PROJECT REPORT

PREDICTING USED CAR PRICES USING MACHINE LEARNING.

Team Members

Bhargava Rathod (bhargavarathod@my.unt.edu)

SaiPrathyushaVeguru (saiprathyushaveguru@my.unt.edu)

Sheshidhar Reddy Shaga (sheshidharreddyshaga@my.unt.edu)

Sai Meghana Eravelli (saimeghanaeravelli@my.unt.edu)

Likith Chowdary Nuthi (likithchowdarynuthi@my.unt.edu)

Contents:

1. Abstract ………………………………………. 3

2. Data specification …………………………………. 3

3. Design and Milestones ……………………………. 3

3.1 Training a Regression Model

3.2 Deployment of project using Heroku

4. Results and Analysis ……………………………… 4

5. Repository / Achieve ……………………………… 12

6. CODE …..………………………………………… 12

1. Abstract: The core objective of this project is to recommend the selling price of a used car. As many people want to sell their used car but most of them are not sure how much is the value of their car. In such cases our project will be more suitable for the consumers. Here we take

2. Data specification: We have used the dataset which was downloaded from the Kaggle repository (https://www.kaggle.com/nehalbirla/vehicle-dataset-from-cardekho). The dataset contains 7 independent variables and 1 dependent variable. The below are the details.

Independent Variables:

1. Car_Name : Name of the Car
2. Year: Year of purchase
3. Km_driven: Number of KMs vehicle travelled
4. Fuel: Fuel type like Petrol, Diesel, CNG
5. Seller_Type: The type of the seller like Dealer or Individual
6. Transmission: The type of vehicle transmission like Automatic, Manual.
7. Owner: Number of owners of the vehicle.
8. Present_price: It the present price of the new car basically Ex showroom price

Dependent Variable:

- Selling_Price: The selling price of the vehicle.

3. Design and Milestones: Our whole project can be categorized into 2 modules

3.1 Training a Regression Model.
3.2 Deployment of Project using Heroku.

3.1.Training a Regression Model: In this section we have trained multiple models from which we have used the best model. We have used a jupyter notebook for training the model.

Before training we performed EDA, Pre-processing, Vectorization and Feature Engineering.

- **Insights from EDA:**
  - The dataset has 309 data points.
  - Name, Fuel, Seller_type, Transmission are the categorical variables and rest are numerical.

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 301 entries, 0 to 300
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Car_Name       301 non-null    object
 1   Year           301 non-null    int64
 2   Selling_Price  301 non-null    float64
 3   Present_Price  301 non-null    float64
 4   Kms_Driven     301 non-null    int64
 5   Fuel_Type      301 non-null    object
 6   Seller_Type    301 non-null    object
 7   Transmission   301 non-null    object
 8   Owner          301 non-null    int64
dtypes: float64(2), int64(3), object(4)
memory usage: 21.3+ KB
```

❖ The dataset has zero null values.

- **Pre-processing:**
  - ❖ We have converted all car names to lowercase.
  - ❖ We have replaced space with underscore in car names.
  - ❖ We have done standard scaling for KMS driven

- **Vectorization and Feature Engineering:**
  - ❖ For all categorical variables except Car_Name we have used label encoding.
  - ❖ For the Car_Name variable we have used one hot encoding.
  - ❖ Since the dataset is from 2020 year, we have created a new feature i.e., **num_years** (number of years since vehicle first purchase)

- **Modelling:**
  - ❖ In Modelling We have tried multiple regression along with cross validation out of which we have picked the best model which gives less mean squared error. Find the below Models and Mean Squared Errors.

Model	**Mean Squared Error**
Linear Regression	1.23
Lasso	3.33
Ridge	1.52
KNeighborsRegressor	1.15
DecisionTreeRegressor	2.53
RandomForestRegressor	1.69
ExtraTreesRegressor	0.59

