CAR RESALE VALUE PREDICTION

1. Introduction

1.1 Project Overview

This project aims to deliver price prediction models to the public, to help guide the individuals looking to buy or sell cars and to give them a better insight into the automotive sector. Buying a used car from a dealercan be a frustrating and an unsatisfying experience as some dealers are known to deploy deceitful sale tactics to close a deal. Therefore, to help consumers avoid falling victims to such tactics, this study hopes to equip consumers with right tools to guide them in their shopping experience. Another goal of the project is to explore new methods to evaluate used cars prices and to compare their accuracies. Considering this is an interesting research topic in the research community, and in continuing their footsteps, we hope to achieve significant results using more advanced methods of previous work.

1.2 Purpose

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. Literature Survey

2.1 Existing Problem

The customer is unable to locate a platform that estimates the worth of owned vehicles for resale. Therefore, we built this tool for precise forecasting to resell the car.

2.2 References

2.2.1 PRICE PREDICTION OF USED CARS USING MACHINE LEARNING:

PUBLISHED IN: 2021 IEEE International Conference on Emergency Science and Information Technology (ICESIT)

This study seeks to develop a model to forecast used car acceptable pricing based on a number of variables, such as engine size, fuel type, gearbox, road tax, and vehicle mileage. In the used car market, models can help vendors, purchasers, and automakers. Based on the data that users submit, it can eventually output a price prediction that is reasonably accurate. Machine learning and data science are utilized during the model development process. Used automobile listings were scraped for the data collection. To attain the highest accuracy, a variety of regression techniques were used in the study, including linear regression, polynomial regression, support vector regression, decision tree regression, and random forest regression. This project visualized the data to better comprehend the dataset before beginning to develop the model.

2.2.2 VEHICLE RESALE PRICE PREDICTION USING MACHINE LEARNING

PUBLISHED IN: 2021 Juni Khyat ISSN: 2278-4632 (UGC Care Group I Listed Journal)

The primary goal in this research is to analyze the Vehicle Resale Predict and then predict the outcomes using training data. The trade-in automobile market is a steadily expanding industry, and during the past few years, its accurately assessed value has practically quadrupled. With the growth of online marketplaces like CarDheko, Quikr, Carwale, and Cars24, among many others, it is now more important than ever for both the buyer and the seller to understand the trends and case studies that determine the value of the prospective used car. A vehicle's retail value can be predicted using AI computations based on a certain configuration of features. Various websites use different calculations to determine the retail price of There is therefore no comprehensive computation for determining the cost for the trade-in automobiles. One may surely acquire a good sense of the cost without really entering the fine details into the ideal site by creating quantifiable models to predict prices. The main goal of this research is to compare the degrees of precision of three different expectation models to predict the retail price of a used car. The informative index primarily consists of qualitative attributes, two quantitative traits, and test data from academics, including not just results from external tests but also each student's total academic success. When the academic performance test is completed in a large number of universities, there is no system in place that anticipates the understudy's performance in advance. If the understudy fails the exam. Here, we examine an understudy's academic performance in the classroom using SVM analysis and then linear regression analysis, taking into account both internal and external influences. The previous outcomes of the preceding data set were used to make these forecasts.

2.2.3. USED CAR PRICE PREDICTION USING K-NEAREST NEIGHBOR BASE MODEL

PUBLISHED IN: 2020 International Journal of Innovative Research in Applied Sciences and Engineering (IJIRASE).

One of the important and fascinating areas of investigation is the estimation of used car prices. The market for used cars has seen a surge in demand, which has boosted sales for both consumers and sellers. The price of automobiles depends on a number of significant elements, therefore dependable and accurate predictions demand comprehensive understanding of the subject. In order to examine the cost of used automobiles, this research proposed a supervised machine learning model utilizing the KNN regression algorithm. We used data on secondhand automobiles that we downloaded from the Kaggle website to train our model. Through this experiment, several trained and test ratios were used to analyze the data. As a consequence, the proposed model is fitted as the optimum model and has an accuracy of about 85%.

2.2.4. PREDICTION OF USED CAR PRICE BASED ON SUPERVISED LEARNING ALGORITHM

PUBLISHED IN: 2021 International Conference on Networking, Communications and Information Technology (NetCIT).

In order to produce more objective results, they apply machine learning techniques in this work to anticipate the price of secondhand cars with minimal human participation. The dataset is preprocessed using Python's Pycaret module, and the performance of each algorithm is compared using the algorithm comparison function. Extra Trees Regressor and Random Forest Regressor both perform rather well in this study. Using the hyper parameter function, the algorithm was then optimized. To arrive at the final algorithm model, the algorithm was acquired and validated with fresh data. The workflow of the used vehicle market will become faster and more competitive as a result of this algorithm, which will automatically produce used car prices when new used car data enters the used car system. The annual growth rate for new cars in China will be 3.5% in the upcoming five years, while the annual growth rate for used cars will be 5%. The annual growth rate of autos and used cars is rising. Customers believe that when purchasing a new car, they will also take into account the cost of a comparable used vehicle. Particularly, they believe that some value-preserving brand vehicles are more deserving of their attention because they have a changed value and offer customers the best return on their investment. When faced with this circumstance, businesses operating the used automobile market turn to traditional marketing techniques (such as often consulting pricing) to conduct business, which significantly raises the company's running costs.

