CSE 4/587 B DATA INTENSIVE COMPUTING

PROJECT REPORT

USED CAR PRICE PREDICTION

Charan Kumar Nara - 50545001
Gokul P - 50510470
Kusha Kumar D - 50532663

December 09, 2023

Phase 1

Problem Statement:

Background:
The automotive industry is one of the key players in the global economy. The market for used cars is growing substantially due to consumers' economic choices. So, the demand for used cars is beginning to grow rapidly in the market. As new cars start depreciating in value soon after they leave the dealership, used cars need to be sold or bought at a deprecated price value. But determining the price of a used car depends on various factors (age, type, size, condition, manufacturer, etc) and it's not straightforward.

Objectives:
Since the pricing of used cars has complexities, our objective is to develop an approach to predict the price of used cars based on relevant features.

Significance:
Due to the growing market for used cars, accurate pricing can benefit both buyers and sellers to have fair trade and prevent potential losses.

Potential contribution and cruciality:
Data-Driven Decision-Making: By developing a model that predicts used car prices based on observed data, we move towards data-driven decision-making which reflects the true value of a vehicle.

Enhanced Market Efficiency: An accurate prediction model can lead to more transparent and consistent pricing in the used car market leading to more efficient market transactions.

Consumer Protection: This model serves as a benchmark for buyers to evaluate the fairness of the price of a used car. This will prevent them from overpaying and ensure good value for their money.

Cruciality of the contribution:

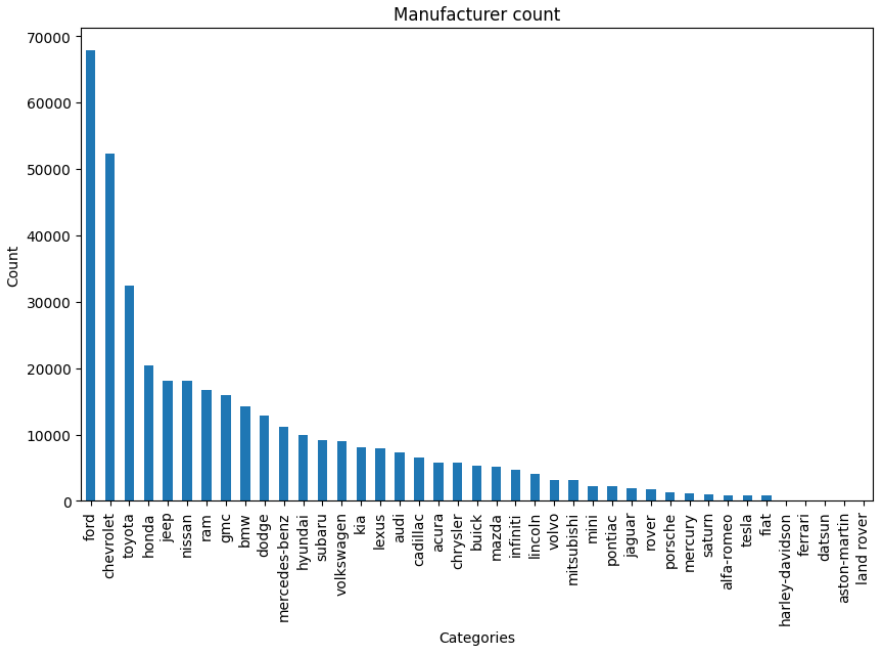
- This model will create transparency in the pricing of used cars, which benefits both buyers and sellers and increases the number of transactions leading to increased economic activity.
- It builds trust and confidence among market participants.
- With this robust model, we encourage the reuse of vehicles reducing the demand for new car production and its associated environmental impact

Data Sources:

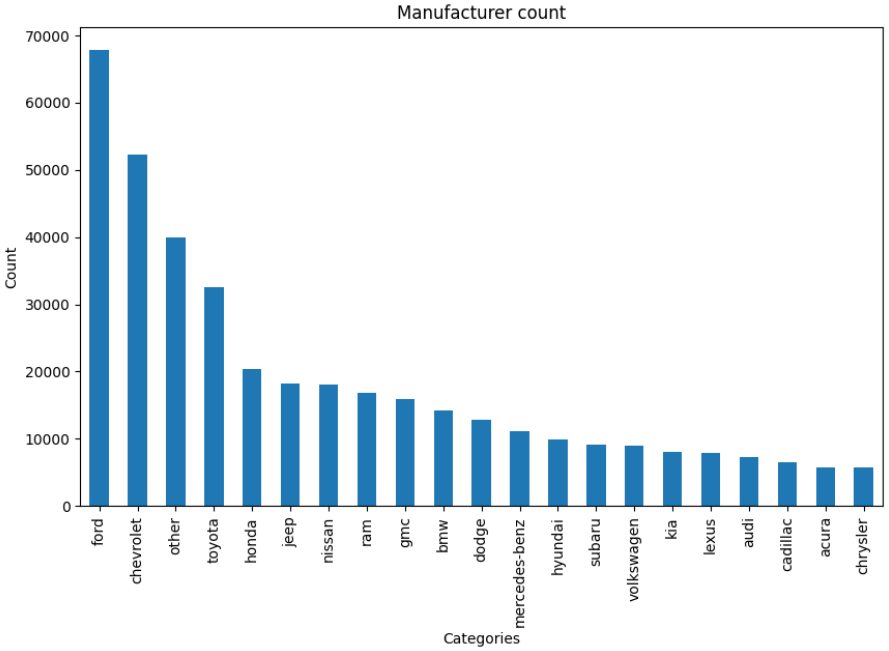
We are making use of the "used cars price prediction" dataset available on Kaggle
https://www.kaggle.com/datasets/austinreese/craigslist-carstrucks-data

