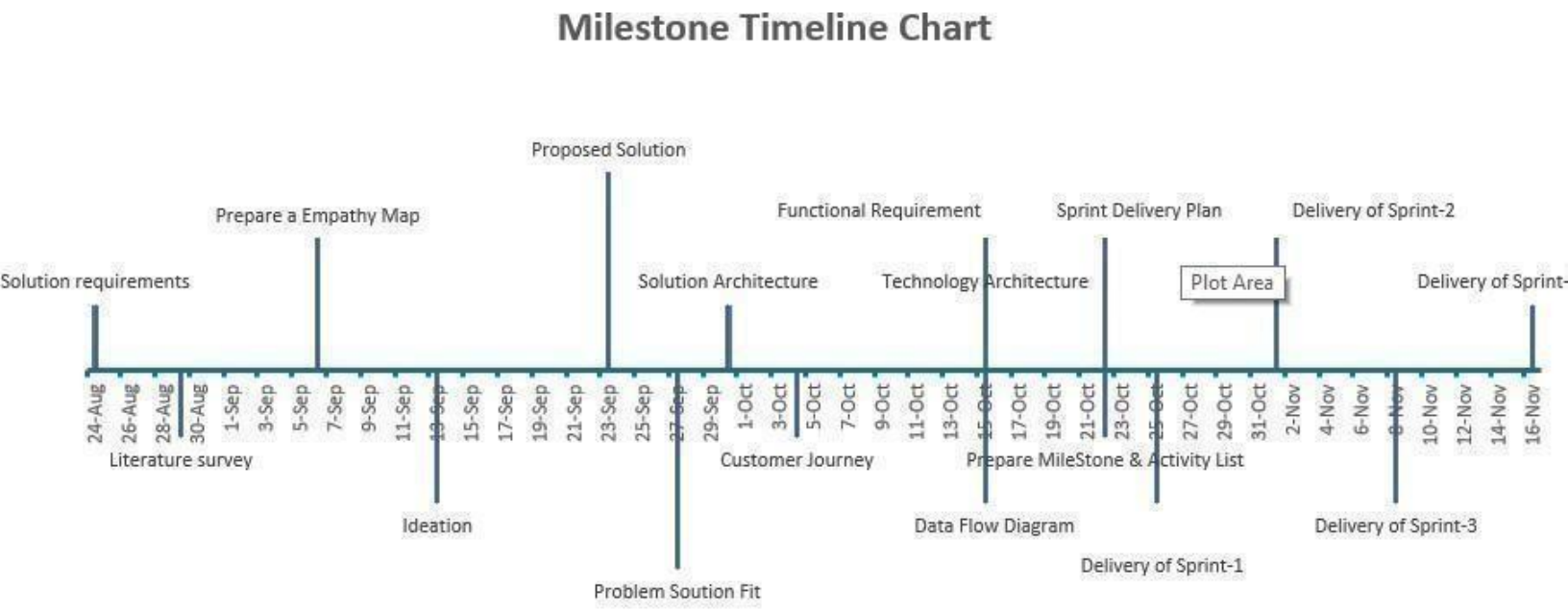
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences withthe application (entry to exit).	KISHORE M M KANNAN M
Functional Requirement	Prepare the functional requirement document	KANNAN M KISHAN KUMAR E
Data Flow Diagrams	Draw the data flow diagramsand submit for review.	KISHORE M M JAGADEEP N
Technology Architecture	Prepare the technology architecture diagram	JAGADEEP N KISHAN KUMAR E
Prepare Milestone & ActivityList	Prepare the milestones & activity list of the project.	KISHORE M M JAGADEEP N
Sprint Delivery Plan	Prepare the Sprint Delivery Plan	KANNAN M KISHAN KUMAR E
Project Development Phase	Coding & Solutioning Sprint-1 Delivery-Develop the code,Test and Push it to GitHub Acceptance Testing Sprint-2 Delivery-Develop the code,Test and Push it to GitHub Sprint-3 Delivery-Develop the code,Test and Push it to GitHub Performance Testing Sprint-4 Delivery-Develop the code,Test and Push it to GitHub	KISHORE M M JAGADEEP N KISHAN KUMAR E KANNAN M

Milestone:

When project begins then it is expected that project related activities must be initiated. In project planning, series of milestones must be established. Milestone can be defined as recognizable endpoint of software project activity. At each milestone, report must be generated.

Milestone is distinct and logical stage of the project. It is used as signal post for project start and end date,need for external review or input and for checking budget, submission of the deliverable, etc. It simply represents clear sequence of events that are incrementally developed or build until project gets successfully completed. It is generally referred to as task with zero-time duration because they are used tosymbolize an achievement or point of time in project. It helps in signifying change or stage in development.