```
ExtraTrees_model = RS_CV.best_estimator_
predicted = ExtraTrees_model.predict(X_te)
```

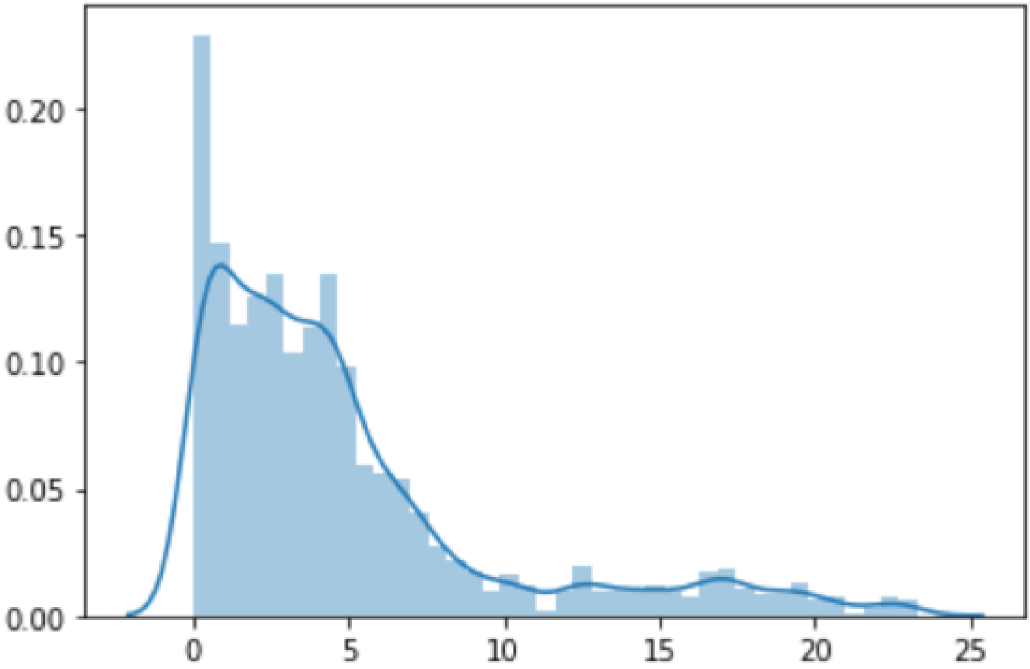
```
mean_squared_error(y_test,predicted)
```

```
0.5988856040667142
```

Distribution Plot

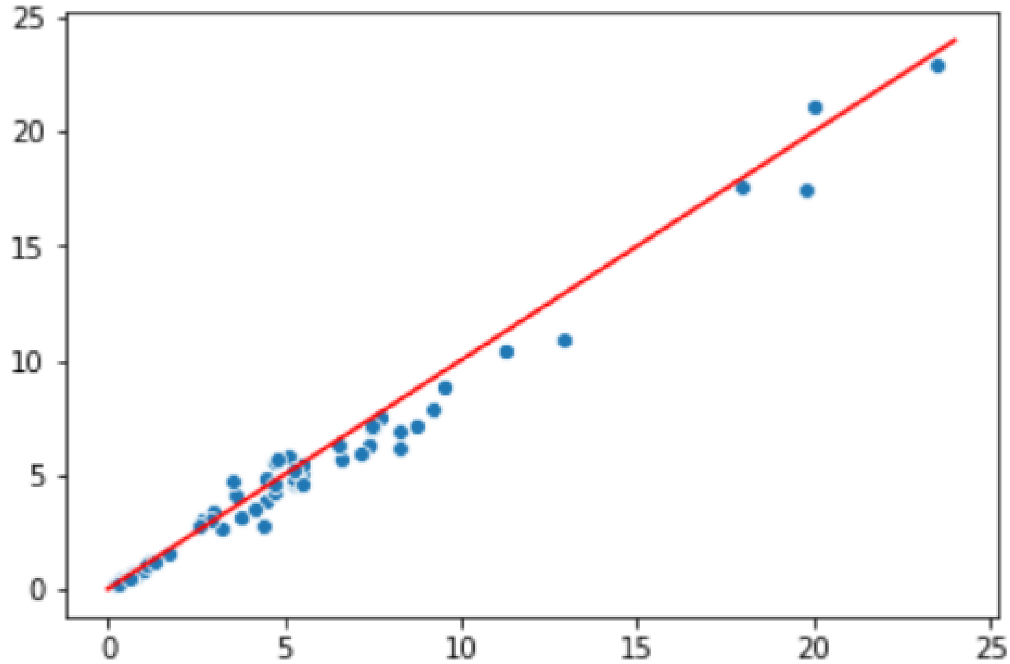
```
sns.distplot(abs(y_test-predicted))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2317f349948
```



Scatter Plot

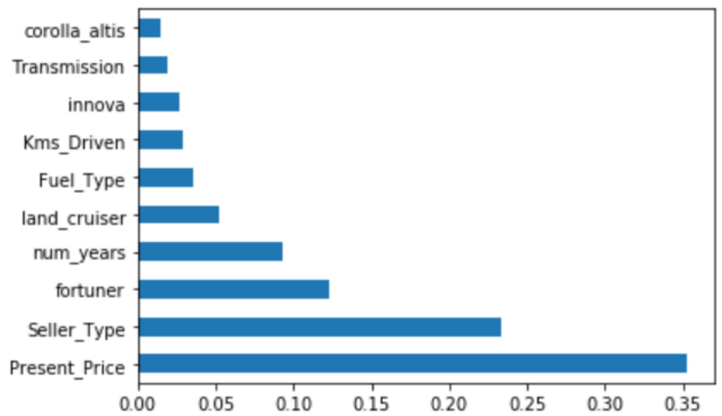
```
sns.scatterplot(y_test.reshape(-1),predicted)
plt.plot(line,color = 'r')
plt.show()
```



- The above red line represents the right prediction. Points closer to the line are having less error.

Top 10 important Features