Data processing and cleaning

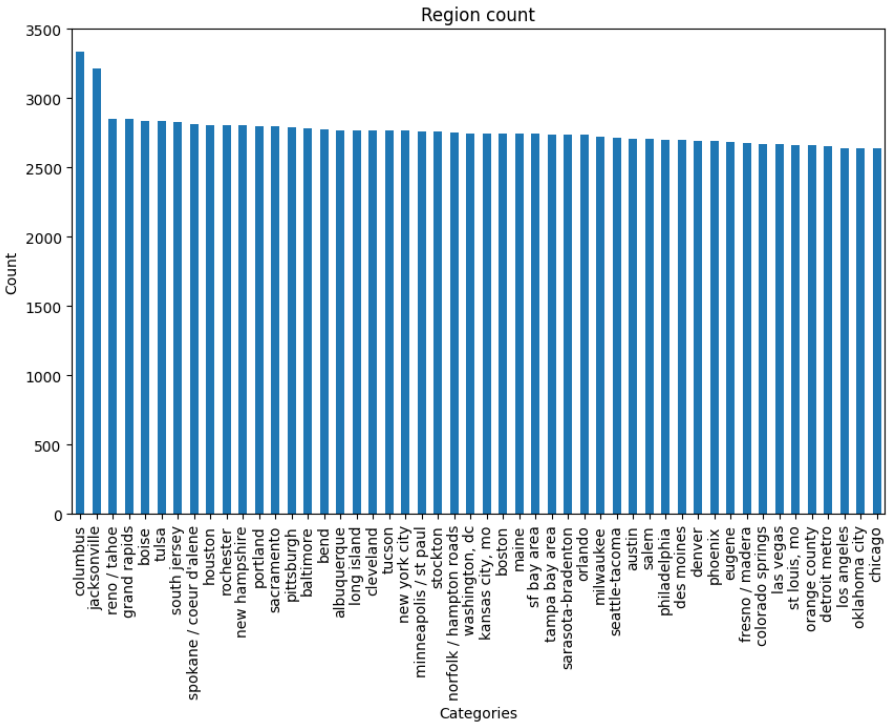
1. **Removing null values**
    a. Sparse and dense nulls: On checking we found that some columns have very large numbers of null values and some have very small numbers of null values. We decided to drop the rows for columns with a very small number of null values and for others, we replaced null values with a new string “uncharted”.
    b. Empty columns: We found that the county column doesn't have any data so decided to drop that column.
2. **Feature Pruning**
    a. Dropped irrelevant Features: Columns like id, URL, region_url, image_url, lat, and long do not provide any useful data for car price prediction. So, we dropped those columns.
3. **Duplicate data handling**
    a. On checking on duplicates in our data set we found that a total of 9 rows are duplicate rows. We removed those duplicate rows.
4. **Categorical data standardization**: Conversion of categorical data to a single case to maintain uniformity.
    a. After converting all the textual data in the dataset to lowercase we then checked unique values in all the categorical data type columns.
    b. All the categorical data type columns have correct data without any redundancy because of typographical errors.
5. **Data imbalance handling in Categorical data type columns.**
    a. Manufacturer distribution: The plot of the Distribution of vehicles from different manufacturers shows some serious insights. Some manufacturers have very little presence in the market compared to some other manufacturers



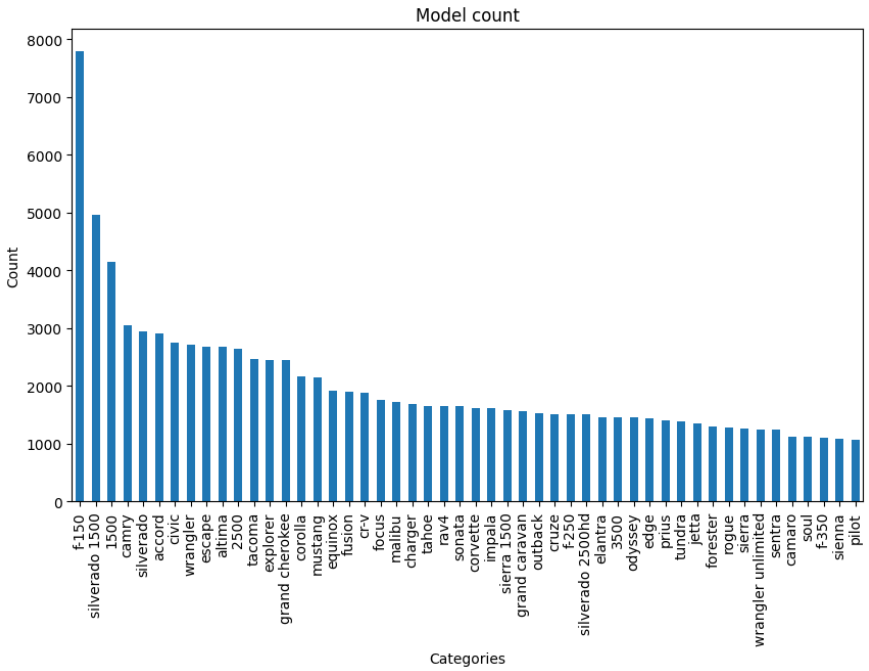
    b. We decided to go with only the first top 20 manufacturers and consider the rest as a single category named “other”. The distribution plot after doing that came out like below



c. Region feature handling: In a similar way, we processed the regions column by taking only the top 50 regions and considering the rest of the regions as “other “. The resulting graph is shown below.