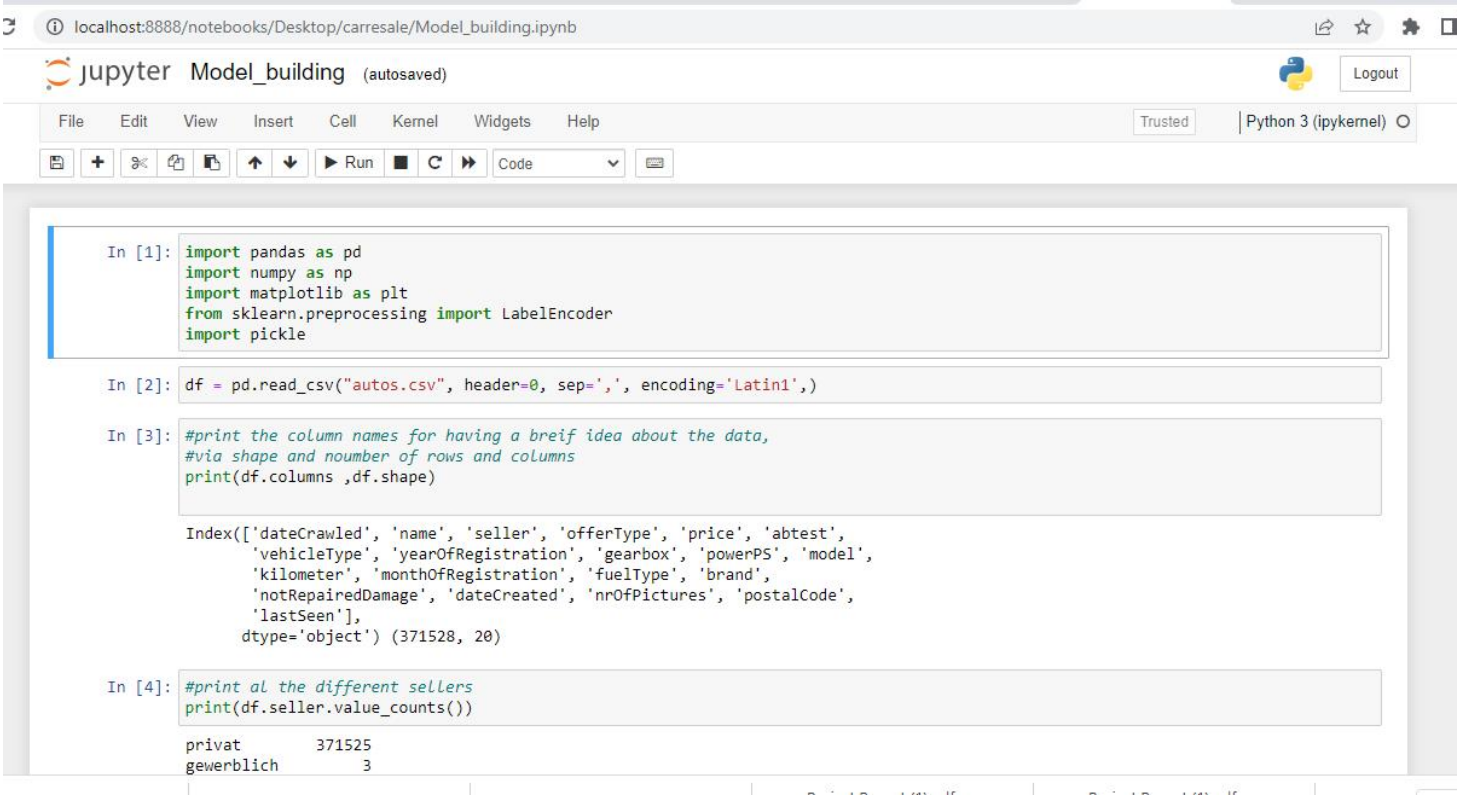
6.3. REPORTS FROM JIRA

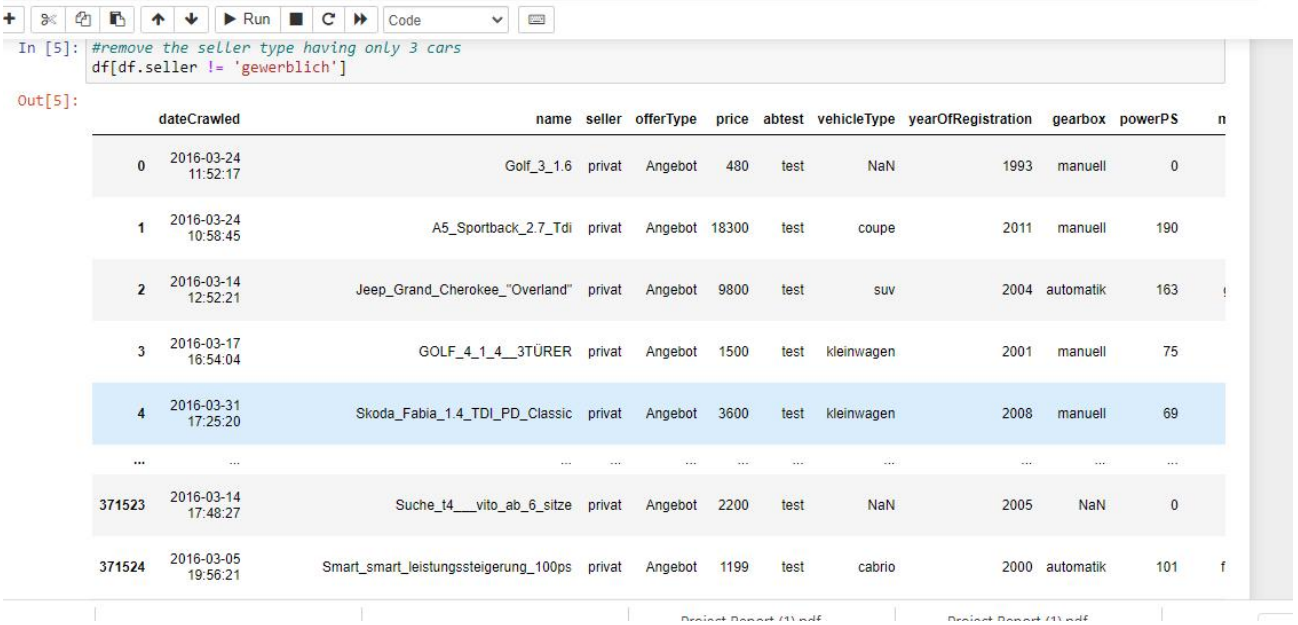


7. CODING & SOLUTIONING

7.1. FEATURE 1

Initally we have planned for Downloading the dataset, import the libraries, Read the dataset, Understanding data types and summary of features, Handling missing values, Replacing the missing values, Label encoding, Split the dataset into dependent and independent variable, split the dataset into train and test set, Model building, Test the model, Evaluate the model and save the model.





	dateCrawled	name	seller	offerType	price	abtest	vehicleType	yearOfRegistration	gearbox	powerPS	n
0	2016-03-24 11:52:17	Golf_3_1.6	privat	Angebot	480	test	NaN	1993	manuell	0	
1	2016-03-24 10:58:45	A5_Sportback_2.7_Tdi	privat	Angebot	18300	test	coupe	2011	manuell	190	
2	2016-03-14 12:52:21	Jeep_Grand_Cherokee_"Overland"	privat	Angebot	9800	test	suv	2004	automatik	163	
3	2016-03-17 16:54:04	GOLF_4_1_4__3TÜRER	privat	Angebot	1500	test	kleinwagen	2001	manuell	75	
4	2016-03-31 17:25:20	Skoda_Fabia_1.4_TDI_PD_Classic	privat	Angebot	3600	test	kleinwagen	2008	manuell	69	
...	...	...	...	...	...	...	...	...	...	...	...
371523	2016-03-14 17:48:27	Suche_t4___vito_ab_6_sitze	privat	Angebot	2200	test	NaN	2005	NaN	0	
371524	2016-03-05 19:56:21	Smart_smart_leistungssteigerung_100ps	privat	Angebot	1199	test	cabrio	2000	automatik	101	f

jupyter Model_building (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

In [7]:
```
#print al the different sellers
print(df.offerType.value_counts())
```

```
Angebot    371516
Gesuch         12
Name: offerType, dtype: int64
```

In [8]:
```
#remove the Offer Type having only 12 listings
df[df.offerType != 'Gesuch']
```

Out[8]:

	dateCrawled	name	offerType	price	abtest	vehicleType	yearOfRegistration	gearbox	powerPS	model	I
0	2016-03-24 11:52:17	Golf_3_1.6	Angebot	480	test	NaN	1993	manuell	0	golf	
1	2016-03-24 10:58:45	A5_Sportback_2.7_Tdi	Angebot	18300	test	coupe	2011	manuell	190	NaN	
2	2016-03-14 12:52:21	Jeep_Grand_Cherokee_"Overland"	Angebot	9800	test	suv	2004	automatik	163	grand	
3	2016-03-17 16:54:04	GOLF_4_1_4__3TÜRER	Angebot	1500	test	kleinwagen	2001	manuell	75	golf	
4	2016-03-31 17:25:20	Skoda_Fabia_1.4_TDI_PD_Classic	Angebot	3600	test	kleinwagen	2008	manuell	69	fabia	

In [10]:
```
#Cars having power less than 50ps and above 900ps seems a little suspicious,
#let's remove them and see what we've got now
print(df.shape)
df = df[(df.powerPS > 50) & (df.powerPS < 900)]
```

```
(371528, 18)
```

In [11]:
```
print(df.shape)
#around 50000 cars ahave been removed which could have inrouduced error to our data
```

```
(319709, 18)
```

In [12]:
```
#simlarly, filtering our the cars having registeration years not in the mentioned range
#print(df.shape)
df = df[(df.yearOfRegistration >= 1950) & (df.yearOfRegistration < 2017)]
```

In [13]:
```
print(df.shape)
# not much of a difference but still, 10000 rows have been reduced. it's better to
#get rid of faulty data instead of keeping them just to increase the size.
```

```
(309171, 18)
```

In [14]:
```
#removing irrelevant columns which are either the same for all the cars in teh dataset, or can
#introduce bias, so removing them too.
df.drop(['name','abtest', 'dateCrawled', 'nrOfPictures', 'lastSeen',
         'postalCode', 'dateCreated'], axis='columns', inplace=True)
```

In [15]:
```
#final data that we have now
print(df.shape)
```

```
(309171, 11)
```

In [16]:
```
#dropping the duplicates from the dataframe and stroing it in a new df.
#here all rows having same value in all the mentioned columns will be deleted and by default,
#only first occurance of anysuch row is kept
new_df = df.copy()
new_df = new_df.drop_duplicates(['price','vehicleType','yearOfRegistration'
                        ,'gearbox','powerPS','model','kilometer','monthOfRegistration','fuelType'
                        ,'notRepairedDamage'])
```

In [17]:
```
#after removing duplicates
print(new_df.shape)
```

```
(285145, 11)
```

In [18]:
```
#As the dataset contained some german words for many features, cahnging them to english
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','kombi','andere'),
                           ('small car','convertible','combination','others'),inplace=True)
new_df.notRepairedDamage.replace(('ja','nein'),('Yes','No'),inplace=True)
```

In [19]:
```
#### Removing the outliers
```

```
In [20]: #Filling NaN values for columns whose data might not be there with the information provider,
         #which might lead to some variance but our model
         #but we will still be able to give some estimate to the user
         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)

In [21]: #can save the csv for future purpose.
         new_df.to_csv("autos(2).csv")

In [22]: #Columns which contain categorical values, which we'll need to convert via label encoding
         labels = ['gearbox', 'notRepairedDamage', 'model', 'brand', 'fuelType', 'vehicleType']

In [23]: #looping over the labels to do the label encoding for all at once and
         #saving the LABEL ENCODING FILES
         mapper = {}
         for i in labels:
             mapper[i] = LabelEncoder()
             mapper[i].fit(new_df[i])
             tr = mapper[i].transform(new_df[i])
             np.save(str('classes'+i+'.npy'), mapper[i].classes_)
             print(i,":",mapper[i])
             new_df.loc[:, i + '_labels'] = pd.Series(tr, index=new_df.index)

         gearbox : LabelEncoder()
```

Train-Test Split

```
In [26]: #Storing price in Y and rest of the data in X
         Y = labeled.iloc[:,0].values
         X = labeled.iloc[:,1:].values

In [27]: #need to reshape the Y values
         Y = Y.reshape(-1,1)

In [28]: 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)
```