2.2.5 PRICE EVALUATION MODEL IN SECOND-HAND CAR SYSTEM BASED ON NEURAL NETWORK

PUBLISHED IN: 2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD).

Second-hand vehicles are now the most popular option for consumers looking to purchase cars due to the rapid increase in the number of private vehicles and the rise of the used vehicle market. The internet marketplace for used cars offers both buyers and sellers the opportunity for online P2P exchange. In such systems, whether the seller and the buyer may have a more effective trading experience is largely determined by the accuracy of the second-hand automobile price evaluation. The price evaluation model based on big data analysis is put forth in this research. It makes use of widely disseminated vehicle data as well as a sizable amount of vehicle transaction data to evaluate the pricing data for each type of car using the BP neural network method that has been tuned. In order to determine the price that best fits the car, it attempts to build a model for evaluating used car prices. In this study, the optimal number of hidden neurons in the BP neural network is chosen using the optimized BP neural network algorithm, which increases the prediction model's precision and the network topology's speed of convergence. The fitting curve of the forecast price is compared with the actual transaction price produced from the improved model through the sample simulation experiments. As a result, the accuracy and fitting of the optimized model are both improved.

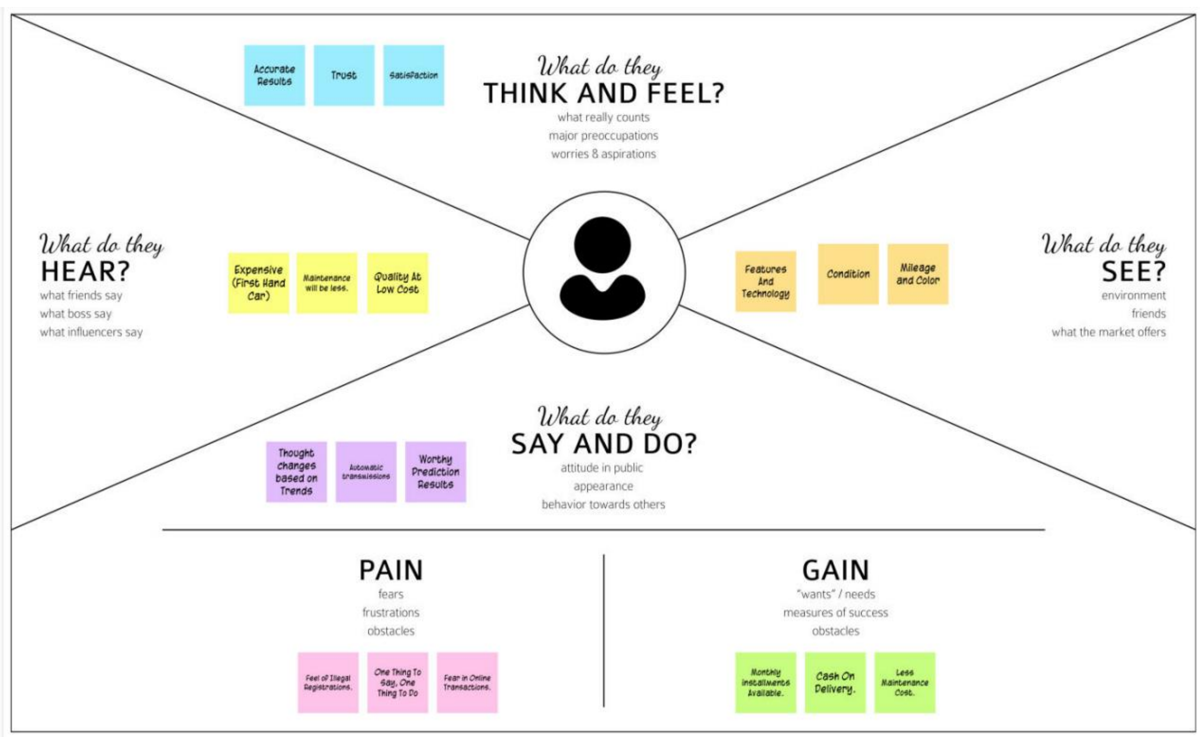
2.3 Problem Statement Definition

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. In order 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. Ideation And Proposed Solution