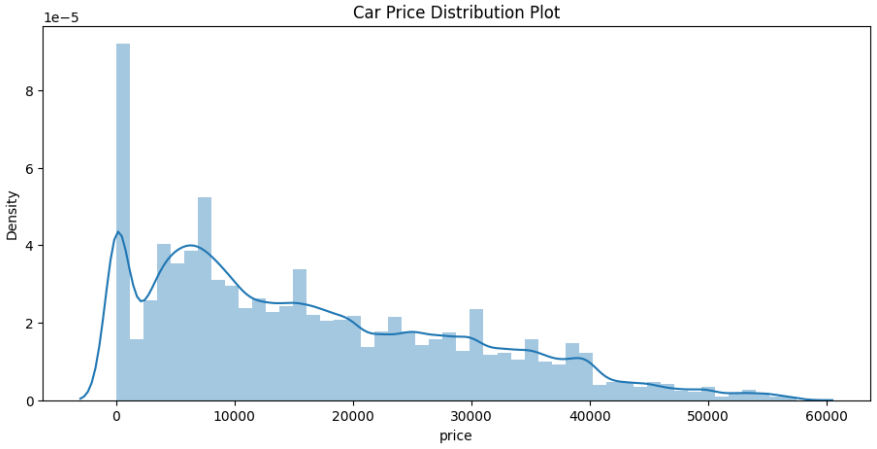
d. Model feature handling: For the model column, we did the same processing. We took the top 50 models and considered the rest as a single model “other”. The graph below shows the distribution of the models excluding the “other” model.



e.

6. Numerical data outlier management.

a. Price distribution: On checking the price distribution we found that the price column has outliers and the major chunk of the data is below 20,000$. We decided to move forward by removing the outliers that fall below the 25th percentile and above the 75th percentile. After doing this we got a better price distribution graph.



b. Odometer reading: We applied the same approach to the odometer readings. From which we have identified outliers and decided to remove rows with data entries at extreme ends.

c. Vehicle age: The year column had entries that were associated with very old vehicle rows. Since the current market price prediction won't be affected by such old vehicle data, we decided to strip them off of the dataset. We removed data that is very old which is the first quarter quantile.

7. Correcting feature data types

a. The year and odometer had a datatype as float. This doesn't make any sense. So, we converted the types of these features to int.

8. Feature engineering

a. Vehicle age column: Instead of the year column, we thought it would be better to have another column named the age of the car. The age of the car made more sense than just a year since the year can be anything like a manufacturing or selling year. So in order to avoid ambiguity we created a new column from the year column and named it Vehicle Age. we subtracted each row entry of the year from the current year and recorded it as car age.
b. Posted date transformation: A similar processing was done for the posted date column. The entries in the posted date were in a little difficult format which included both strings and numbers. We converted that data into a number of months format. We converted the current data into a date object and then calculated the month count from the current date till the entry date. We named this new column "Posted ago".

9. Dropping of duplicates of VIN column.

a. VIN uniqueness: Since the VIN is a unique ID for each car it doesn't make any sense to have multiple entries with the same VIN and different other features. On checking further we found that the rows with the same VIN had almost all other features also the same. So it is found that these are duplicates of the same vehicle with some data entry mistakes. We decided to keep only one of the such entries for each VIN.
b. From the above step, we now have removed duplicates with respect to the VIN column. Hence, we can drop the VIN column as well.

10. String to int conversion in cylinder column.

a. The data entries in the cylinder feature were found to be easy to convert to numerical for better calculation. We converted 1 cylinder to a single integer 1 and 2 cylinders to 2 and so on. Since this column had entries like "uncharted" and "other" that are unknown data, we replaced them with zero.

11. Sentiment analysis on Description column.

a. Description column might give us some insights about the overall condition of the vehicle.
b. Therefore, we performed sentiment analysis on this feature to determine the vehicle's sentiment, whether it is a positive or negative description about the vehicle.

12. Encoding categorical columns into numerical.

a. Created dummy numerical features from the existing categorical features, namely region, manufacturer, model, condition, cylinders, fuel, title_status, transmission, drive, size, type, paint_color, state, and sentiment, into numerical features, making it suitable for training with machine learning model.

13. Scaling Features.

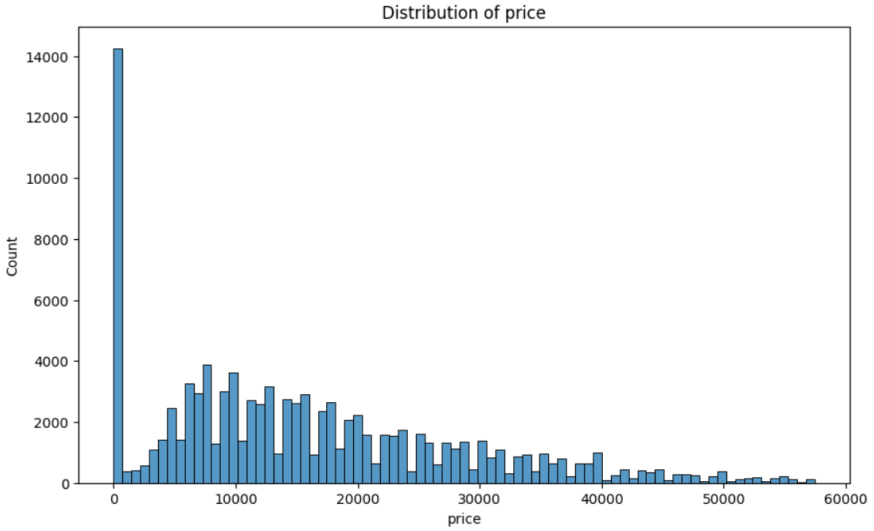
a. Scaled the independent features using the StandardScaler class available in the scikit learn module. This scaler will transform our data into a uniform scale, where the mean is 0 and the standard deviation is 1.

