Used Cars Price Prediction

Venkata Manikanta Monic Kamisetty (A20446683)
Jesleen Sonia Pradeep Kamalesh (A20448891)
Venkata Sai Akshay Kishore Khanderao (A20458999)

1 Introduction

New Cars: Cars are financial investments apart from being a mode of transportation. Annually many new cars are being launched and the price range keeps increasing based on the technology and safety improvements. Customers who can afford and are attracted to new models buy them. It has lot of tangible advantages like safety, fuel efficiency, new car deals, technology and many more. However, a new car has its own disadvantages too. They are to be purchased only from a particular franchise and it includes a lot of tax which makes it more expensive for a lot of people. According to a research, the value of a car deprecates by 20 percent in the first year of purchase. A car loses its original value as year passes even if it is in the best condition.

Pre-Owned Cars: The affordability rate is more for used cars than new cars. The sales of used cars are directly proportional to the price of the new car. On purchase of used car, the customers do not face huge depreciation in price. The taxes are comparatively less, and it allows customer to buy their favorite cars as they are less expensive than the new cars. The used cars are a lottery for people who can't afford new cars.

Previous work: Used car price prediction is an interesting and necessary problem that would help many people from getting lured. The cars price is usually estimated based on the various distinct and relevant features which adds value to the car. Price prediction of used car was studied in various research and it was implemented using different machine learning algorithms to get better precision. The previous implementations included Random Forest, Artificial Neural Networks, Support Vector Machines and the results were compared and analyzed. In our project, we have implemented six algorithms like XGBoost, Gradient Boosting, Lasso Regressor, Ridge Regressor and Model Averaging. The price predicted by our model was better than the previous models.

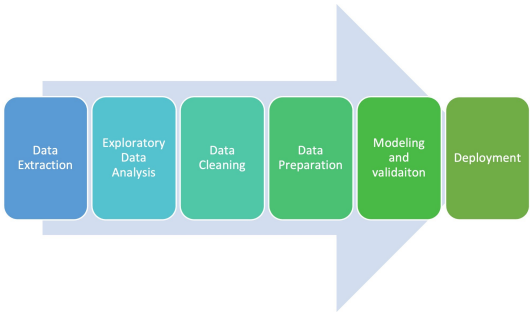
2 Problem Description

As the used cars sales increases, people are getting defrauded by unrealistic prices showcased by the vendors. There is a need for used car price prediction based on the worthiness of car using their features. There are a lot of websites that offer the service, but we are unsure if they are the best. Predicting the price of a used car is an important problem that would help customers from getting deceived by fake prices and it'll help dealers to offer quality service to the customers based on their requirements. The project aims to provide a better prediction model to customers for buying or selling a car using various machine learning techniques. The price of a used car is estimated based on relevant features

3 Data Description

The data used for this project (Used Car Dataset) is taken from Kaggle. The data contains vehicle listing from craigslist.org. Craigslist is the world's largest collection of used vehicles for sale. The Data consists of 26 columns and 458213 rows. A brief description of each column is as in Table 1.

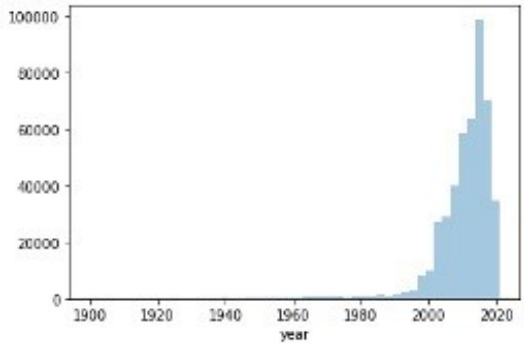
4 Methodology



5 EDA and Data Cleaning

5.1 Year

From the plot, we can state that most of the cars in the dataset are manufactured between the years 2000 and 2020.