```
#plot graph of feature importances for better visualization
feat_importances = pd.Series(ExtraTrees_model.feature_importances_, index=columns)
feat_importances.nlargest(10).plot(kind='barh')
plt.show()
```



- **<u>Saving The Model, Scaler and Count vectorizer:</u>**
    - After the whole training process, we have picked the best model as ExtraTreesRegressor. Now we have saved the model as well as Scalar and Count vectorizer in a pickle file(.pkl).

```
import pickle
# open a file, where you ant to store the data
file = open('ExtraTrees_model.pkl', 'wb')

# dump information to that file
pickle.dump(ExtraTrees_model, file)
file.close()
```

```
file = open('scaler.pkl', 'wb')

# dump information to that file
pickle.dump(scaler, file)
file.close()
```

```
file = open('vectorizer.pkl', 'wb')

# dump information to that file
pickle.dump(vectorizer, file)
file.close()
```

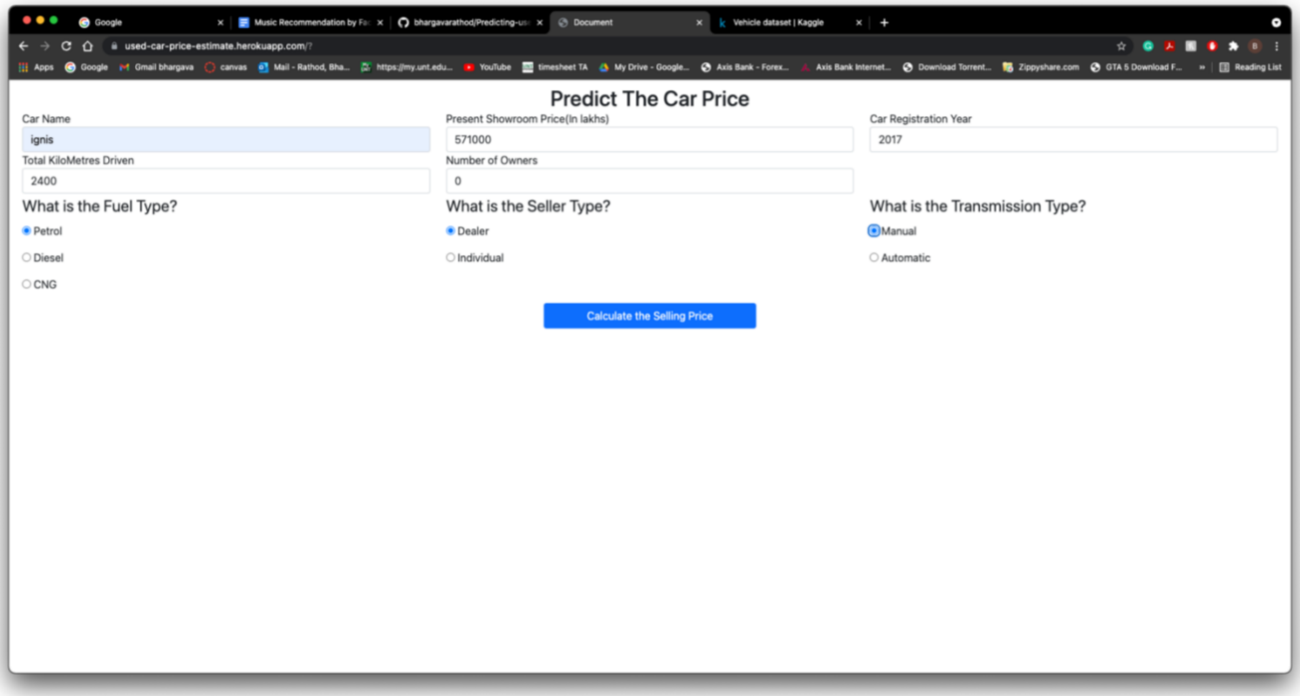
3.2 Deployment of Project using Heroku:

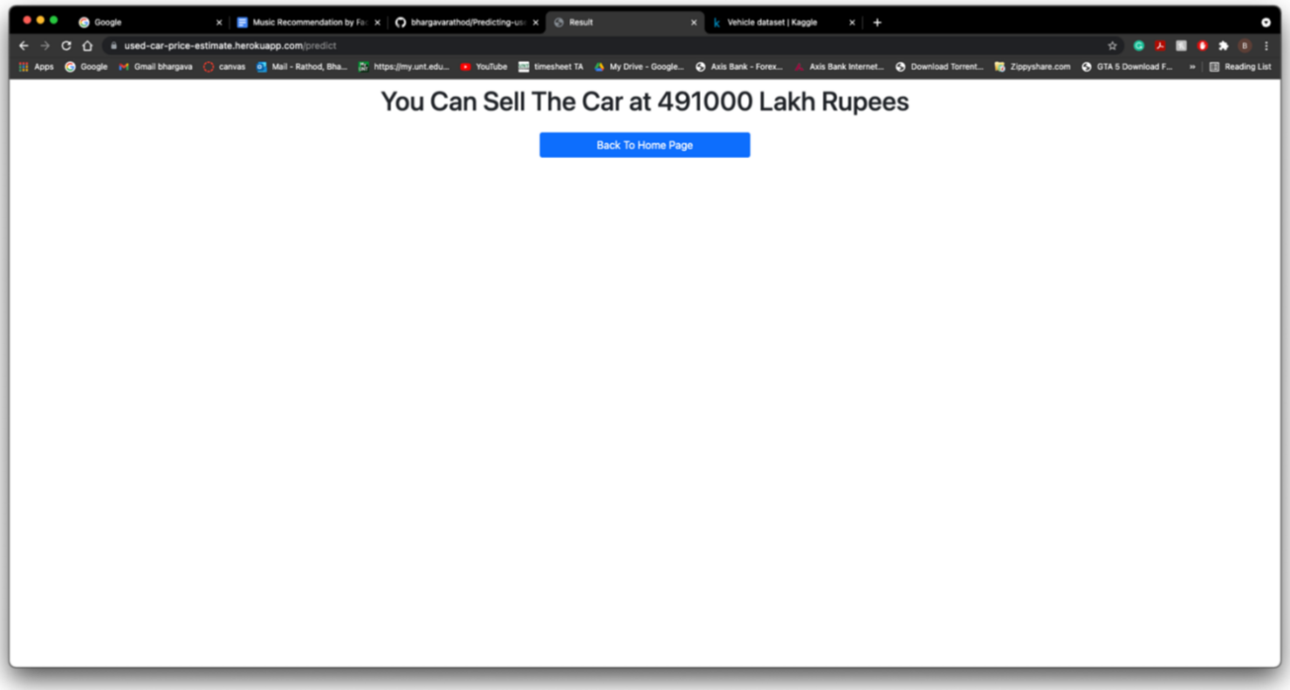
In this section we want to deploy our trained model into Heroku. The main reason for choosing it is because it is very simple, free of cost for small scale projects and, we can push the code which is present in the GitHub repository.

In the first place by using flask API, we have integrated the front-end application with the backend and then we have uploaded the whole code to the GitHub repository. After creation of the Heroku account, in just a few steps we had deployed our model into production.

4. Results & Analysis:

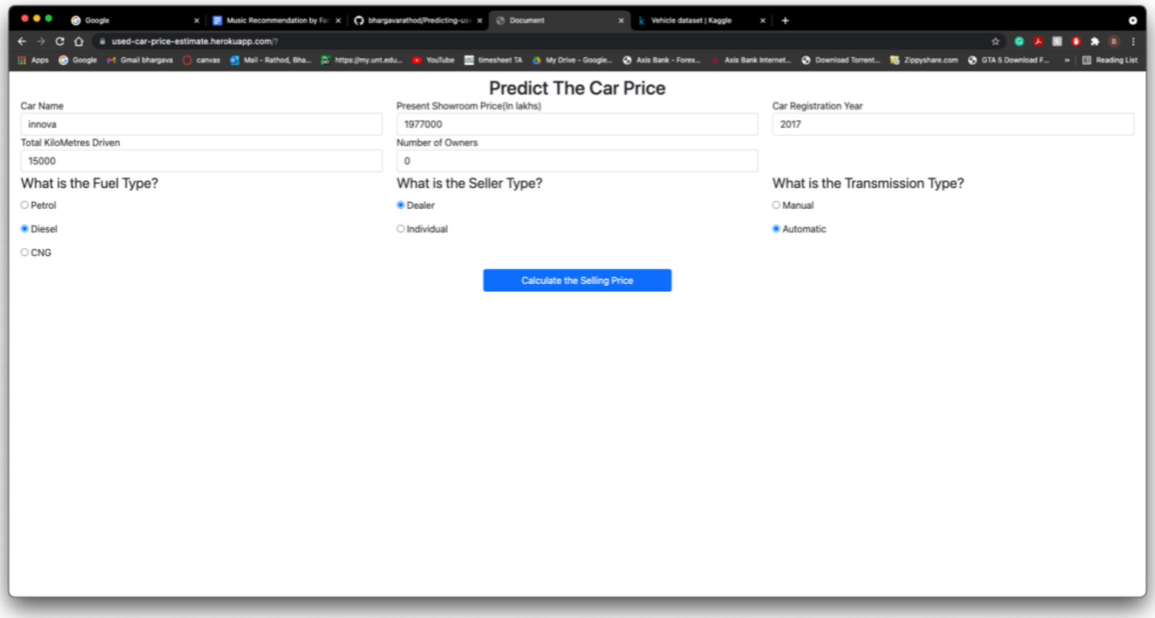