Exploratory Data Analysis:

We have performed 12 different exploratory data analyses and built relationships between the features.

Below are the visualizations of our analysis and insights from the analysis.

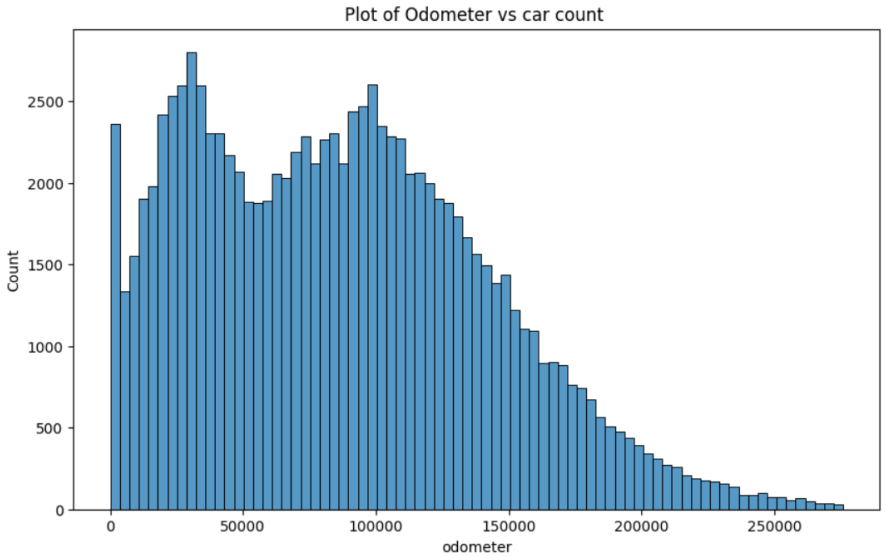
1. Distribution of the price of the vehicles



Insights:

- The majority of vehicles are priced below $30,000, with a peak around $1,000 to $5,000.
- Only a few vehicles are expensive, priced above $30000 up to $60000.

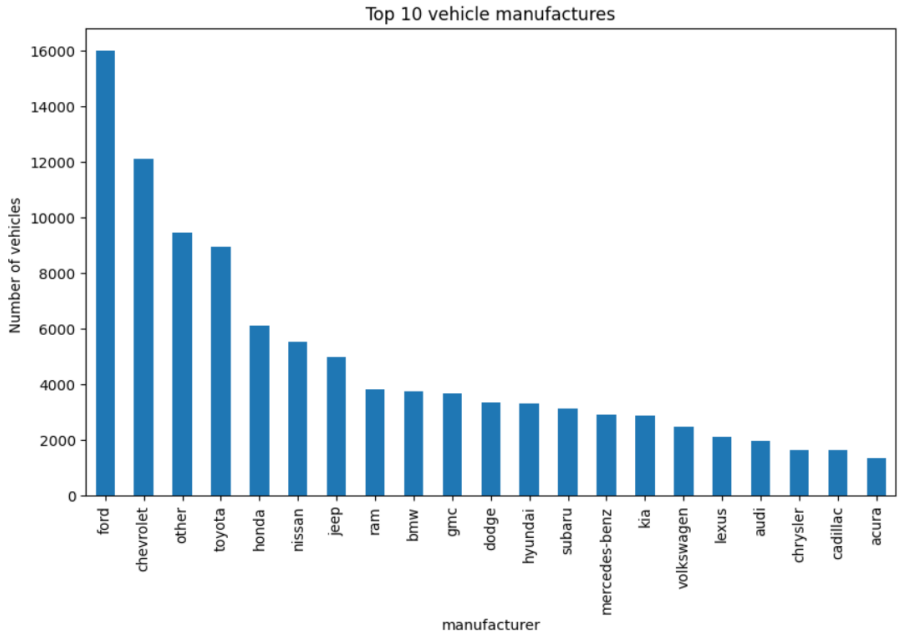
2. Odometer vs Car count



Insights:

- Most vehicles have odometer readings below 200,000 miles.
- There's a peak around 20,000 to 40,000 miles, indicating many vehicles with relatively low mileage.

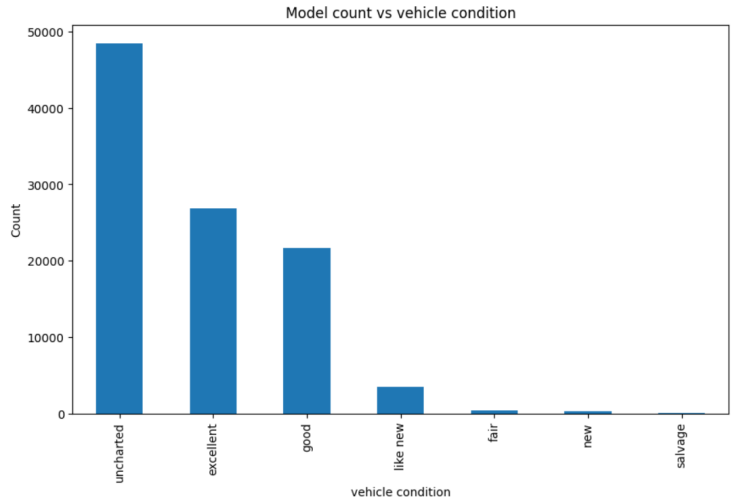
3. Top Manufacturers



Insights:

- Ford and Chevrolet are the top two manufacturers with the highest number of listings.
- Other manufacturers like Toyota, Honda, and Nissan also have a significant number of listings.