Model building and Fitting

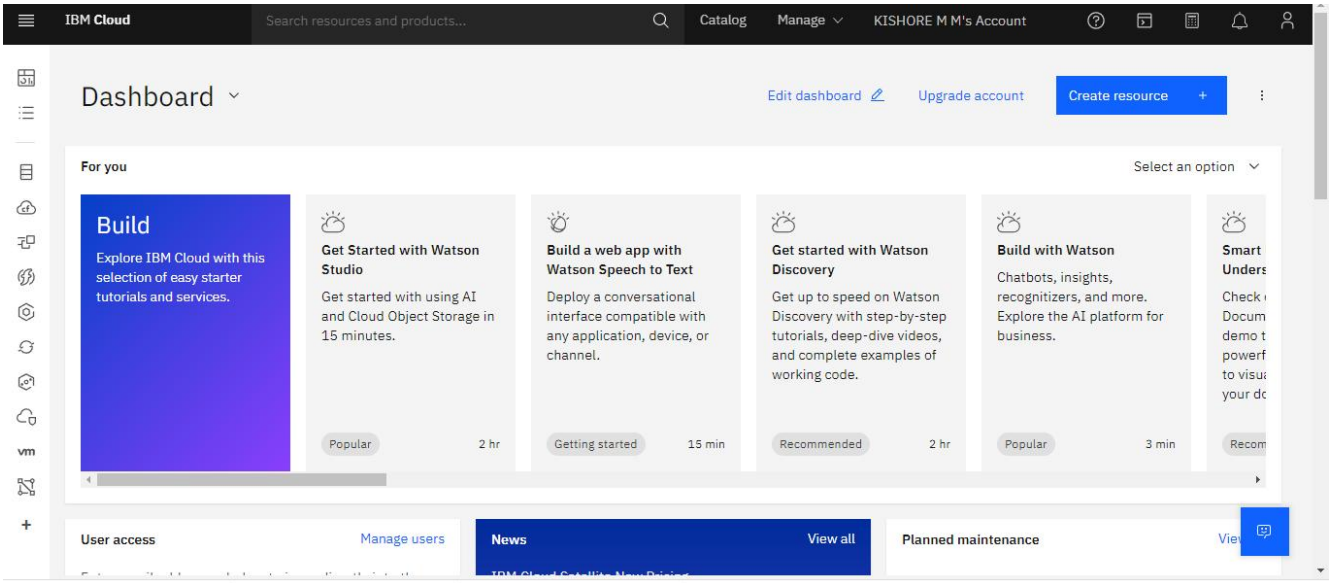
```
In [29]: from sklearn.ensemble import RandomForestRegressor
         from sklearn.metrics import r2_score
         regressor = RandomForestRegressor(n_estimators=1000,max_depth=10,random_state=34)

In [30]: #fitting the model
         regressor.fit(X_train, np.ravel(Y_train,order='C'))

Out[30]: RandomForestRegressor(max_depth=10, n_estimators=1000, random_state=34)
```

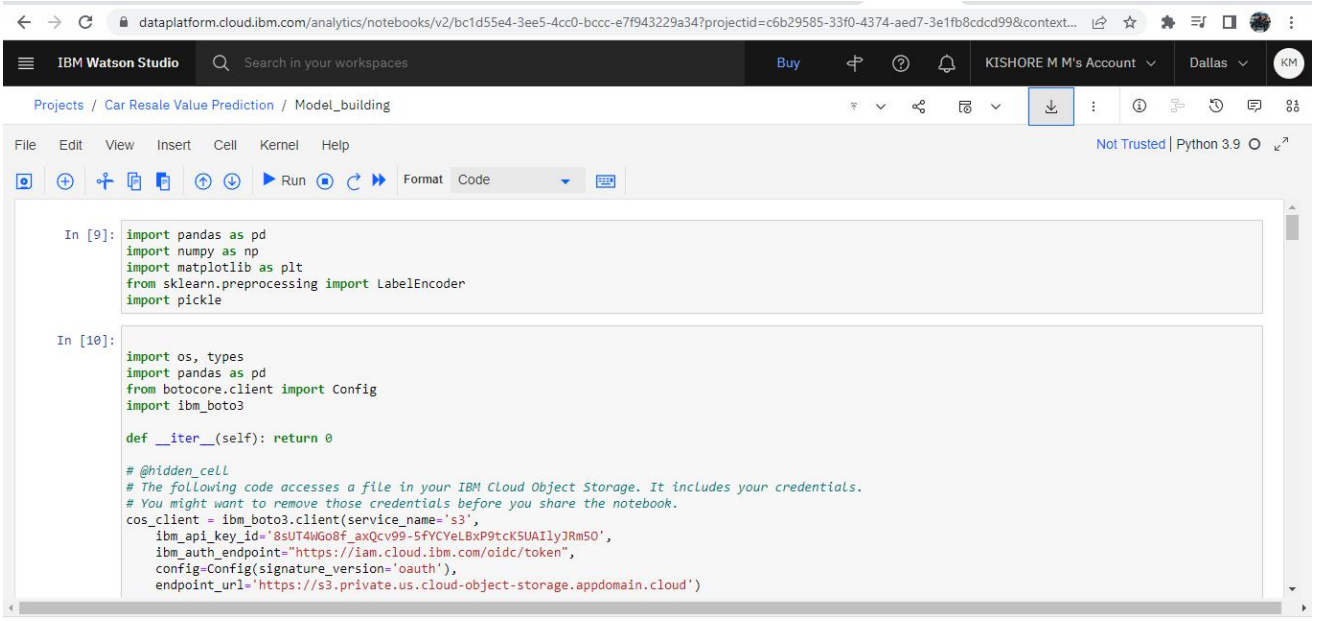
jupyter Model_building (autosaved)

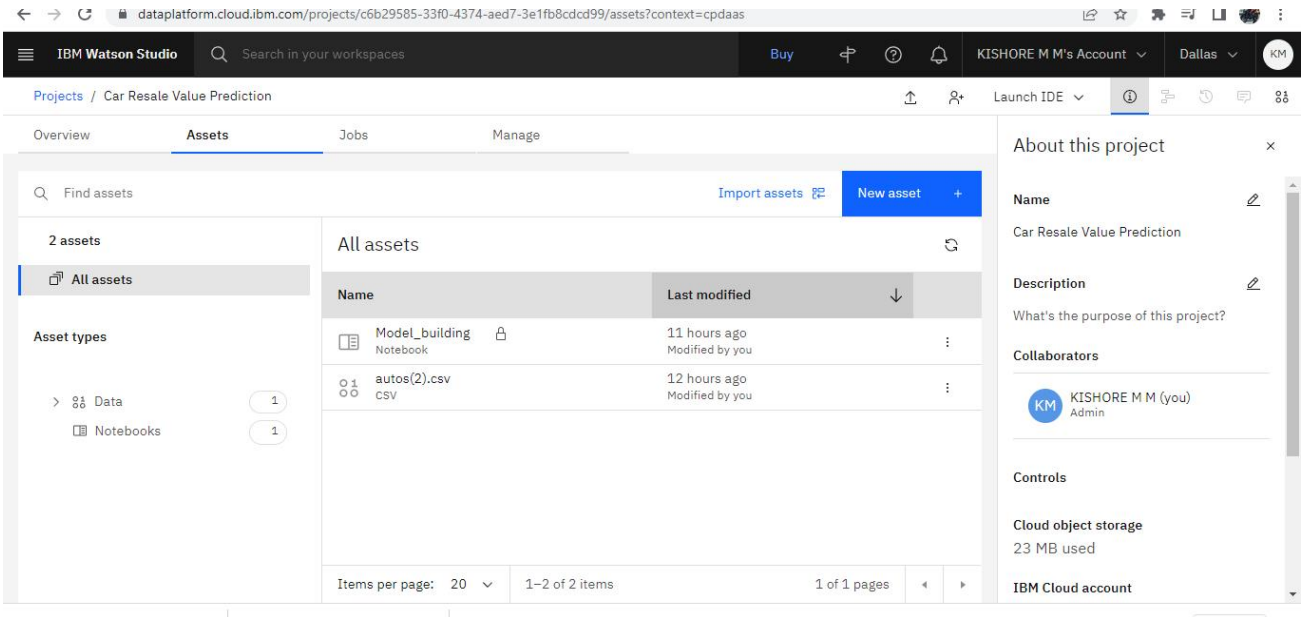
File Edit View Insert Cell Kernel Widgets Help — Trusted — Python 3 (ipykernel)