Here in the input fields, I have given the values based on the dataset values. You can check it is very close to the selling price column in the dataset. Hence it predicted correctly.

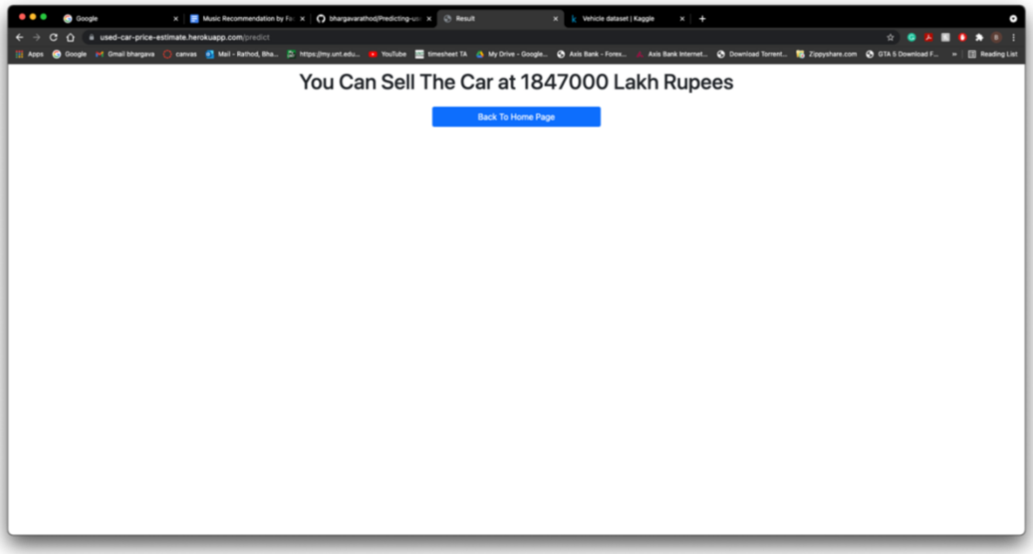




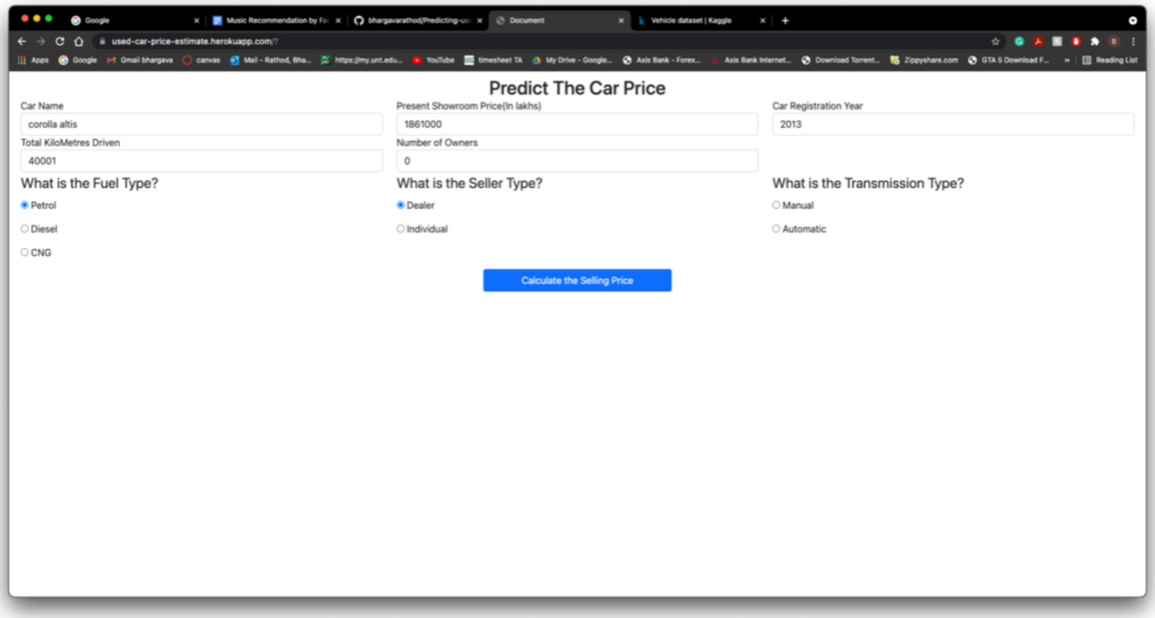
8	ertiga	2015	7.25	10.79	41678	Diesel	Dealer	Manual	0
9	ertiga	2016	7.75	10.79	43000	Diesel	Dealer	Manual	0
0	wagon r	2015	3.25	5.09	35500	CNG	Dealer	Manual	0
1	sx4	2010	2.65	7.98	41442	Petrol	Dealer	Manual	0
2	alto k10	2016	2.85	3.95	25000	Petrol	Dealer	Manual	0
3	ignis	2017	4.9	5.71	2400	Petrol	Dealer	Manual	0
4	sx4	2011	4.4	8.01	50000	Petrol	Dealer	Automatic	0
5	alto k10	2014	2.5	3.46	45280	Petrol	Dealer	Manual	0
6	wagon r	2013	2.9	4.41	56879	Petrol	Dealer	Manual	0
7	swift	2011	3	4.99	20000	Petrol	Dealer	Manual	0
8	swift	2013	4.15	5.87	55138	Petrol	Dealer	Manual	0
9	swift	2017	6	6.49	16200	Petrol	Individual	Manual	0