4. Model count vs Condition



Insights:

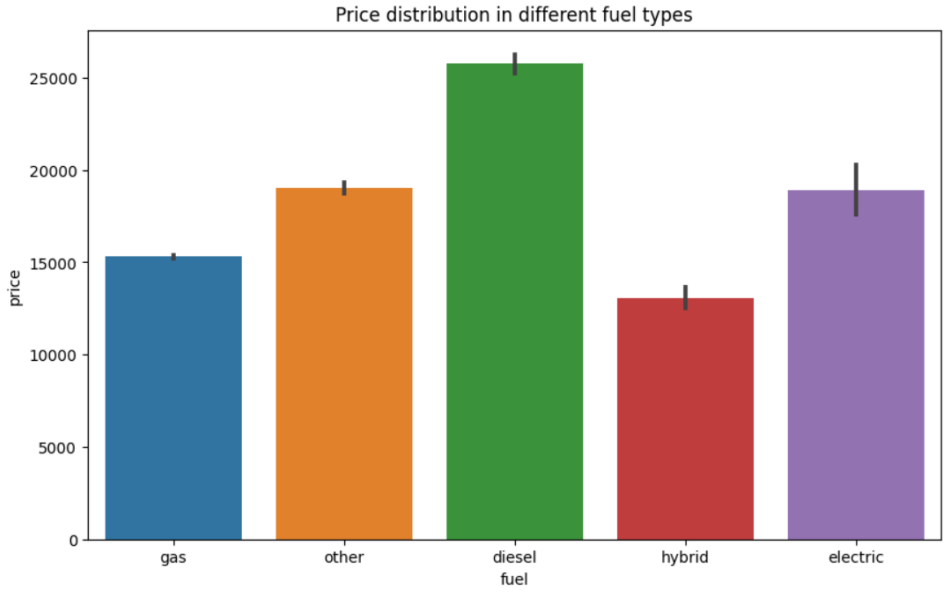
- A significant number of vehicles have an "uncharted" condition. Most vehicles are in "excellent" or "good" condition.
- Very few vehicles are in "like new", "fair", "new", or "salvage" condition.

5. Price distribution with respect to the age of the vehicle

Insights:

- Age has an inverse relation with price. As age increases, we can see that price is decreasing.
- Price is high for vehicles with age between 3 to 6 years. Vehicles with an age less than 3 years surprisingly have less price.

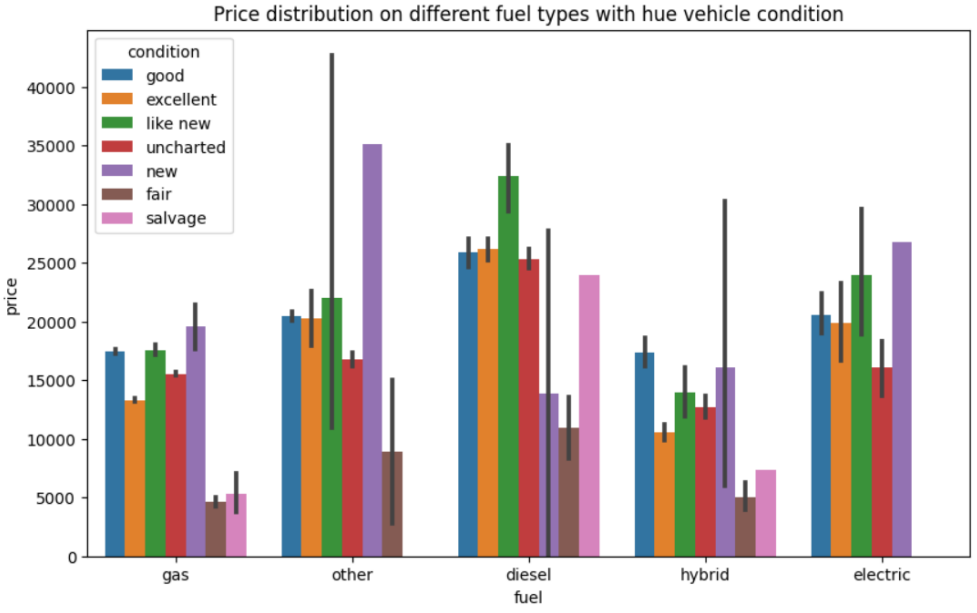
6. Price distribution across fuel types



Insights:

- Vehicles with diesel fuel type are costlier than other fuel type vehicles.
- Vehicles with hybrid fuel types are cheaper than other fuel-type vehicles.

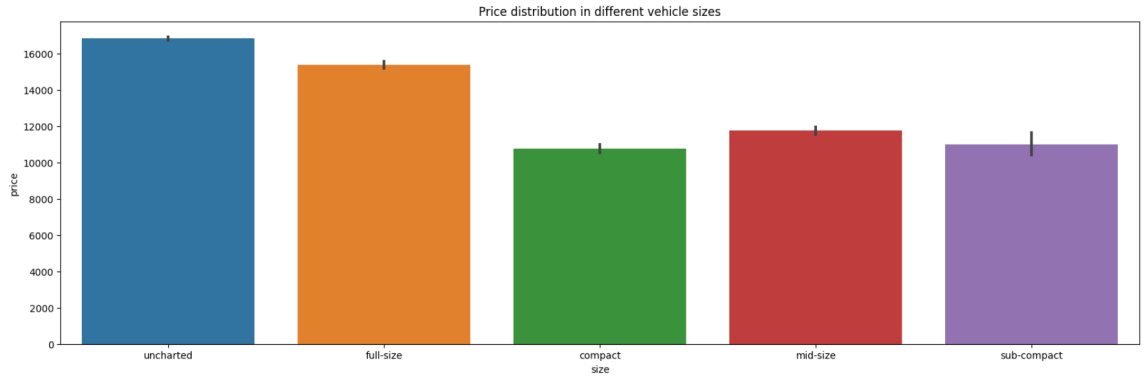
7. Price distribution based on different fuel types and the vehicle condition