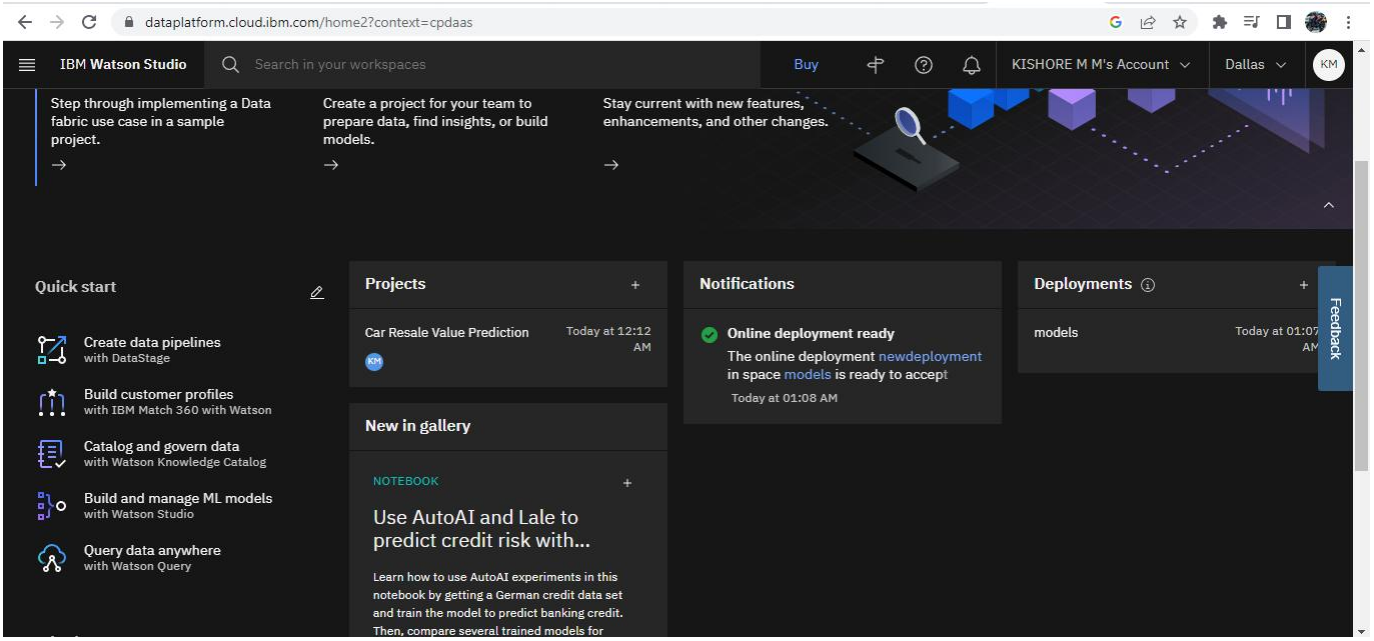
```
In [31]: #predicting the values fo test test
         y_pred = regressor.predict(X_test)

In [32]: #printing the Accuraccy for test set
         print(r2_score(Y_test,y_pred))

         0.834527626497731

In [33]: #for testing on user input values
         y_pred1 = regressor.predict([[2011,190,125000,5,1,0,163,1,3,3]])
         #predticting price for a user input values
         print(y_pred1)

         [19559.28944983]

In [35]: #saving the model for future use.
         filename = 'car_resale_value.sav'
         pickle.dump(regressor, open(filename, 'wb'))

In [ ]:
```

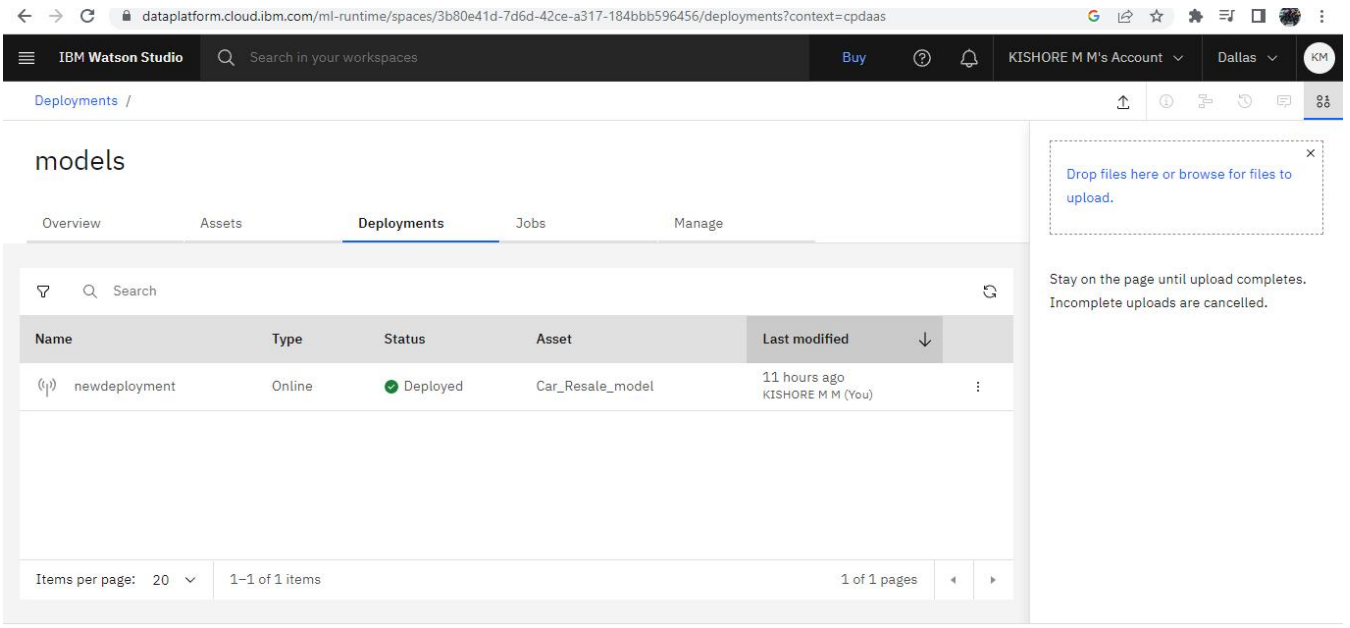
TRAIN MODEL ON IBM:

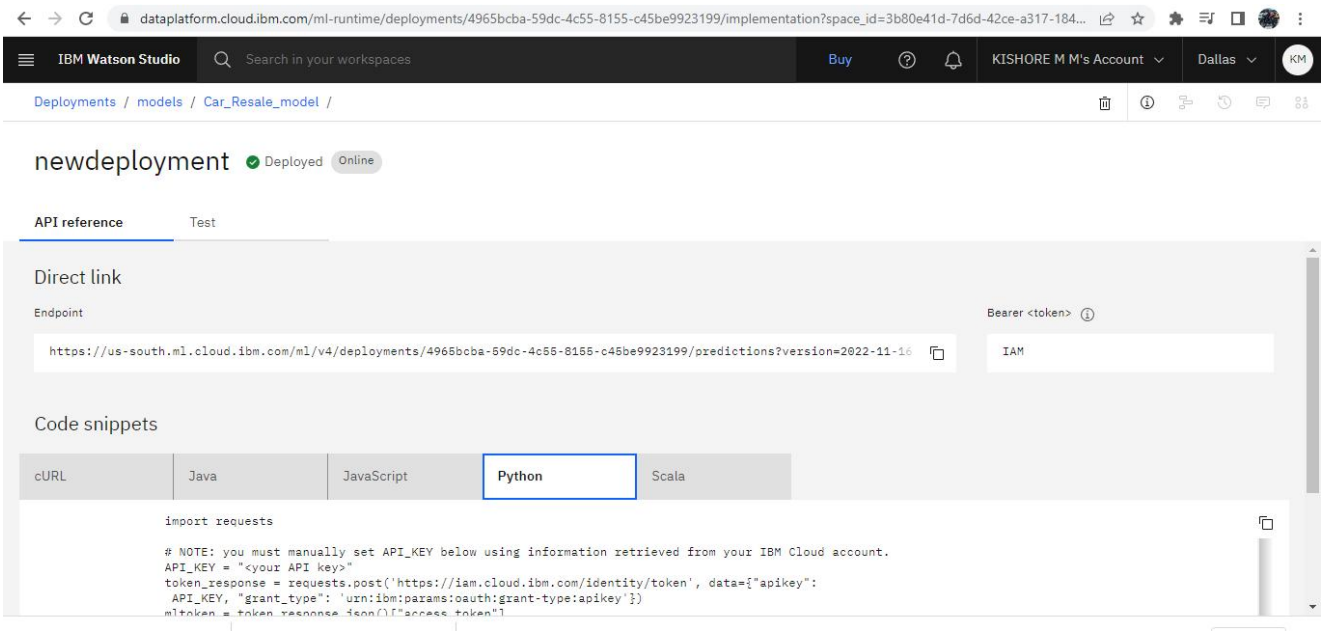




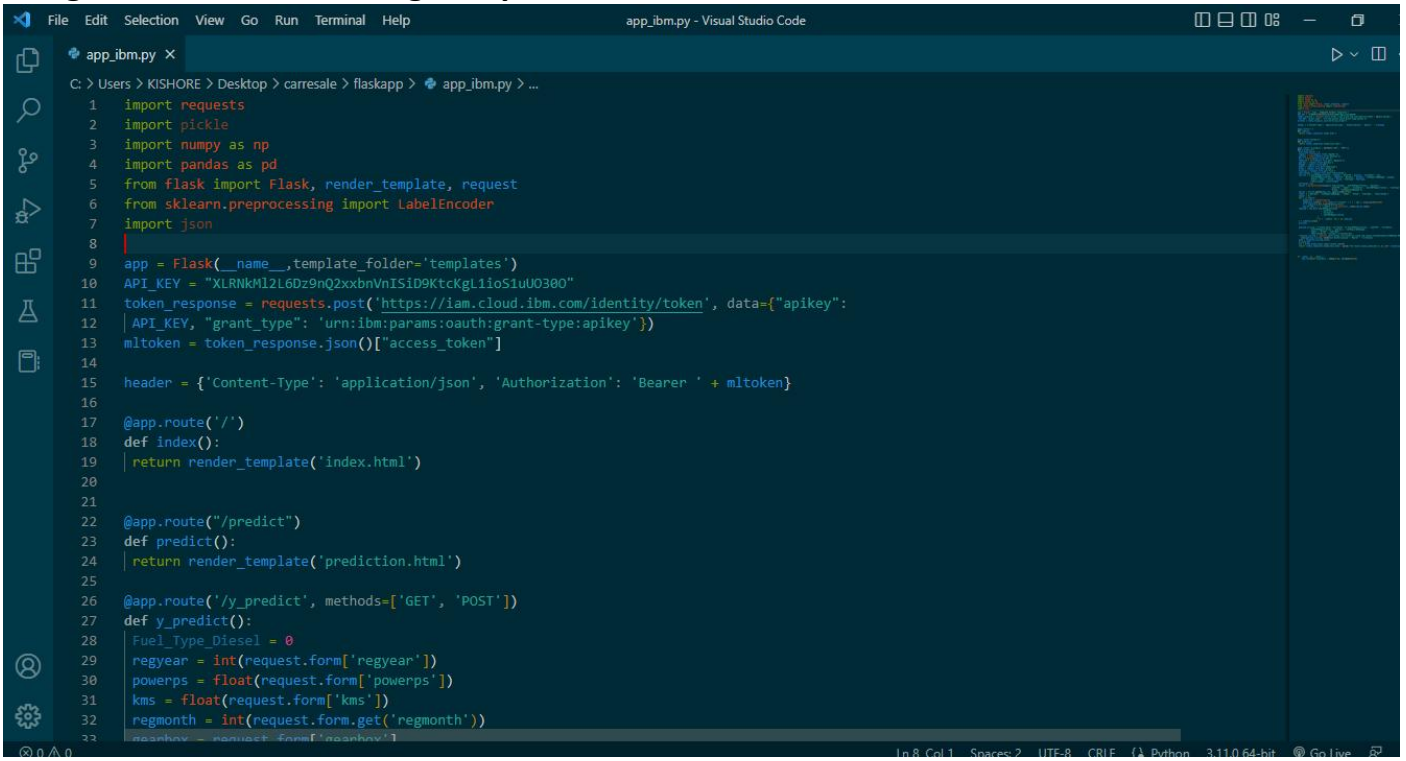



dataplatform.cloud.ibm.com/home2?context=cpdaas
IBM Watson Studio
Search in your workspaces
Buy
KISHORE M M's Account
Dallas
KM
Step through implementing a Data fabric use case in a sample project.
Create a project for your team to prepare data, find insights, or build models.
Stay current with new features, enhancements, and other changes.
Quick start
Create data pipelines with DataStage
Build customer profiles with IBM Match 360 with Watson
Catalog and govern data with Watson Knowledge Catalog
Build and manage ML models with Watson Studio
Query data anywhere with Watson Query
Projects
Car Resale Value Prediction
Today at 12:12 AM
New in gallery
NOTEBOOK
Use AutoAI and Lale to predict credit risk with...
Learn how to use AutoAI experiments in this notebook by getting a German credit data set and train the model to predict banking credit. Then, compare several trained models for
Notifications
Online deployment ready
The online deployment newdeployment in space models is ready to accept
Today at 01:08 AM
Deployments
models
Today at 01:07 AM
Feedback

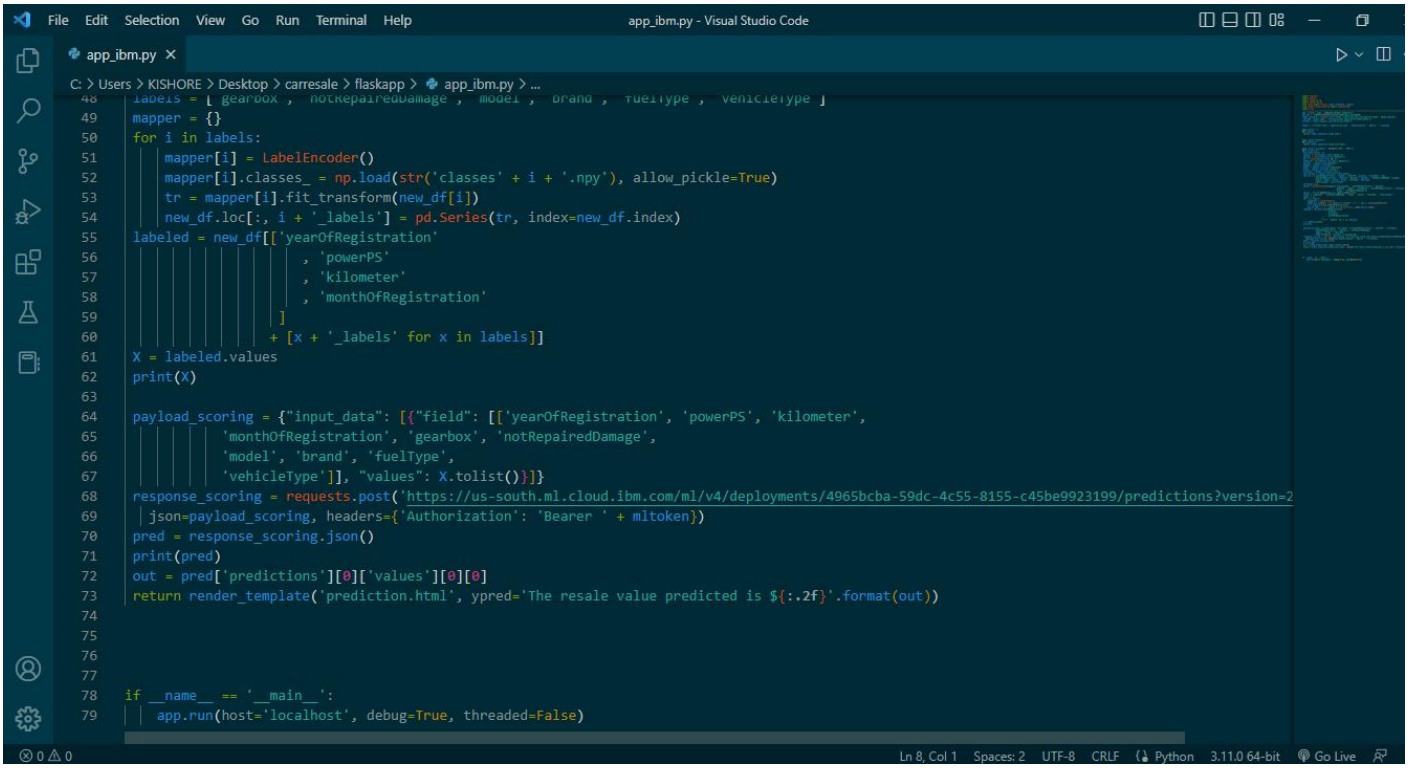

dataplatform.cloud.ibm.com/ml-runtime/spaces/3b80e41d-7d6d-42ce-a317-184bbb596456/deployments?context=cpdaas
IBM Watson Studio
Search in your workspaces
Buy
KISHORE M M's Account
Dallas
KM
Deployments /
models
Overview
Assets
Deployments
Jobs
Manage
Search
Name
Type
Status
Asset
Last modified
newdeployment
Online
Deployed
Car_Resale_model
11 hours ago
KISHORE M M (You)
Items per page: 20
1–1 of 1 items
1 of 1 pages
Drop files here or browse for files to upload.
Stay on the page until upload completes. Incomplete uploads are cancelled.


dataplatform.cloud.ibm.com/ml-runtime/deployments/4965bcba-59dc-4c55-8155-c45be9923199/implementation?space_id=3b80e41d-7d6d-42ce-a317-184...
IBM Watson Studio
Search in your workspaces
Buy
KISHORE M M's Account
Dallas
KM
Deployments / models / Car_Resale_model /
newdeployment
Deployed
Online
API reference
Test
Direct link
Endpoint
https://us-south.ml.cloud.ibm.com/ml/v4/deployments/4965bcba-59dc-4c55-8155-c45be9923199/predictions?version=2022-11-16
Bearer <token>
IAM
Code snippets
cURL
Java
JavaScript
Python
Scala
import requests
# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "<your API key>"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})