id	Unique ID given to every car and is the primary key to the data set
url	URL from where the row data has been taken
region	Region where the vehicle is available
price	Price is given in US dollar and has not been adjusted for inflation
year	Year in which the car was manufactured
manufacturer	with 43 unique businesses engaged in the manufacture of automobiles
model	The exact model of the vehicle
condition	The condition of the car
cylinders	The number of cylinders in the car engine
fuel	There are five types of fuel
odometer	The distance the car has covered from the purchased date
title_status	The cars have 6 types of statuses
transmission	Transmission type automatic or semi-automatic
VIN	Vehicle identification number
drive	There are 3 types of drive transmissions
size	Size of the vehicle
type	There are 13 unique type values in this feature.
paint_color	Color of the vehicle
Image_url	URL of the image posted
state	The states in the United States are represented in a short form
lat, long	When both features are combined, they give the location of where the car is being sold at
posting_date	Posting date of the vehicle for resale

Table 1. Feature Description

It is better to keep the original values given by the dataset any time than replacing the NaN values with mean or median. For the sake of this, we found out the year with the fewer missing values (1050). So, we dropped those training examples. Now we have a feature with the original data.

Detecting Outliers: From the boxplot, most of the outliers are before 1995. So, it is better to remove outliers as we know they will highly affect the model's prediction.

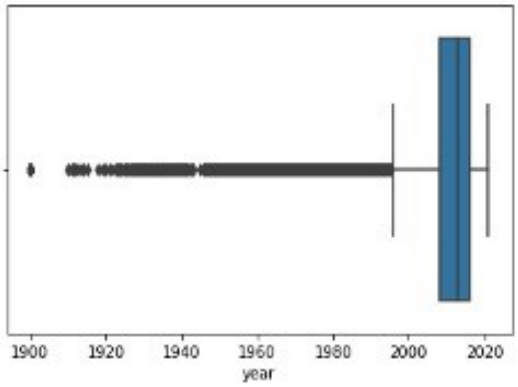


Figure 1. Year

5.2 Manufacturer

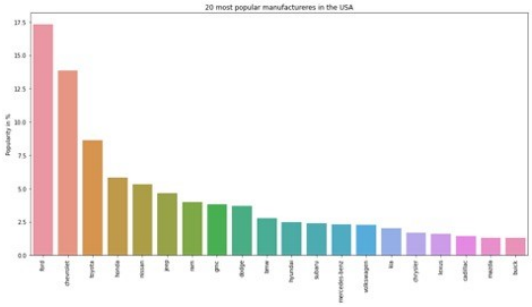


Figure 2. Manufacturer

We have taken the top 20 popular manufacturers in the USA to plot the popularity of the vehicle brands. From the above bar plot, we can conclude that ford and Chevrolet are the most popular brands.

We observed a slight increase in missing values (14655) for the manufacturer compared to the year. As the manufacturer plays important role in predicting the price of a vehicle it's better not to fill the missing values by traditional filling mechanisms (filling with mode, forward fill, etc.) So, we decided to replace the null values with "unknown". The reason for this is tree-based algorithms (bagging, random forest) are good at detecting null values

by this small change. This may increase the prediction accuracy by tree algorithms.

5.3 Model

We have taken the top 20 popular models in the USA to plot the popularity of the vehicle models. From the above bar plot, we can conclude that f-150 and Silverado 1500 are the most popular models.

For the same reason as the manufacturer, we filled the null values with “unknown”.

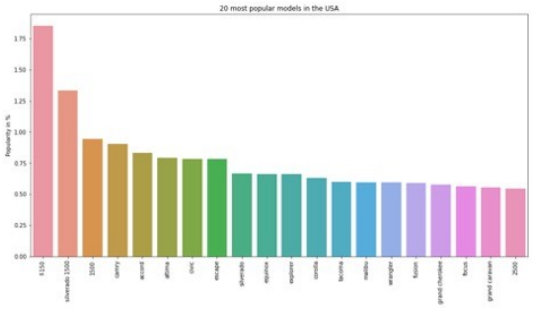


Figure 3. Model

5.4 Odometer

From the boxplot, we can observe two outliers will highly affect the prediction so, it is safe to remove those outliers.