Insights:

- Across different types of fuel, vehicles that are in ‘fair’ condition have very low prices.
- The Price of vehicles in ‘new’ condition has a lower price for Diesel fuel type compared to other fuel types.
- ‘Like new’ condition vehicles have good demand across all the fuel types. Surprisingly the salvage condition type is showing higher price than fair condition typed vehicles.

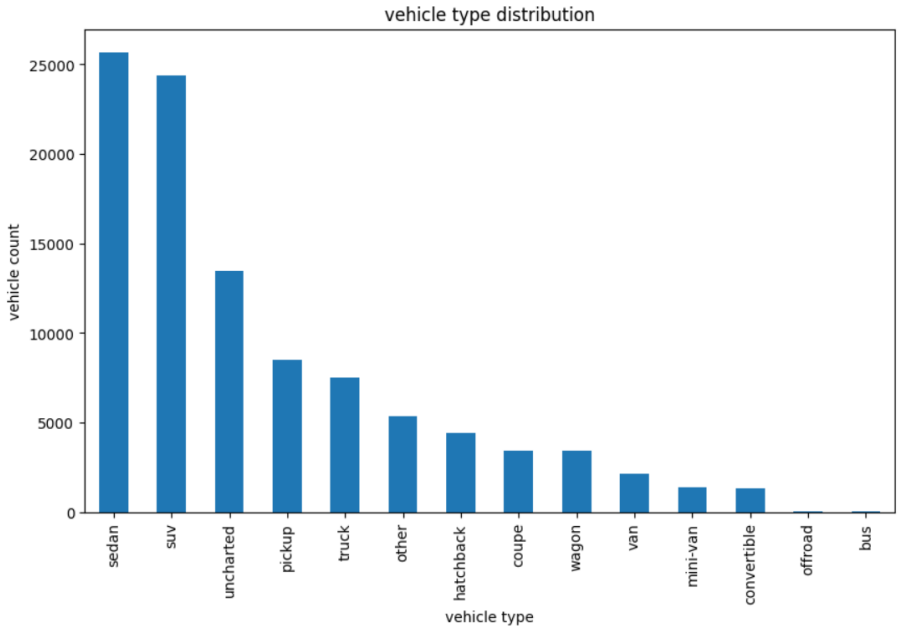
8. Price distribution across different vehicle sizes



Insights:

- In known types of vehicle sizes, full-size vehicles have high prices.
- The compact type vehicles have the average price.
- The unknown type uncharted has having higher price among others

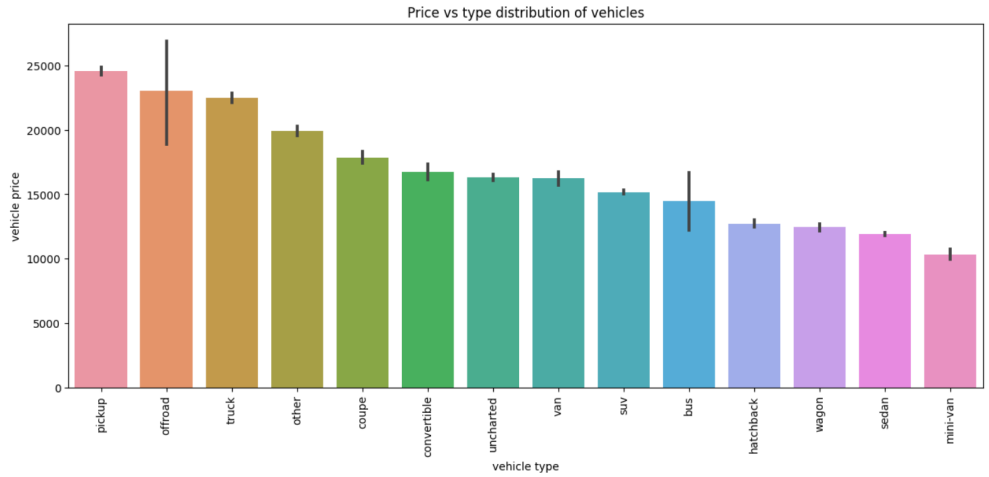
9. Distribution of vehicle types



Insights:

- Sedan and SUV types are the highest available cars in the market followed by uncharted, pickup, and truck types.
- Van, mini-van, convertible, offroad and bus are least available for resale in the market

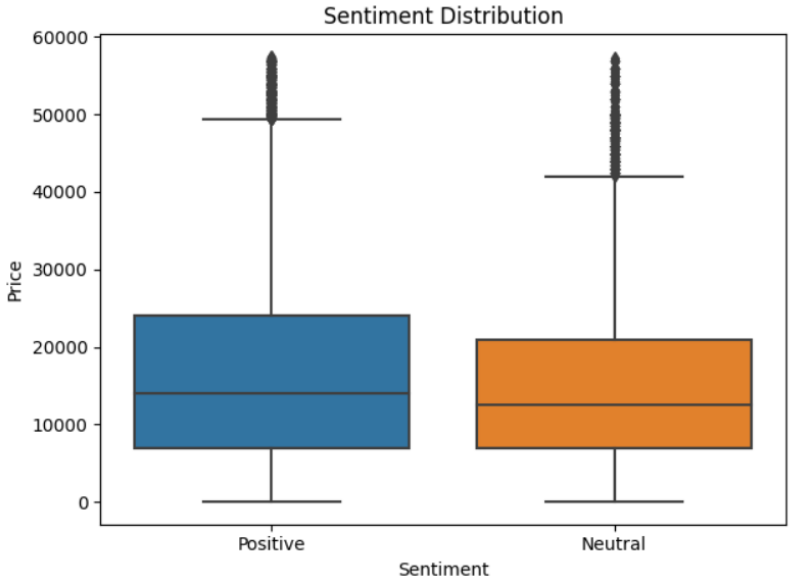
10. Price distribution based on vehicle type