Integrate flask with scoring end point



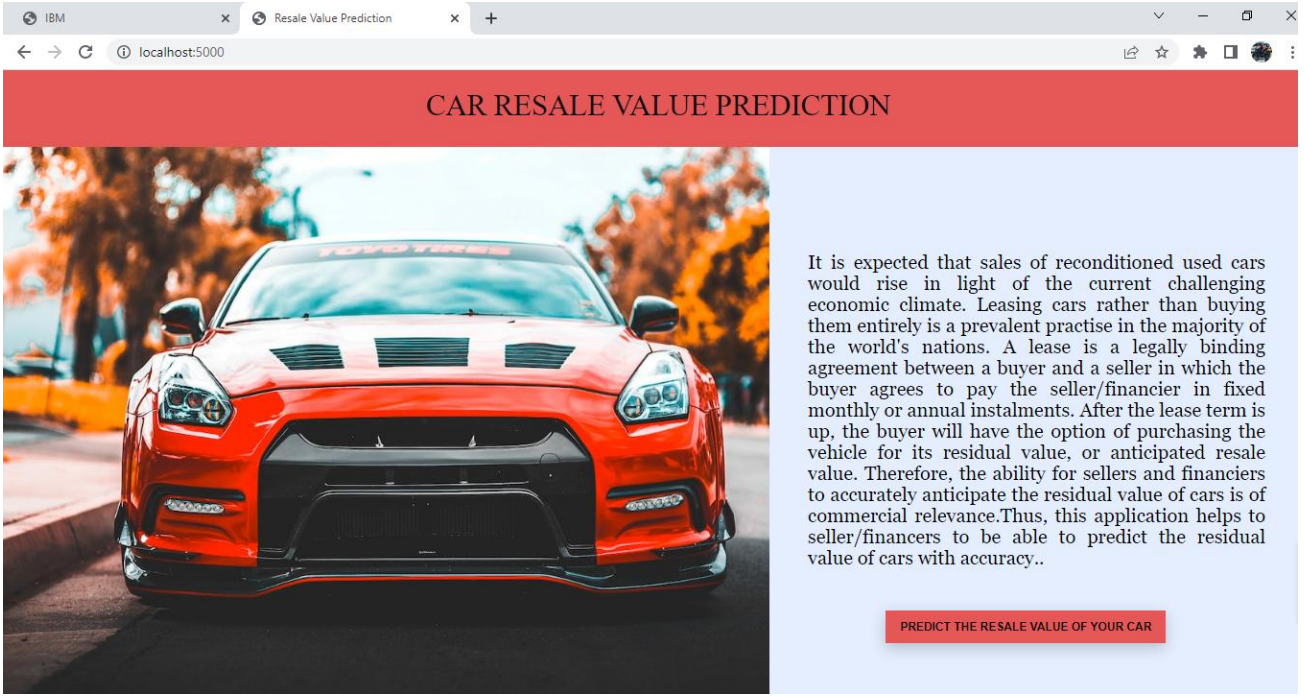
```
import requests
import pickle
import numpy as np
import pandas as pd
from flask import Flask, render_template, request
from sklearn.preprocessing import LabelEncoder
import json

app = Flask(__name__,template_folder='templates')
API_KEY = "XLRNkMl2L6Dz9nQ2xxbnVnISiD9KtcKgL1ioS1uUO30O"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
 API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

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


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

@app.route('/y_predict', methods=['GET', 'POST'])
def y_predict():
 Fuel_Type_Diesel = 0
 regyear = int(request.form['regyear'])
 powerps = float(request.form['powerps'])
 kms = float(request.form['kms'])
 regmonth = int(request.form.get('regmonth'))
```

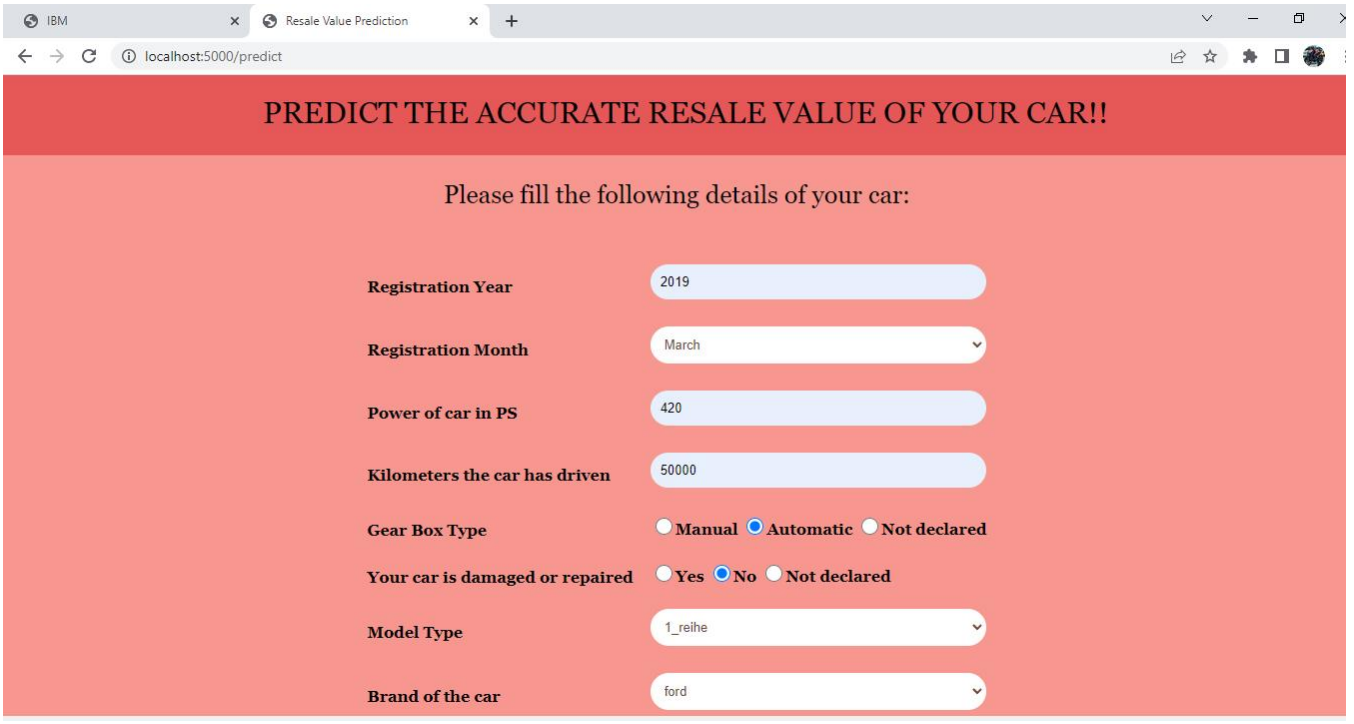


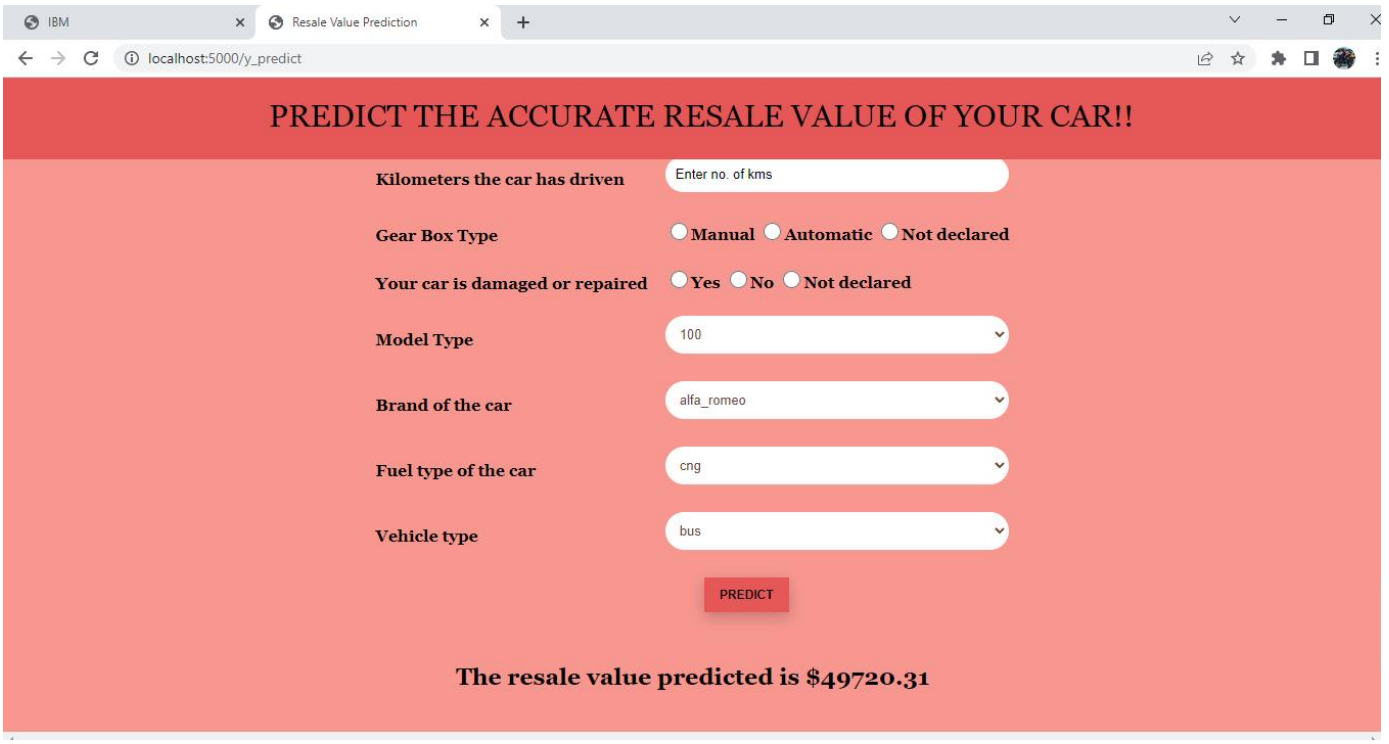
```
labels = ['gearbox', 'notRepairedDamage', 'model', 'brand', 'fuelType', 'vehicleType']
mapper = {}
for i in labels:
    mapper[i] = LabelEncoder()
    mapper[i].classes_ = np.load(str('classes' + i + '.npy'), allow_pickle=True)
    tr = mapper[i].fit_transform(new_df[i])
    new_df.loc[:, i + '_labels'] = 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)

payload_scoring = {"input_data": [{"field": [['yearOfRegistration', 'powerPS', 'kilometer',
       'monthOfRegistration', 'gearbox', 'notRepairedDamage',
       'model', 'brand', 'fuelType',
       'vehicleType']], "values": X.tolist()}]}
response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/4965bcba-59dc-4c55-8155-c45be9923199/predictions?version=2
 json=payload_scoring, headers={'Authorization': 'Bearer ' + mltoken})
pred = response_scoring.json()
print(pred)
out = pred['predictions'][0]['values'][0][0]
return render_template('prediction.html', ypred='The resale value predicted is ${:.2f}'.format(out))



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

OUTPUT SCREENSHOTS