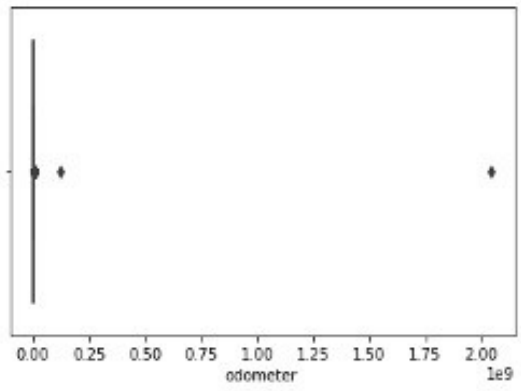


Figure 4. Odometer with Outliers

We considered the odometer values that are less than 250000 and greater than 10. Because of this reason all the null values have also been removed.

5.5 Condition

Based on the condition plot, a significant amount of the vehicles in the data set are in good condition followed by vehicles in excellent condition. But the fewer number of vehicles in salvage and new makes the price prediction of this category a little off the chart.

This problem can be avoided by a little common knowledge on people as they don’t tend to buy salvage vehicles. To compensate for the fewer number of new vehicles we added the ‘new’ category to the ‘like new’ category.

There are a significant number of null values in condition (192940). There is a high chance of losing a lot of information

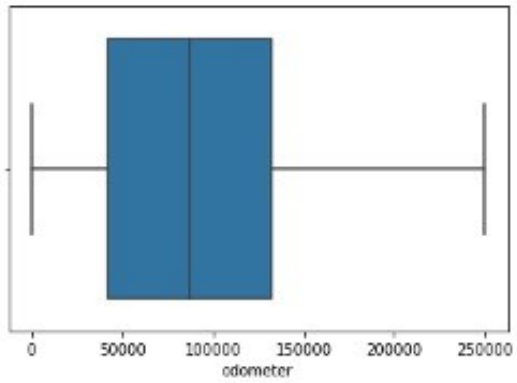


Figure 5. Odometer without outliers

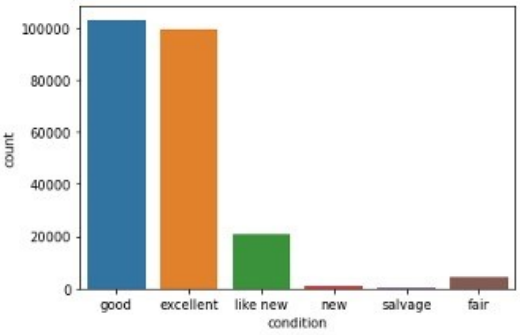


Figure 6. Condition

about the price of a vehicle if the null values are dropped or replaced with Mode (since it is a categorical feature).

So, we did a domain search and found out that the condition of the car depends on the odometer value. Based on the value of the odometer the values of condition are either replaced or null values are filled.

Approach: We calculated the mean values of the odometer for each category of condition. Then, a condition is set for the odometer values based on the mean values of each category of condition. Then they are replaced with the appropriate category of condition.

5.6 Cylinders

Based on the cylinder plot, most of the vehicles that are coming to resale are the vehicles with 6 cylinders followed by 4 cylinders.

As there are a significant number of null values (128249) in the cylinders feature, it is not a good approach to fill the null values with mode. So, it is better to replace them with 'unknown' as the tree algorithms will detect null values based on this value.

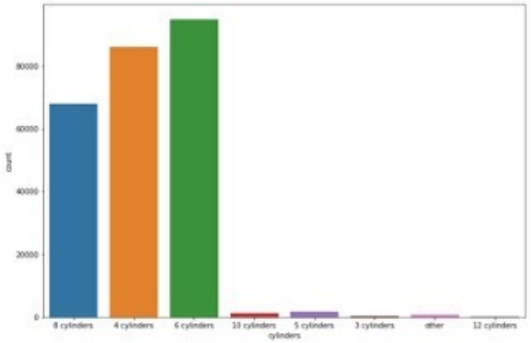


Figure 7. Cylinders

5.7 Fuel

Based on the fuel plot, most of the vehicles in the USA run with gas followed by diesel.

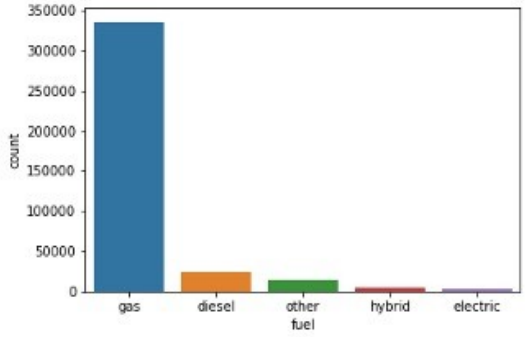


Figure 8. Fuel

As there are very few null values in the fuel feature (2172), replacing those with mode doesn't affect the prediction much.