3.1 Empathy Map Canvas

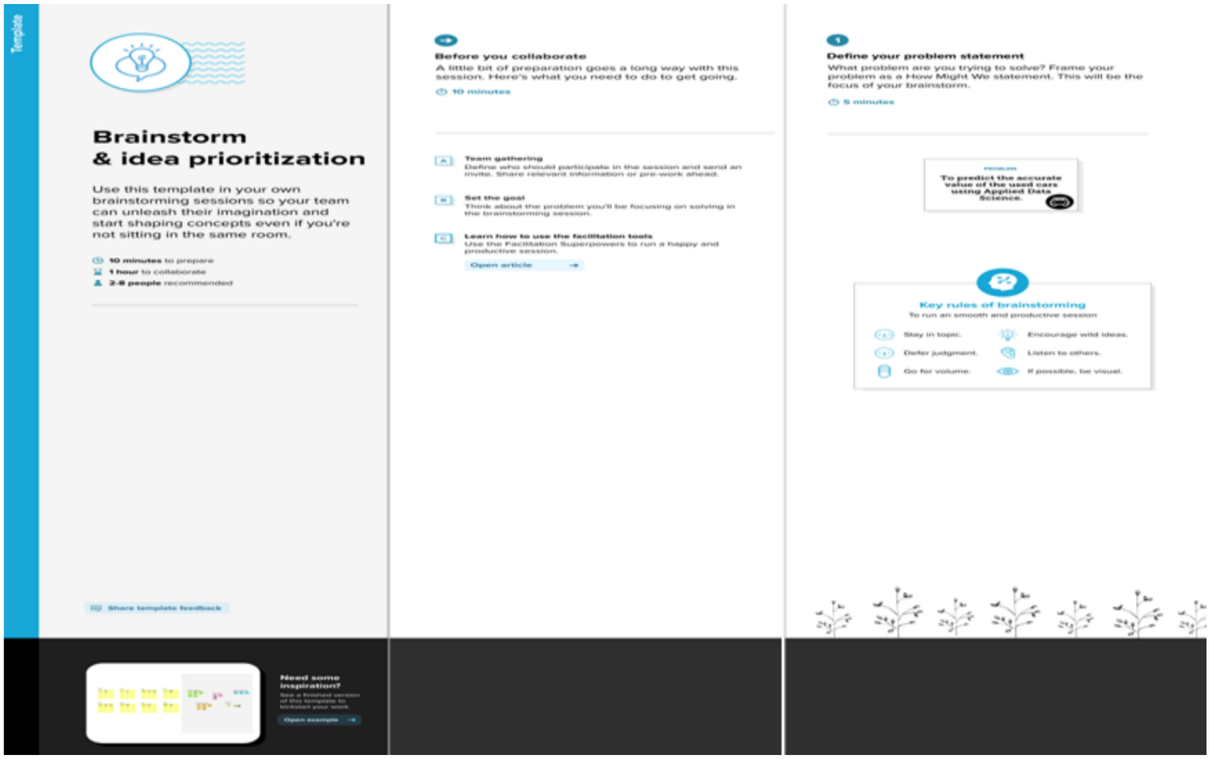
An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise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