Insights:

- pickup, offroad and truck vehicle types are expensive, whereas hatchbacks, wagons, sedans, and mini-vans are cheap in price.
- price distribution is almost equal for the remaining vehicle types.

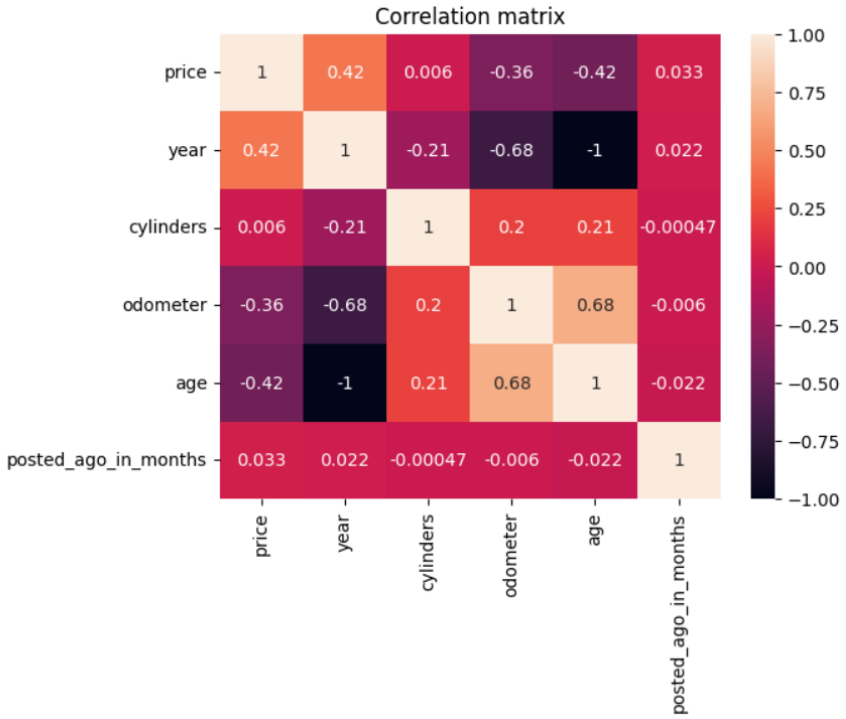
11. Sentiment Distribution



Insights:

- We can see that vehicles with positive sentimental descriptions have higher prices than neutral ones.
- We can also deduce that there is no vehicle with a negative description. It makes sense since no one will give a negative description of an advertisement for selling a vehicle

12. Correlation Matrix



Insights:

- price and cylinders have a negligible correlation. So, cylinder type does not affect the price of the vehicle.
- posted_ago_in_months have almost no correlation with any of the features. So, it does not play any role in deciding the price of a vehicle.
- price and age are inversely proportional to each other. As the age of the vehicle increases price of the vehicle decreases

What did we learn?

- There are various factors like model, manufacturer, age, fuel type, condition, etc impact the used car market.
- We have understood the significance of each feature in predicting the price of a car.
- Age is a significant player in impacting the price of used cars. This was our assumption and the graph data also indicated the same.
- We observed that the distribution of the target variable ('price') is concentrated around the average price range. This suggests that our model may perform well in predicting prices within this central range of values, which could be beneficial for common people seeking to understand or estimate used car prices.

How will we use this information?

- From the different plots above, we have observed some linear patterns between the target and independent features, which suggests that a linear regression model can be suitable for this data.
- From the correlation matrix visualization, we have observed some high correlations between year and age, which may affect the model's performance. So we may drop one of these columns in the later phases, based on our experiments. Also, based on the analysis, we will select important features that play a key role in the model's performance.
- Additionally, we will use this information to build various models and develop a user-friendly GUI with which a seller or a buyer can pass relevant inputs and get a predicted price of used cars which helps them make informed decisions.