```
C:\Windows\System32\cmd.exe - python  app_ibm.py
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\KISHORE\Desktop\carresale\flaskapp>python app_ibm.py
 * Serving Flask app 'app_ibm'
 * 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://localhost:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 498-621-506
127.0.0.1 - - [19/Nov/2022 08:45:21] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [19/Nov/2022 08:45:23] "GET /static/img.jpg HTTP/1.1" 304 -
127.0.0.1 - - [19/Nov/2022 08:45:23] "GET /favicon.ico HTTP/1.1" 404 -
```




IBM
Resale Value Prediction
localhost:5000/predict
PREDICT THE ACCURATE RESALE VALUE OF YOUR CAR!!
Please fill the following details of your car:
Registration Year
2019
Registration Month
March
Power of car in PS
420
Kilometers the car has driven
50000
Gear Box Type
Manual
Automatic
Not declared
Your car is damaged or repaired
Yes
No
Not declared
Model Type
1_reihe
Brand of the car
ford


IBM
Resale Value Prediction
localhost:5000/y_predict
PREDICT THE ACCURATE RESALE VALUE OF YOUR CAR!!
Kilometers the car has driven
Enter no. of kms
Gear Box Type
Manual
Automatic
Not declared
Your car is damaged or repaired
Yes
No
Not declared
Model Type
100
Brand of the car
alfa_romeo
Fuel type of the car
cng
Vehicle type
bus
PREDICT
The resale value predicted is $49720.31

8. TESTING

8.1.USER ACCEPTANCE TESTING

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT)

				Date	19-Nov-22						
				Team ID	PNT2022TMID02177						
				Project Name	Project - Car Resale Value Prediction						
				Maximum Ma	4 marks						
Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	
InitialScreen_TC_001	Functional	Home Page	Verify user able to see the Prediction		1.Enter URL 2.Click on Prediction button	https://localhost:5000/	Entering into data input page	Working as expected	Pass	Normal test case	
Input_data_TC_002	Functional	Prediction value input page UI	Verify user able to enter input value		1.Check entering into prediction page 2.Check if user can enter value	http://localhost:5000/predict	Application should show below UI elements to enter numeric values: a.Registration Year b.Power of car in PS c.Kilometers the car has driven Software should accept only numeric values	Should allow entering numeric values	Pass	Normal test case	
Input_data_TC_003	Functional	Prediction value input page UI	Verify user able to enter input value		1.Check entering into prediction page 2.Check if user can select option from drop down box	https://localhost:5000/predict	Application should show below UI elements to select from drop down menu: a.Registration Month b.Model Type c.Brand of the car d.Fuel type of the car e.Vehicle Type	should allow selection from pull down menu	Pass	Normal test case	
Input data_TC_004	Functional	Prediction value input page UI	Verify user able to enter input value		1.Check entering into prediction page 2.Check if user can select option from radio button	https://localhost:5000/predict	Application should show below UI elements to select from drop down menu: a.Gear Box Type b.Your car is damaged or repaired	should allow selection from Radio bitton	Pass	Normal test case	
Result_data_TC_008	Functional	Prediction Result Page	Verify No Chronic Kidney Disease (No CKD) test values		1.Enter submit button after entering values 2.Redirect to index page and display correct result	http://localhost:5000/y_predict	Application should show predicted price value of car	should show price of car	Pass	Robustness test case	

.

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	0	1	1	0	2
Duplicate	0	0	0	0	0
External	2	2	0	1	5
Fixed	1	0	0	0	1
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	2	3	2	1	8

3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

8.2.Model Performance Test

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Metrics	Regression Model R Squared = 0.8396847388211943 MAE = 1615.420354114289 MSE = 11204674.310407598 RMSE = 3347.3383919776616	See below

In [39]:
```
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
print ('R Squared =',r2_score(Y_test, y_pred))
print ('MAE =',mean_absolute_error(Y_test, y_pred))
print ('MSE =',mean_squared_error(Y_test, y_pred))
print('RMSE =', mean_squared_error(Y_test, y_pred, squared=False))
```

```
R Squared = 0.8396847388211943
MAE = 1615.420354114289
MSE = 11204674.310407598
RMSE = 3347.3383919776616
```

9. RESULTS

Run Code

In [39]:
```
#fitting the model
regressor.fit(X_train, np.ravel(Y_train,order='C'))
```

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

In [40]:
```
#predicting the values fo test test
y_pred = regressor.predict(X_test)
```

In [41]:
```
#printing the Accuraccy for test set
print(r2_score(Y_test,y_pred))
```

```
0.8387903149180778
```

In [42]:
```
#for testing on user input values
y_pred1 = regressor.predict([[2011,190,125000,5,1,0,163,1,3,3]])
#predticting price for a user input values
print(y_pred1)
```