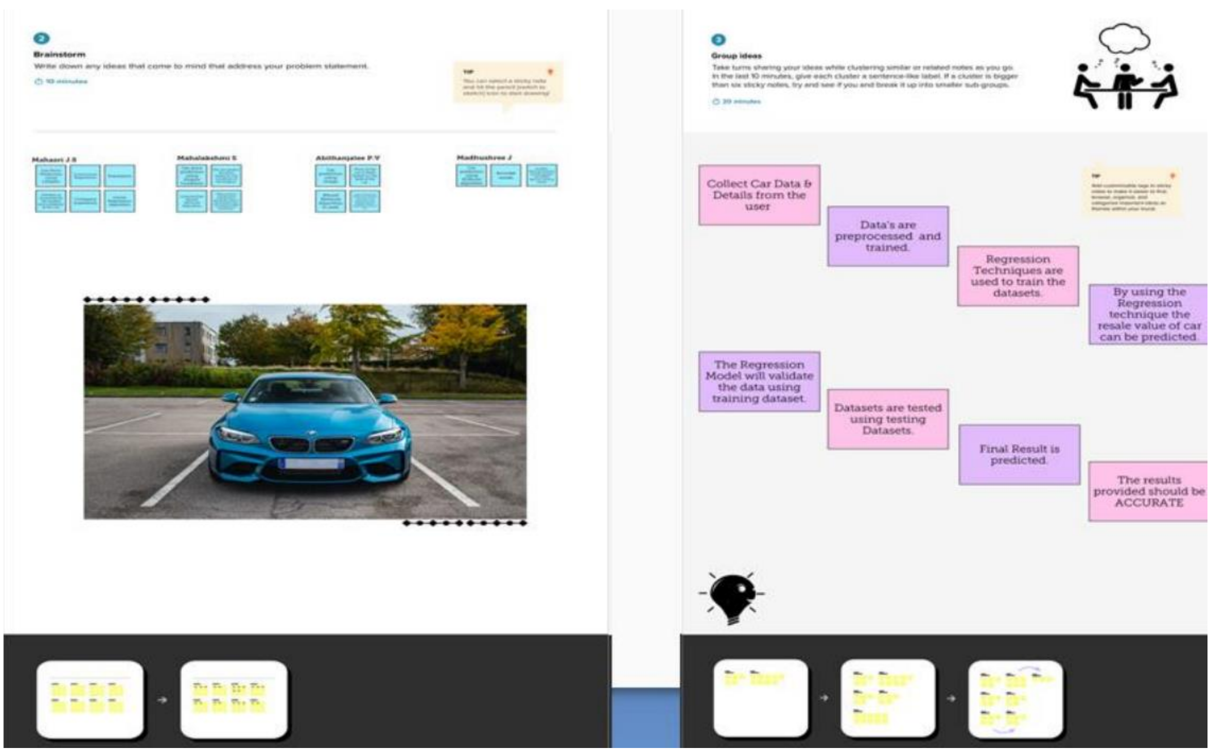
3.2 Ideation And Brainstorming

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. Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

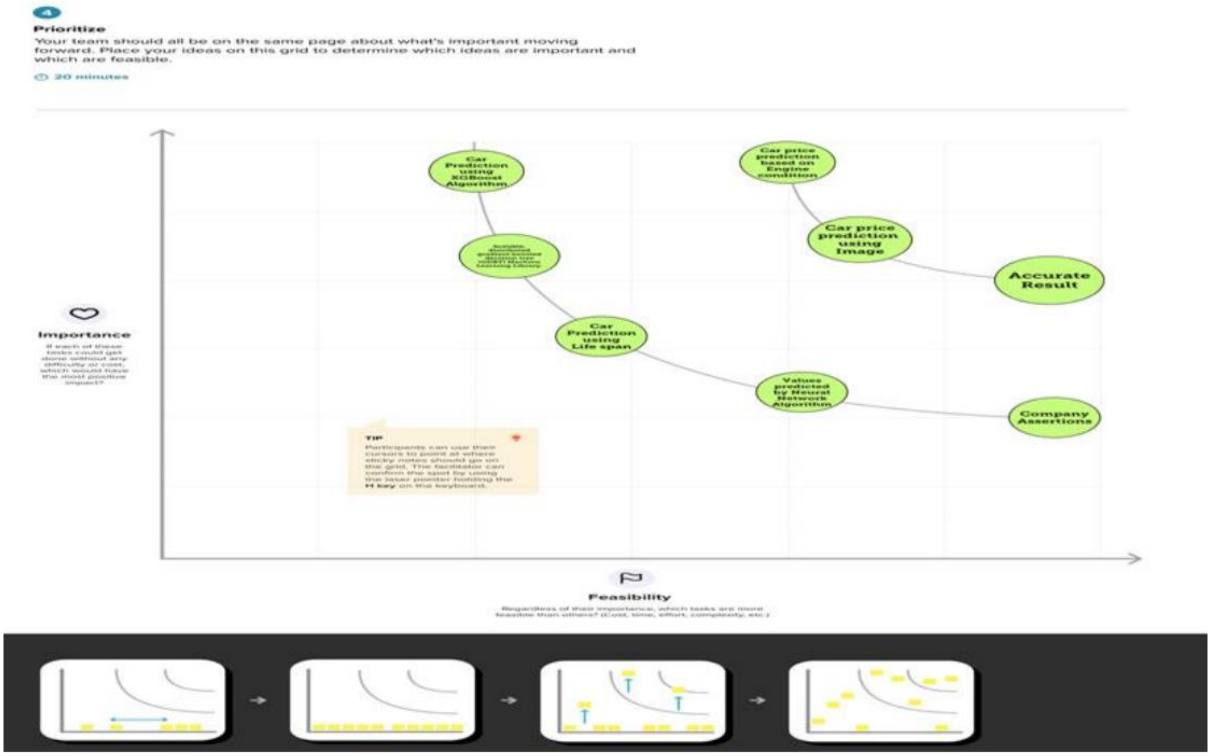
Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization



3.3 Proposed Solution

S.No	Parameter	Description
1	Problem Statement (Problem to be solved)	To predict the resale value of second hand car or used car considering its features.
2	Idea / Solution description	To develop a Machine learning algorithm which predicts the resale value of any used car which is shown in web design.
3	Novelty / Uniqueness	The model predicts the resale value of car with high accuracy.
4	Social Impact / Customer Satisfaction	A good platform with more reliability and portability.
5	Business Model (Revenue Model)	The model deployed in cloud so anyone can access it anywhere and anytime.
6	Scalability of the Solution	It is a web page model so it can be viewed and accessed in both computer as well as mobile phones

3.4 Problem Solution Fit

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem. It helps entrepreneurs, marketers and corporate innovators identify behavioral patterns and recognize what would work and why

Purpose:

❑ Solve complex problems in a way that fits the state of your customers.