5.8 title_status

Based on title_status, most of the vehicles in the USA are clean.

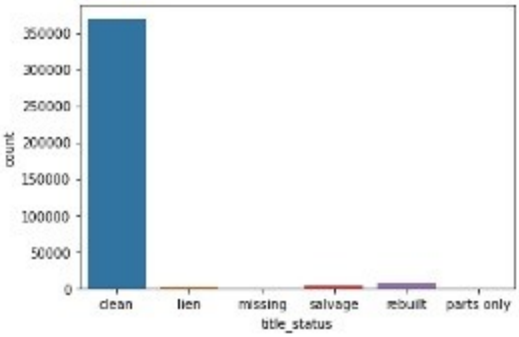


Figure 9. title status

As there are very few null values in the title status feature (1582), we replaced the null values with the mode of the feature.

5.9 Type

This feature gives us information about the type of the car. Ex: sedan, SUV, pickup etc.

From the histogram, we can see that from all the cars the sedans are the most popular cars followed by SUVs and trucks and sedans having the highest average resale price. The type of car feature has a lot of missing values and they are filled by checking the description feature for the car type and the extracted value is assigned to the Type feature. At last, few data points which still had null values were discarded.

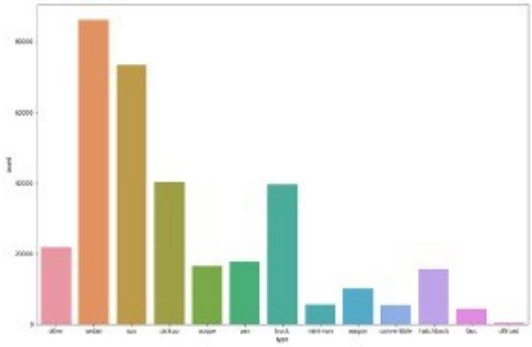


Figure 10. Type

5.10 Drive

This feature gives us information about the Drive of the car it is classified into 3 types i.e., rear-wheel drive, front-wheel drive 4-wheel drive. Four-wheel drives are most popular with the highest sales and highest average price followed by front-wheel drive. The rear wheel drive Is the least popular having the least average price. The Drive feature has a lot of missing values and they are filled

by checking the description feature for the car drive and the extracted value is assigned to the Drive feature. At last, few data points which still had null values were discarded.

5.11 Size

The cars are divided into classes based on the sizes they are broadly classified into 4 groups full-size, mid-size, compact, sub-compact. Below is the distribution of the cars. The Size feature has a lot of missing values and they are filled by checking the description feature for the car driver and the extracted value is assigned to the Drive feature. At last, few data points which still had null values were discarded.

With the full-size being the most popular both in terms of several sales and high average resale value closely followed by mid-size in the USA and subcompact being the least preferred segment int the used cars.

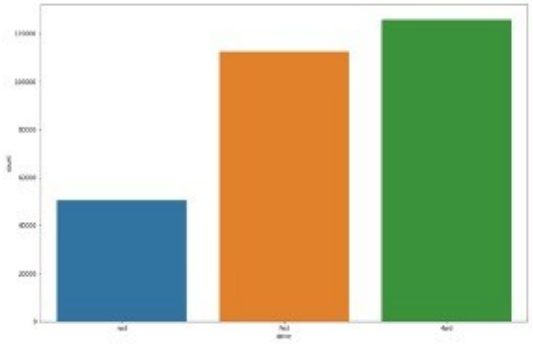


Figure 11. Drive

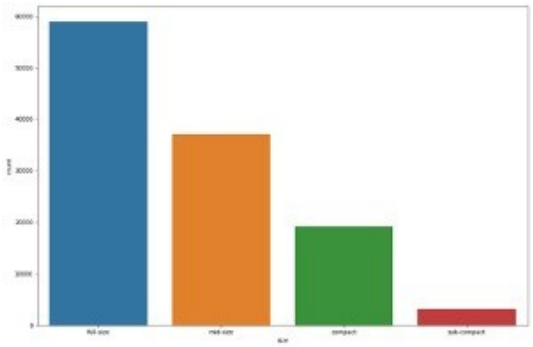


Figure 12. Size

5.12 Transmission

The cars are divided into classes based on the sizes they are broadly classified into 2 groups Automatic and Manual. With the automatic being the most popular type of transmission. The transmission feature had few missing values as the data is skewed towards the automatic value, so the few missing values are also filled using the Mode values.