```
[19070.72434796]
```

In [43]:
```
#saving the model for future use.
filename = 'car_resale_value.sav'
pickle.dump(regressor, open(filename, 'wb'))
```

In [44]:
```
!pip install ibm_watson_machine_learning
```

10. ADVANTAGES & DISADVANTAGES

Advantages:

- Variants usually don't matter in the used car market. If you search well, you can get a top-spec less driven car in the used car market at a price which you would have otherwise paid for a lower variant in case of buying a new car.
- If you buy a car from a brand authorised dealership,you get a warranty on the repair.
- If we are buying a used car that was launched a year ago,you can save up to 20% on its original cost.

Disadvantages:

- Some cars may be lemons. They look fine on the outside but can land in huge repair costs while you use them.
- Be a very informed customer and check each and every possible detail before buying.

11. 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.

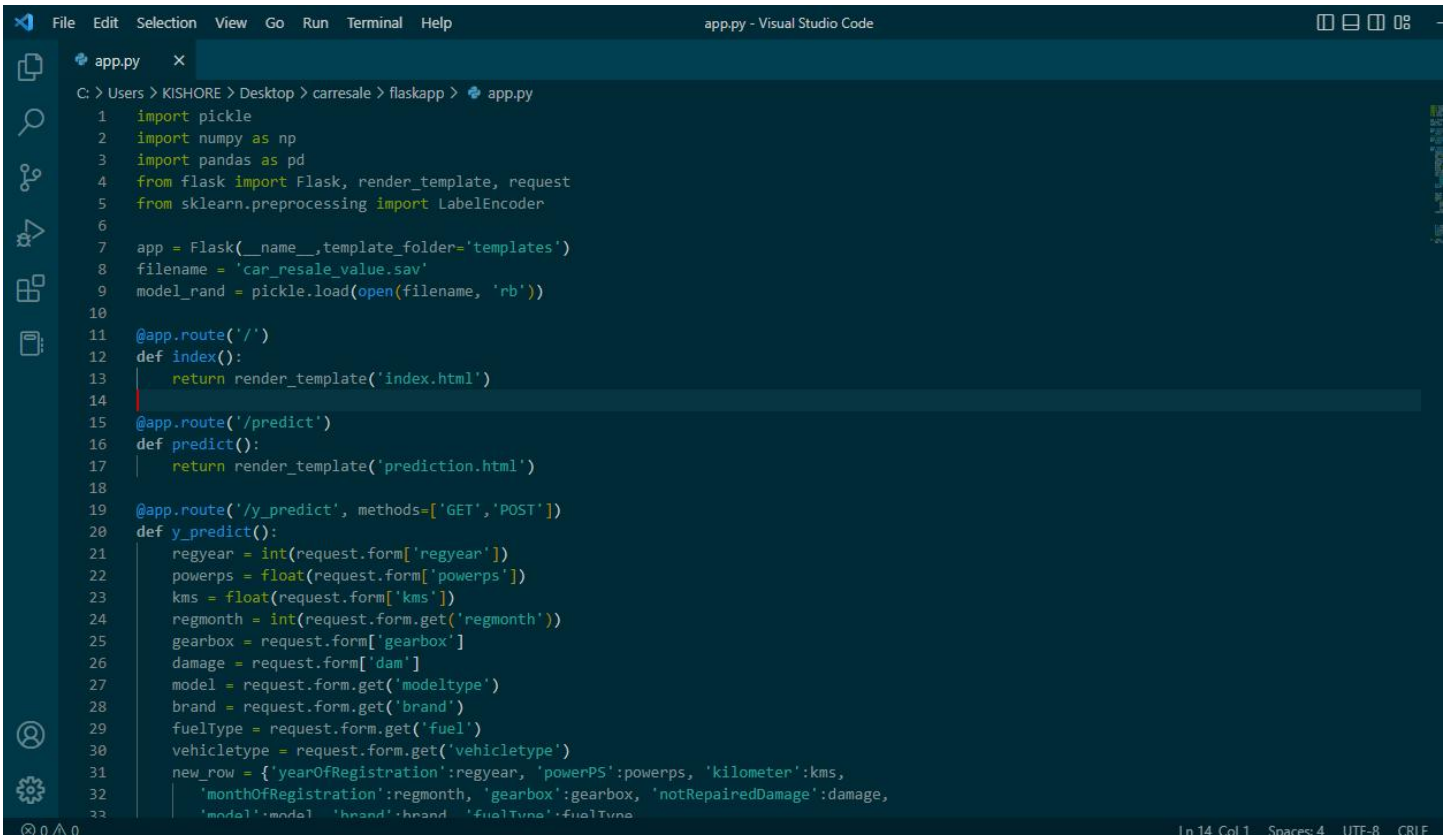
12. FUTURE SCOPE

In 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 of car price which can help to improve 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.

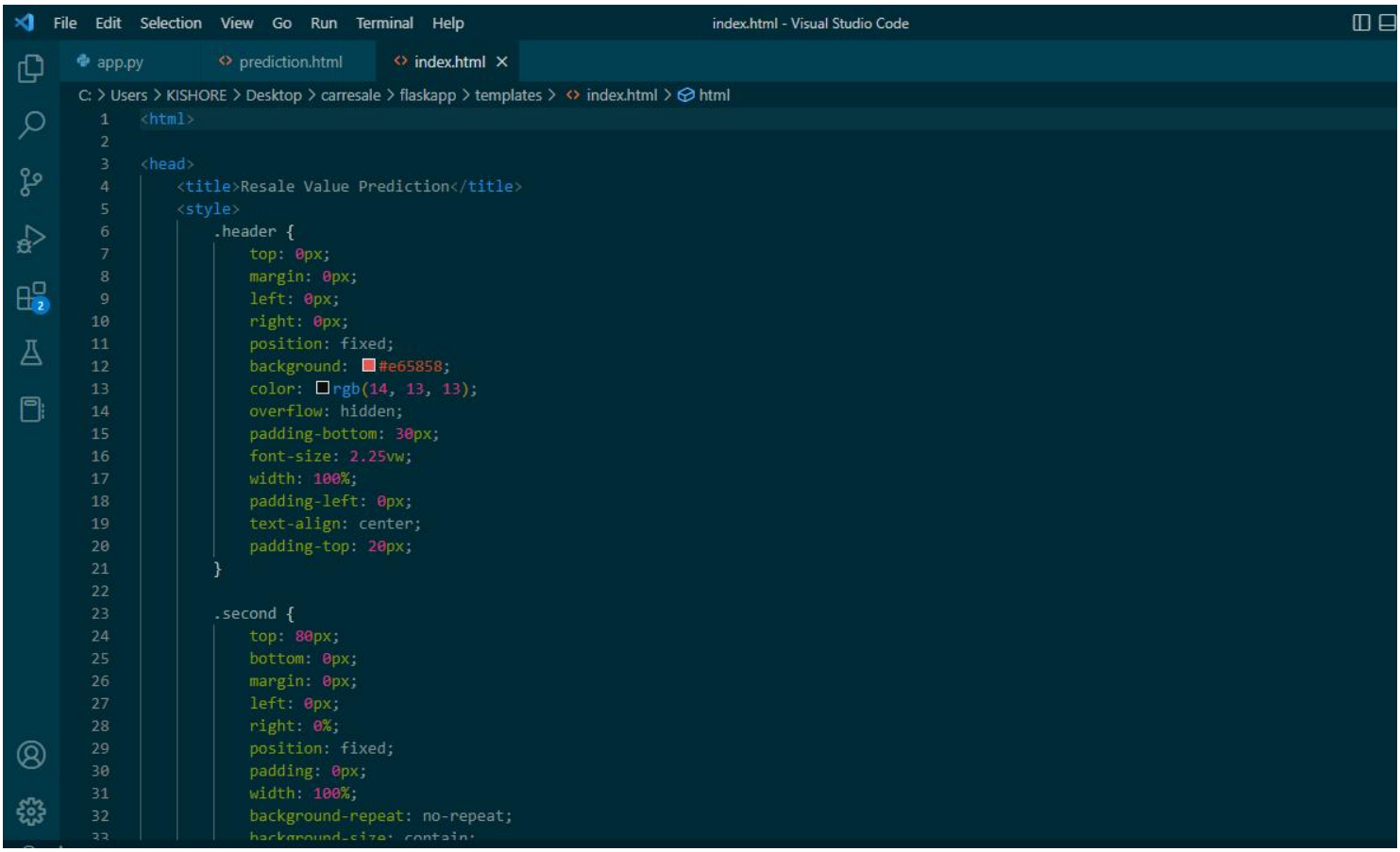
13. APPENDIX

Source code screeenshots:

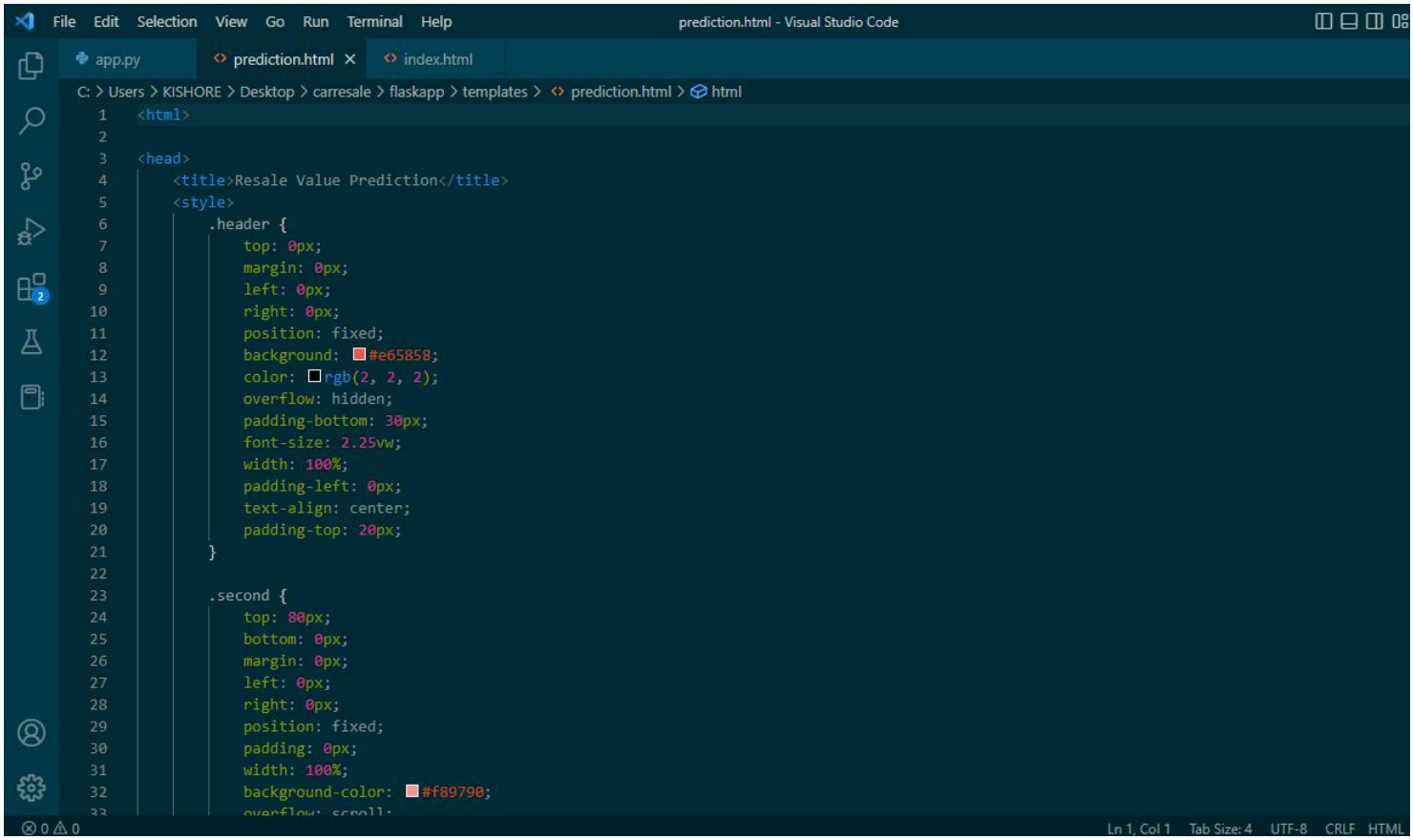
app.py



index.html



prediction.html



GitHub :
https://github.com/IBM-EPBL/IBM-Project-19118-1659693410

Project Demo Link:
https://drive.google.com/file/d/1WujVAXvfQFGS4tPBeRwyPo-HjSwKJPSc/view?usp=share_link