❑ Succeed faster and increase your solution adoption by tapping into existing mediums and channels of behavior.

❑ Sharpen your communication and marketing strategy with the right triggers and messaging.

❑ Increase touch-points with your company by finding the right problem-behaviour fit and building trust by solving frequent annoyances, or urgent or costly problems.

❑ Understand the existing situation in order to improve it for your target group.

Template

4. Requirement Analysis

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	Registration through Website
FR-4	Car Information	Getting the car details through Website
FR-5	Value Prediction	Shows the resale value of the car through website

4.2 Non-Functional Requirements

Following are the non-functional requirements of the proposed solution

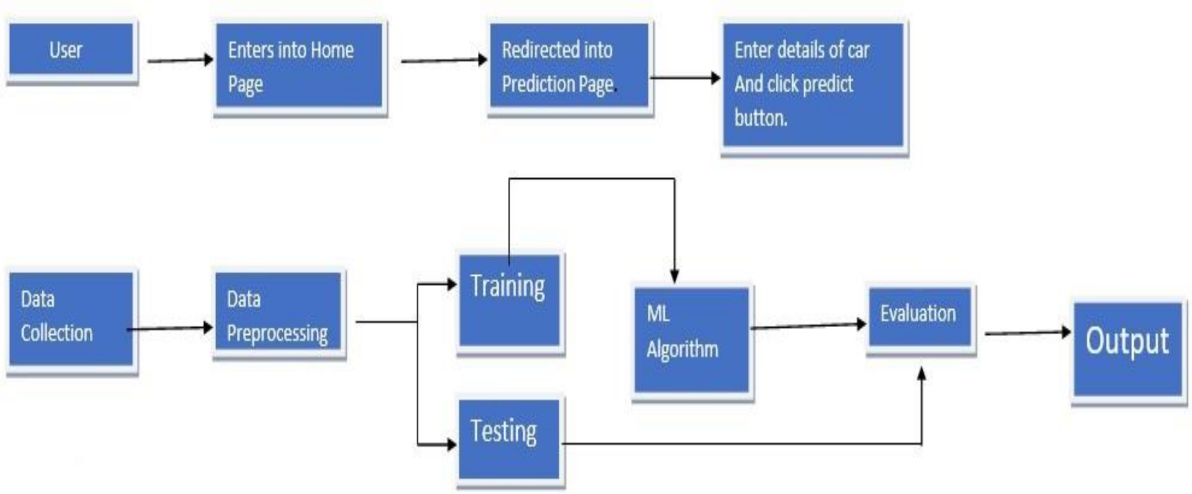
.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The model predicts the resale value of the car with more accuracy.
NFR-2	Security	Protect the user information as well as their car details.
NFR-3	Reliability	The model performs consistently well and also it begins trust to the user.
NFR-4	Performance	The model performance has high accuracy and with portable from one machine to another machine.
NFR-5	Availability	The model can be available anywhere at anytime.
NFR-6	Scalability	The model can be viewed and accessed in both

		computer as well as mobile phone.

5.Project Design

4.3 Data Flow Diagrams

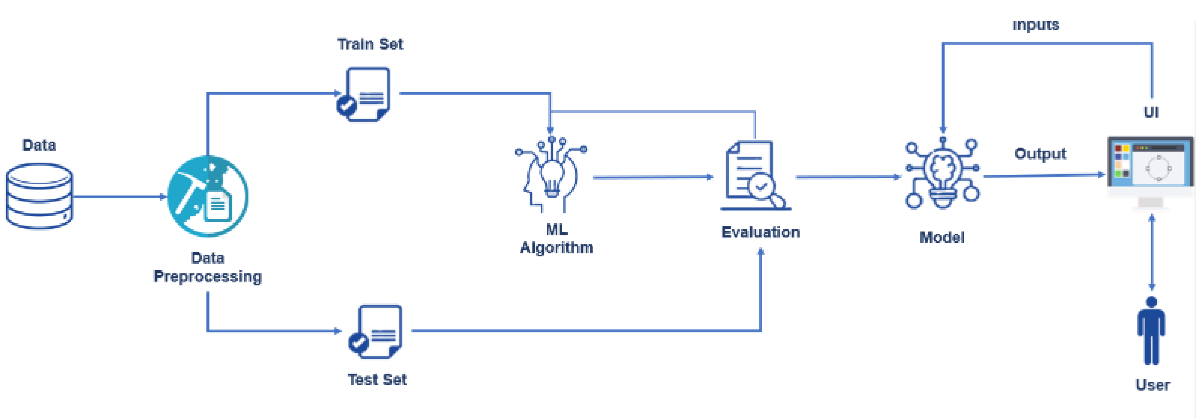


4.4 Solution And Technical Architecture

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.