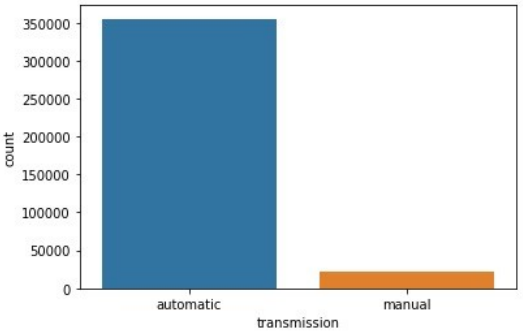


Figure 13. Transmission

6 Data Preparation

Since raw data is not suitable for analysis and predictions, we need to prepare the data for modeling algorithms. Since machine learning modelling algorithms do not accept string values, we need to encode categorical variables. We have employed Ordinal Encoding for encoding ordinal variables and One-hot Encoding for nominal variables. In addition to this, we performed normalization to several attributes to bring it to one scale.

	year	cylinders	odometer	drive	manufacturer_acura	manufacturer_alfa-romeo	manufacturer_aston-martin	manufacturer_audi	manufacturer_bmw	manufacture
220052	0.958333	0.714286	0.040324	2	0	0	0	0	0	
10665	0.333333	0.428571	0.273475	2	0	0	0	0	0	
102772	0.750000	0.428571	0.308018	2	0	0	0	0	0	
90016	0.791667	0.428571	0.528394	4	0	0	0	0	0	
294161	0.708333	1.000000	0.386033	4	0	0	0	0	0	

Figure 14. Data Frame after Encoding

7 Modeling

7.1 Gradient Boosting

It is one of the boosting algorithms known as additive model. The name is since the best possible next model is derived by combining the previous models. The intuition is that it reduces the overall prediction error. The target outcome for each iteration in the data depends on how much changing that iteration's prediction impacts the overall prediction error:

- A small change in the prediction for an iteration causes a large drop in error, then next target outcome of the iteration is a high value. Predictions from the new model that are close to its targets will reduce the error.
- A small change in the prediction for an iteration causes no change in error, then next target outcome of the case is zero. Changing this prediction does not decrease the error.

The name Gradient Boosting arises because the target outcomes for each iteration are set based on the gradient of the error (residual) with respect to prediction. Each new model takes a step in the direction that minimizes prediction error, in the space of possible predictions for each training iteration.

The Model

```
#GradientBoosting model
gb_model = GradientBoostingRegressor(
    n_estimators = 500,
    max_depth = 8,
    learning_rate = 0.3)
```

Figure 15. Gradient Boost Model

Metrics

- RMSE:
  Train set - 0.17601133539453415
  Test set - 0.24086182205288478
- R-Square:
  Train set - 0.9419211760677981
  Test set - 0.8905386759116185

Feature Importance

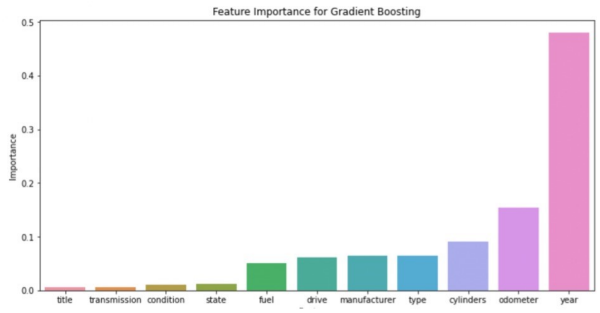


Figure 16. Gradient Boost Model Feature Importance

7.2 XGBoost

It is like Gradient Boosting including a second order approximation of objective function. It is one of the ensemble tree methods that apply the principle of boosting weak learners using gradient descent architecture. However, XGBoost improves upon the base GBM framework through systems optimization and algorithmic enhancements. How XGBoost optimizes the Gradient Boosting