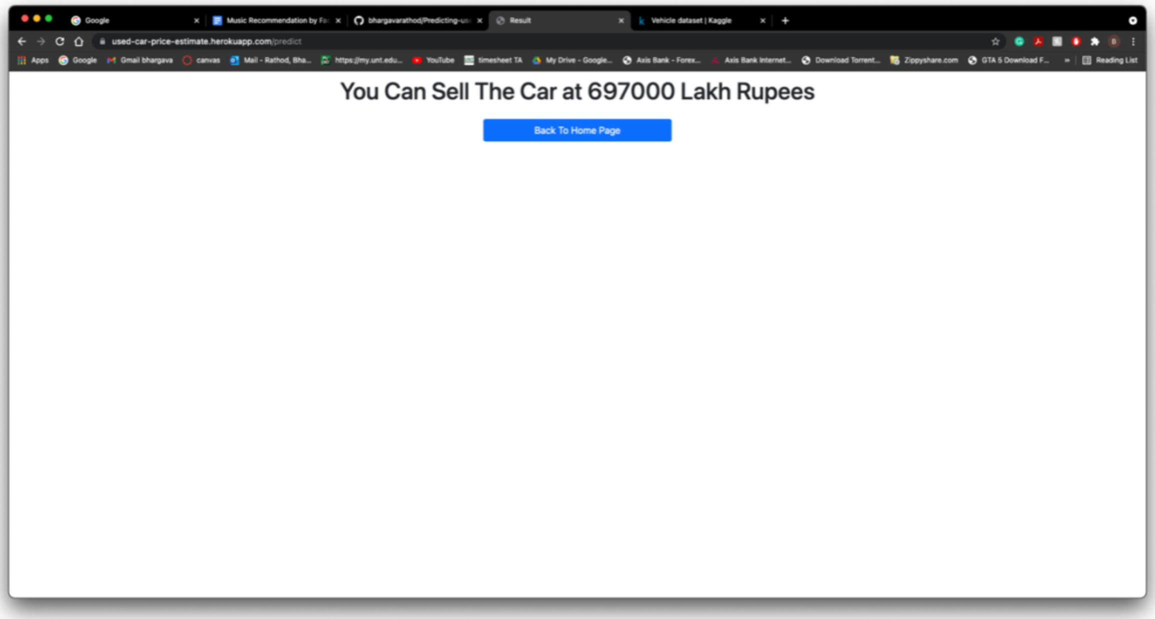
Where 4.9L (490000) is the selling price of the car name IGNIS.





omni	2012	1.25	2.69	50000	Petrol	Dealer	Manual	0
ciaz	2014	7.5	12.04	15000	Petrol	Dealer	Automatic	0
ritz	2013	2.65	4.89	64532	Petrol	Dealer	Manual	0
wagon r	2006	1.05	4.15	65000	Petrol	Dealer	Manual	0
ertiga	2015	5.8	7.71	25870	Petrol	Dealer	Manual	0
ciaz	2017	7.75	9.29	37000	Petrol	Dealer	Automatic	0
fortuner	2012	14.9	30.61	104707	Diesel	Dealer	Automatic	0
fortuner	2015	23	30.61	40000	Diesel	Dealer	Automatic	0
innova	2017	18	19.77	15000	Diesel	Dealer	Automatic	0
fortuner	2013	16	30.61	135000	Diesel	Individual	Automatic	0
innova	2005	2.75	10.21	90000	Petrol	Individual	Manual	0
corolla alti:	2009	3.6	15.04	70000	Petrol	Dealer	Automatic	0





fortuner	2015	23.5	35.96	47000	Diesel	Dealer	Automatic	0
fortuner	2017	33	36.23	6000	Diesel	Dealer	Automatic	0
etios liva	2014	4.75	6.95	45000	Diesel	Dealer	Manual	0
innova	2017	19.75	23.15	11000	Petrol	Dealer	Automatic	0
fortuner	2010	9.25	20.45	59000	Diesel	Dealer	Manual	0
corolla alti	2011	4.35	13.74	88000	Petrol	Dealer	Manual	0
corolla alti	2016	14.25	20.91	12000	Petrol	Dealer	Manual	0
etios liva	2014	3.95	6.76	71000	Diesel	Dealer	Manual	0
corolla alti	2011	4.5	12.48	45000	Diesel	Dealer	Manual	0
corolla alti	2013	7.45	18.61	56001	Petrol	Dealer	Manual	0
etios liva	2011	2.65	5.71	43000	Petrol	Dealer	Manual	0
etios cross	2014	4.9	8.93	83000	Diesel	Dealer	Manual	0

5. CODE

GitHub repository to access all the files:

https://github.com/bhargavarathod/Predicting-used-car-prices-using-machine-learning.git

Link to access the User interface to get predictions:

https://used-car-price-estimate.herokuapp.com/?