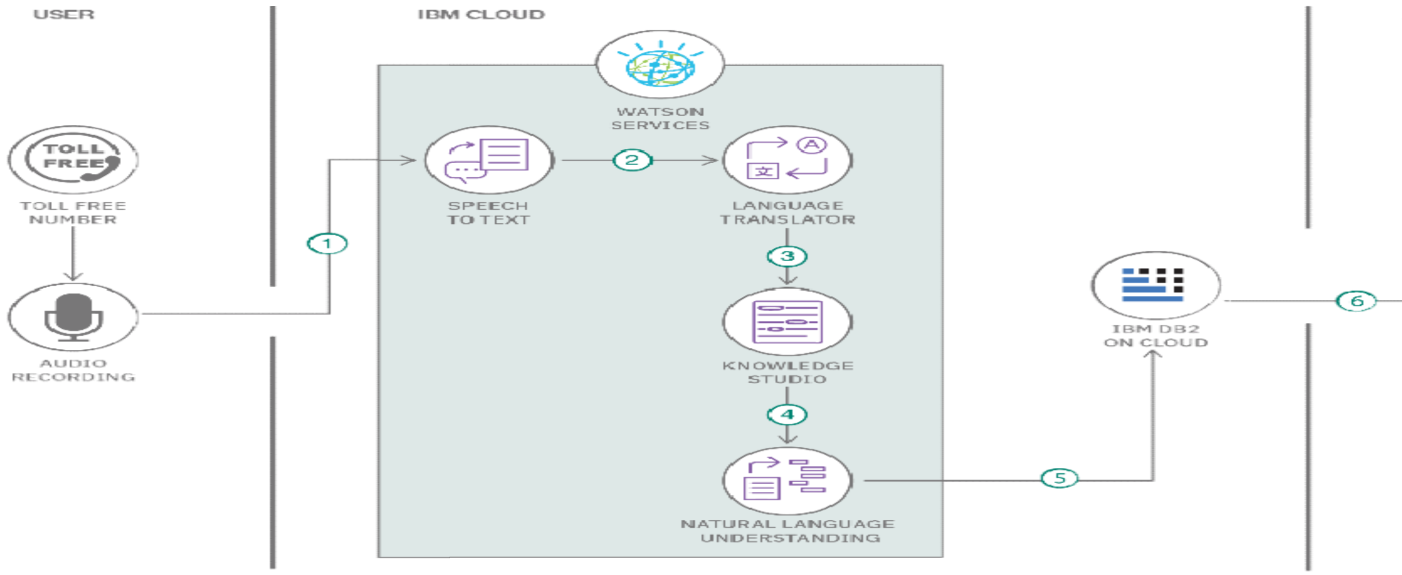
Solution Architecture Diagram:



Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

Example: Order processing during pandemics for offline mode



S.No	Component	Description	Technology	
1.	User Interface	The application interacts with Web UI	HTML, CSS.	
2.	Application Logic-1 **Data Pre-processing**	Clean the dataset in order to remove the duplicate values, fill the missing values and replace the German words with English words.	Python	
3.	Application Logic-2 **Build Python Flask**	Load the model and initialize Flask app. To fetchthe parameter values from the UI, and return the prediction.	Python	
4.	Application Logic-3 **Build an HTML Page**	To take the values from the user in a form and upon clicking on the button for submission it has to redirect to URL for **“y_predict”** which returns the predicted resale value	HTML, CSS.	
5.	Cloud Database	Database Service on Cloud	IBM Cloudant	
6.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem	
7.	External API-1	External API used in the application	IBM Weather API, etc.	
8.	Machine Learning Model	To improve the predictive accuracy and control over-fitting.	Random Forest Regressor Python	
9.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Heroku Platform	

Table-2: Application Characteristics:

No	Characteristics	Description	Technology
1.	Open-Source Frameworks	To establish an connection between the flask andan HTML page.	Python Flask
2.	Security Implementations	To Protect the user information as well as their car details.	SHA-256, Encryptions
3.	Scalable Architecture	The model can be viewed and accessed in both computer as well as mobile phone.	Web UI, Mobile Android app
4.	Availability	The model can be available anywhere at any time.	IBM Cloud
5.	Performance	The model performance has high accuracy and with portable from one machine to another machine.	HTML,CSS