- Parallelized tree building
- Tree pruning using depth-first approach
- Cache awareness and out-of-core computing
- Regularization for avoiding overfitting
- Efficient handling of missing data
- In-built cross-validation capability

The Model

```
#XGBoost Model
regressor = xgb.XGBRegressor(
    n_estimators=100,
    reg_lambda=1,
    gamma=0,
    max_depth=5
)
```

Figure 17. XGBoost

Metrics

- RMSE:
  Train set - 0.2754892285733874
  Test set - 0.28333842752516675
- R-Square:
  Train set - 0.8577193686904432
  Test set - 0.8485268276098412

Feature Importance

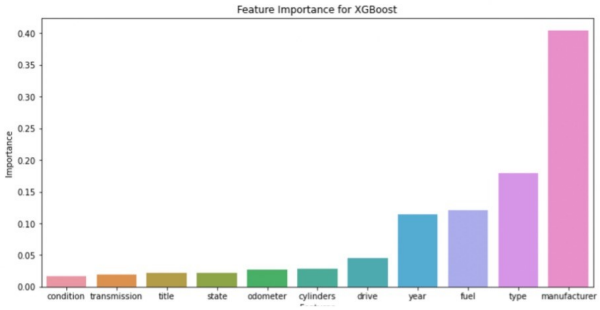


Figure 18. XGBoost Feature Importance

7.3 Random Forest

Random Forest Regression is a supervised learning algorithm that uses ensemble learning method for regression. It combines the predictions from multiple decision trees to make an accurate prediction. The trees run in parallel with no interaction among them. The algorithm operates by constructing several decision trees during training and outputting the average of the value as the prediction. It performs great on many problems including features with non-linear relationship. The main

drawback is there is no interpretability, overfitting may easily occur.

The Model

```
#Random Forest Model
random_forest = RandomForestRegressor(
    n_estimators=20,
    random_state=0,
    n_jobs=-1)
```

Figure 19. Random Forest

Metrics

- RMSE:
  Train set - 0.09451153577574196
  Test set - 0.23749006346416754
- R-Square:
  Train set - 0.9832541892672594
  Test set - 0.8935818635080255

Feature Importance

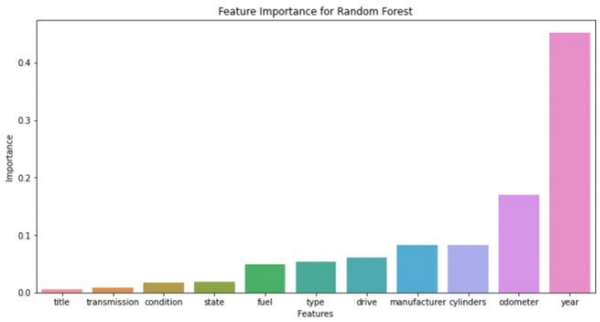


Figure 20. Random Forest Feature Importance

7.4 Ridge Regression

Ridge Regression is for analyzing multiple regression data that suffer from multi collinearity. When multi collinearity occurs, least squares are unbiased, variances are large so predicted value is far from true value. Adding a degree of bias to the regression estimate reduces the standard errors. This is the idea of Ridge Regression. The net effect is to give estimate that is more reliable.

The Model

```
#Ridge Regression Model
ridgeRegressor=Ridge(alpha=0.415545)
```

Figure 21. Ridge Regression Model

Metrics

- RMSE:
  Train set - 0 .15301317558731573
  Test set - 0.23054624123627698
- R-Square:
  Train set - 0.9561071081879771
  Test set - 0.8997138741242393

7.5 Lasso Regression

LASSO stands for Least Absolute Shrinkage and Selection Operator. It is a type of linear regression that uses shrinkage. Shrinkage is where the data values are shrunk towards a central point. Lasso procedures encourage simple, sparse models like models with less parameters. Like Ridge Regression, Lasso regression is well suited for models showing multi collinearity.