4.5 User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Web Browser	USN-1	As a user, I can visit to the website directly.	I can access the website by simply clicking available link .	High	Sprint-1
		USN-2	I can move to the homepage	I can able to visit the webpage without any acceptance.	High	Sprint-1
		USN-3	After reading the description of the model I can move to the prediction page by clicking the prediction button(RESALE VALUE OF YOUR CAR)	I can move to prediction page without any acceptance	High	Sprint-2
		USN-4	After filling the details in the prediction pagethe	I can get the result without any	High	Sprint-3

			accurate value should be shown in the webpage	acceptance.		
Customer(Mobile User)	Mobile app(Sign up)	USN-1	As a user can register for the application bygiving email as a username and setting a password	I can register if the username is in the correct format.	Medium	Sprint-4
	(Sign in)	USN-2	As a user I can login to the app by filling theusername and password field.	I can login to the app if the username and password matches with database	Medium	Sprint -5
Customer(Mobile User)	Dashboard	USN-3	As a user I can move to the dashboard aftersuccessful login and navigate to next page	Without any acceptance I can move to the next page.	Medium	Sprint-5
		USN-4	After filling the required details click predictbutton to get the result.	Without any acceptance I can get the result	Medium	Sprint-6

5. Project Planning And Scheduling

5.1 Sprint Planning & Estimation

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.	22 SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	24 SEPTEMBER2022

Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	25 SEPTEMBER2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	27 SEPTEMBER 2022
Problem Solution Fit	Prepare problem - solution fit document.	27 SEPTEMBER 2022
Solution Architecture	Prepare solution architecture document.	27 SEPTEMBER 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).	6 OCTOBER 2022
Functional Requirement	Prepare the functional requirement document	7 OCTOBER 2022
Technology Architecture	Prepare the technology architecture diagram.	7 OCTOBER2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review	9 OCTOBER 2022
Prepare Milestone & ActivityList	Prepare the milestones & activity list of the project	26 OCTOBER 2022
Project Development - Delivery of Sprint-1, 2, 3 & 4	Develop & submit the developed code by testing it	IN PROGRESS

6.2 Sprint Delivery Schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Collect Dataset	USN-1, USN-2, USN-3, USN-4	Download the dataset.	1	Low	J S Mahasri, S Mahalaskhmi, J Madhushree, P V Abithanjalee
	Pre-process Dataset		Import required Libraries	1	Low	
			Read the dataset.	1	Low	
			Cleaning the dataset.	2	Medium	
			Split data into Independent and Dependent Variables.	2	Medium	
Sprint-2	Model Building	USN-1, USN-2, USN-3, USN-4	Apply Random Forest Regressor model	3	High	J S Mahasri, S Mahalaskhmi, J Madhushree, P V Abithanjalee
			.Save the Random Forest Regressor model	2	Medium	
Sprint-3	Application Buliding	USN-1, USN-2, USN-3, USN-4	Build the Python Flask	3	High	J S Mahasri, S Mahalaskhmi, J Madhushree, P V Abithanjalee
			Build the HTML Page	3	High	
			Execute and Test	3	High	
Sprint-4	Train the Model On IBM	USN-1, USN-2, USN-3, USN-4	Train The ML Model On IBM	3	High	J S Mahasri, S Mahalaskhmi, J Madhushree, P V Abithanjalee
			Integrate Flask	3	High	

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	31 Oct 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	07 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	14 Nov 2022

7 CODING AND SOLUTIONING

7.1 Preprocessing the Dataset

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

#READING THE DATASET
df=pd.read_csv("autos.csv",header=0,sep=',',encoding='Latin1',)


print(df.seller.value_counts())
df[df.seller!='gewerblich']

df=df.drop('seller',1)

print(df.offerType.value_counts())
df[df.offerType!='Gesuch']

df=df.drop('offerType',1)

print(df.shape)

df=df[(df.powerPS>50)&(df.powerPS<900)]
print(df.shape)

df=df[(df.yearOfRegistration>=1950)&(df.yearOfRegistration<2017)]

print(df.shape)

df.drop(['name','abtest','dateCrawled','nrOfPictures','lastSeen','postalCode','dateCreated'],axis='colum
ns',inplace=True)
```

In [17]:

```
new_df=df.copy()
new_df=new_df.drop_duplicates(['price','vehicleType','yearOfRegistration','gearbox','powerPS','mod
el','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','kombi','andere'),('small
car','convertible','combination','others'),inplace=True)
new_df.notRepairedDamage.replace(('ja','nein'),('Yes','No'),inplace=True)

#FILLING MISSING VALUES USING fillna() METHOD.
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")
new_df.head()
labels=['gearbox','notRepairedDamage','model','brand','fuelType','vehicleType']
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 )
#Final data to be put in a new dataframe called " LABELED " ,
labeled = new_df [ [ 'price','yearOfRegistration','powerPS','kilometer','monthOfRegistration']+ [ x + "
_labels " for x in labels ] ]
print ( labeled.columns )

Y = labeled.iloc [ : , 0 ] .values
X = labeled.iloc [ : , 1 : ] . values
#need to reshape the Y values
Y = Y.reshape ( -1,1 )

#SPLITTING THE DATASET
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 )
```

7.2 Feature 2

Applying Random Forest Regressor

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

```
regressor.fit(X_train, np.ravel(Y_train,order='C'))


y_pred = regressor.predict(X_test)
print(r2_score(Y_test,y_pred))
```

8.TESTING

User Acceptance Testing

1. Purpose of Document

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

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	10	2	3	5	20
Duplicate	1	1	3	0	5
External	4	0	5	1	10
Fixed	9	5	8	15	37
Not Reproduced	0	2	1	0	3
Skipped	1	0	0	1	2
Won't Fix	2	4	2	0	8
Totals	26	12	11	26	75

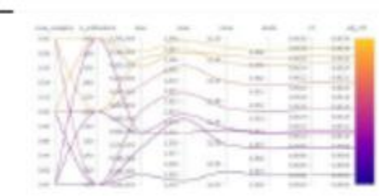
Test Analysis

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	8	0	0	8
Client Application	10	0	0	10
Security	2	0	0	2

Outsource Shipping	4	0	0	4
Exception Reporting	7	0	0	7
Final Report Output	3	0	0	3
Version Control	2	0	0	2

9.Results

9.1 Performance Metrics

S.No.	Parameter	Values	Screenshot
1.	Metrics	**Regression Model:** MAE - , MSE - , RMSE - , R2 score	
2.	Train the Model on IBM cloud	Integrate Python Flask - Validation Method -	
3.	Build the HTML page	Execute- Validate Results	

10.Advantages and Disadvantages

Advantages

With the help of our platform user can get the accurate price of their car resale price. We developed only website based prediction so anyone can get the value without any authentication.

Disadvantages

We didn't developed for mobile application if we use mobile for prediction it will be more useful than website based one.

11.Conclusion

Thus we predicted the accurate value of the preowned car using machine learning algorithm and Flask. By doing this project got a knowledge in machine learning and webpage designing.

12.Future Scope

We will implement it in android os based mobile application and we will develop other features like review gathering .

13.Appendix

Source Code

Python note book

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

#READING THE DATASET
df=pd.read_csv("autos.csv",header=0,sep=',',encoding='Latin1',)

print(df.seller.value_counts())
df[df.seller!='gewerblich']

df=df.drop('seller',1)

print(df.offerType.value_counts())
df[df.offerType!='Gesuch']

df=df.drop('offerType',1)

print(df.shape)

df=df[(df.powerPS>50)&(df.powerPS<900)]
print(df.shape)

df=df[(df.yearOfRegistration>=1950)&(df.yearOfRegistration<2017)]

print(df.shape)

df.drop(['name','abtest','dateCrawled','nrOfPictures','lastSeen','postalCode','dateCreated'],axis='colum
ns',inplace=True)
```

In [17]:

```
new_df=df.copy()
new_df=new_df.drop_duplicates(['price','vehicleType','yearOfRegistration','gearbox','powerPS','mod
el','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','kombi','andere'),('small
car','convertible','combination','others'),inplace=True)
new_df.notRepairedDamage.replace(('ja','nein'),('Yes','No'),inplace=True)

#FILLING MISSING VALUES USING fillna() METHOD.
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")
new_df.head()
labels=['gearbox','notRepairedDamage','model','brand','fuelType','vehicleType']
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 )
#Final data to be put in a new dataframe called " LABELED " ,
labeled = new_df [ [ 'price','yearOfRegistration','powerPS','kilometer','monthOfRegistration']+ [ x + "
_labels " for x in labels ] ]
print ( labeled.columns )

Y = labeled.iloc [ : , 0 ] .values
X = labeled.iloc [ : , 1 : ] . values
#need to reshape the Y values
Y = Y.reshape ( -1,1 )

#SPLITTING THE DATASET
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 )
```

Applying Random Forest Regressor

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score
regressor = RandomForestRegressor(n_estimators=1000,max_depth=10,random_state=34)
regressor.fit(X_train, np.ravel(Y_train,order='C'))



y_pred = regressor.predict(X_test)
print(r2_score(Y_test,y_pred))
```

Code for python file(app.py)

```
from flask import Flask, render_template, Response, request
import pickle
from sklearn.preprocessing import LabelEncoder
import numpy as np
import pandas as pd
app=Flask(__name__)
filename='resale_model.sav'
model_rand=pickle.load(open(filename,'rb'))
@app.route('/')
def index():
        return render_template('resaleintro.html')

@app.route('/resalepredict')
def predict():
        return render_template('resalepredict.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('modeltype')
    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'))
      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('rescalepredict.html',ypred="The resale value predicted is
{:.2f}$".format(y_prediction[0]))
if __name__ == '_main_':
  app.run(host='localhost',port=8000, debug=True)
```

GITHUB LINK & VIDEO DEMO LINK