The Model

```
#Lasso Regression model
lassoRegressor=Lasso(alpha= 0.007761033)
```

Figure 22. Lasso Regression Model

Metrics

- RMSE:
  Train set - 0.39881175668101554
  Test set - 0.40101499468558394
- R-Square:
  Train set - 0.7018241659711835
  Test set - 0.6965787546457696

8 Model Evaluation

Table 2 shows the results of all machine learning model's metrics

Algorithm	RMSE (Train)	R-square (Train)	RMSE (Test)	R-square (Test)
Gradient Boosting	0.17	0.94	0.24	0.89
XGBoost	0.27	0.85	0.28	0.84
Random Forest	0.009	0.98	0.23	0.89
Ridge Regression	0.15	0.95	0.23	0.89
Lasso Regression	0.39	0.70	0.40	0.69
Average Regressor	0.15	0.95	0.23	0.89

Table 2. Comparison of Metrics

Visualizations of RMSE for train and test datasets:

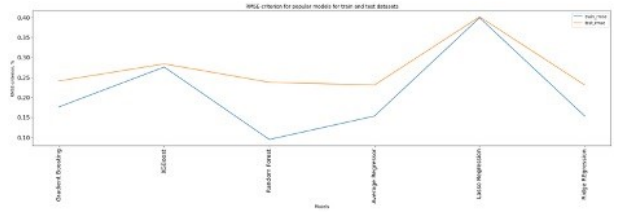


Figure 23. RMSE Comparison of Algorithms for Train and Test

Visualizations of R2 for train and test datasets:

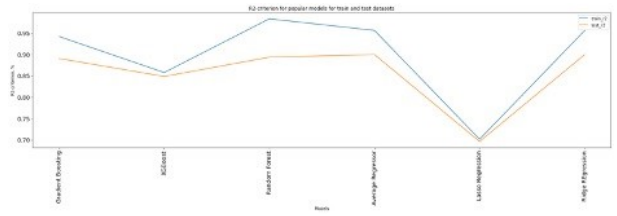


Figure 24. R-Square Comparison of Algorithms for Train and Test

9 Deployment

Deployment facilitates the integration of the model with production environments to make practical decisions. Random forest model has the best RMSE and R2 scores out of all the other models. The model is deployed into an application.

9.1 Using Flask

The Flask framework has been used in order to create the web API. Flask is a micro framework that helps us create web services with Python. For simple requirements, Flask lets you hit the ground running.

There are many Python web frameworks being utilized, but Flask is the best among all of them as it provides users with many libraries, tools, and modules in order to build web-applications.

Flask is relatively easier to use, easier to extend, and has high flexibility along with good HTTP request and handling.

9.2 Serialization / Deserialization

In order to save the python objects, we have used the module named “pickle”. Pickle is a module in Python that helps in serializing and de-serializing Python objects. It converts any object into a character stream. Pickle makes the object as a form of a pre-compiled library, which we can later import into our deployment code.

9.3 The Application

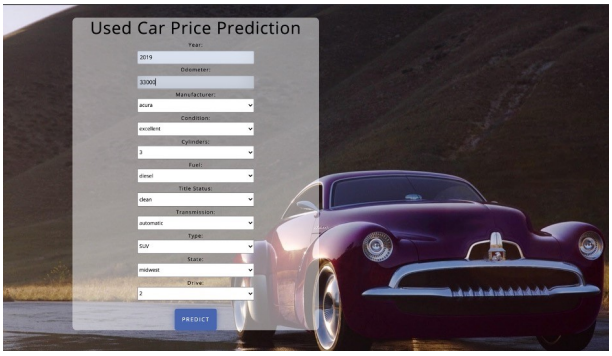


Figure 25. The Application

10 Future Work

There are several socio-economic factors such as GDP per capita, market potential which may affect the pricing of the car. These factors can be used to better estimate the price of a pre-owned car. Adding the data from different websites likes cars.com, truecars.com and many other helps us get a broad spectrum and can practically predict the price of any car. We can also give user-friendly recommendations to customer by considering weather and accident datasets. For example, the model can suggest a SUV car for a customer who lives in an area where it snows heavily.

11 References

[1]https://www.temjournal.com/content/81/TEMJournalFebruary2019_113_118.pdf
[2] https://arxiv.org/pdf/1711.06970
[3]https://www.kaggle.com/austinreese/craigslist-carstrucks-data