Phase 2

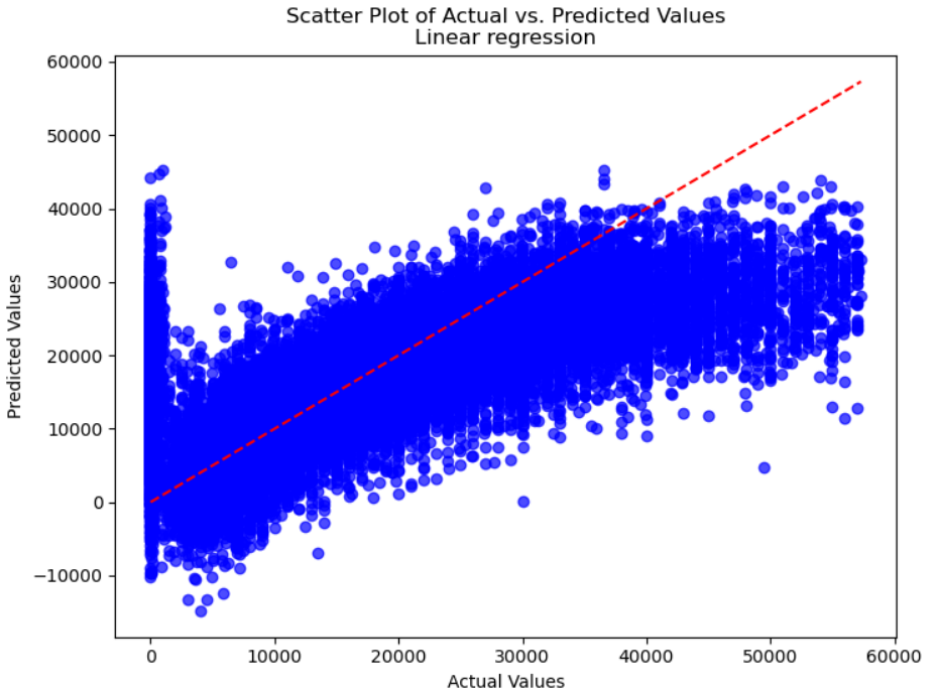
Objective:- Since the pricing of used cars has complexities, our objective is to develop an approach to predict the price of used cars based on relevant features.

For training and testing our model, we utilized the 'train_test_split' module from sci-kit-learn (sklearn). We created a data split with a test size of 30% of the entire dataset. We employed metrics such as mean squared error, mean absolute error, and r-squared score (r2_score) from the 'sklearn.metrics' library during testing.

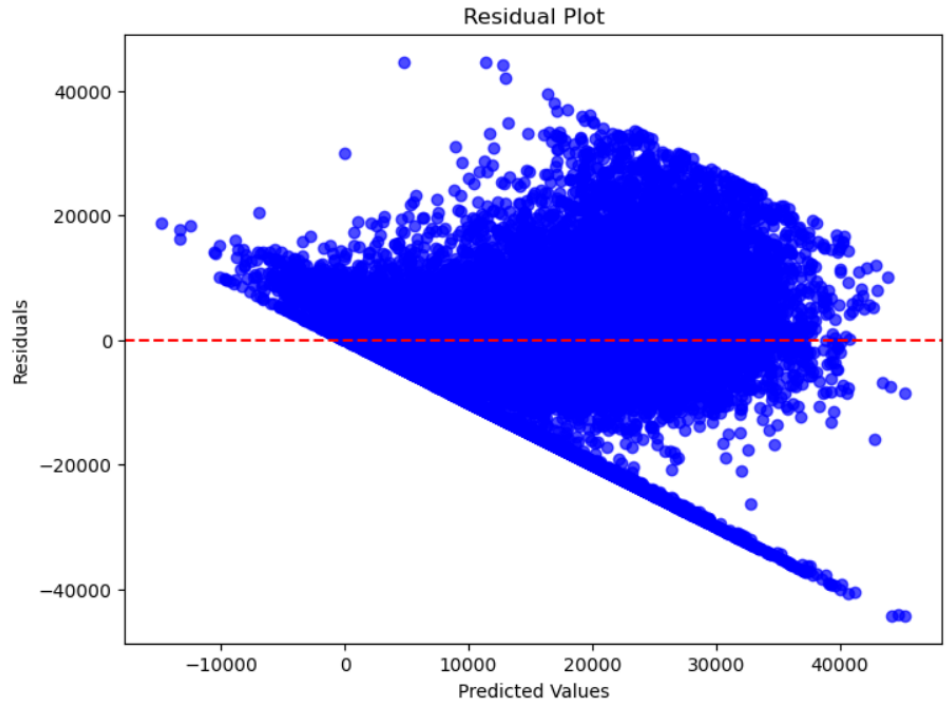
Below is the list of models that we have applied to gain intelligence from our data.

1) Linear regression

- Why did we choose this algorithm?
    - We initially chose to begin with Linear Regression as our primary algorithm due to its simplicity and interpretability. Since our problem involves predicting car prices, which is a regression task, Linear Regression seemed like a logical starting point. During the exploratory data analysis (EDA) in the first phase, we observed some linear patterns in the data. This reinforced our belief that Linear Regression would be a suitable algorithm for our price prediction problem. An additional advantage of this choice is that we can employ regularisation techniques to enhance accuracy, even if we encounter suboptimal performance initially. Therefore, we see ample potential in utilising this algorithm, so we have decided to proceed with it.
- How did we train the model for our problem?
    - We used the training dataset, which was obtained through the 'train_test_split' method, directly with the linear regression model. Given that there were no parameter tuning requirements for linear regression, we proceeded to train the model using the training data and then tested its performance on the test data.



Observation: The plot, which compares actual and predicted values, illustrates the effectiveness of our model. It is evident that the model consistently delivers accurate predictions for price values above 1000 and below 35,000. However, beyond this threshold, the model's predictive performance diminishes, resulting in less accurate predictions as prices increase



Observation: The residual plot reveals a linear pattern among the data points centered around the horizontal line. This indicates that the model is not adequately capturing the relationships between the variables.

Effectiveness of the algorithm/model using below evaluation metrics:

R2_score: 0.4327
Mean squared error: 89620773.13
Mean absolute error: 6706.76
Root mean squared error: 9466.82

R-squared (R^2) Score:
R^2 value for our training data is 0.4327 which means that 43.27% of the variation in used car prices is explained by the model. This indicates that the model has captured some of the relationships between the features and the car prices but more than half of the variance is still unexplained by the model.

Mean Squared Error:

MSE value is very high, which means that the model has large errors and predictions are off by a considerable amount on average.

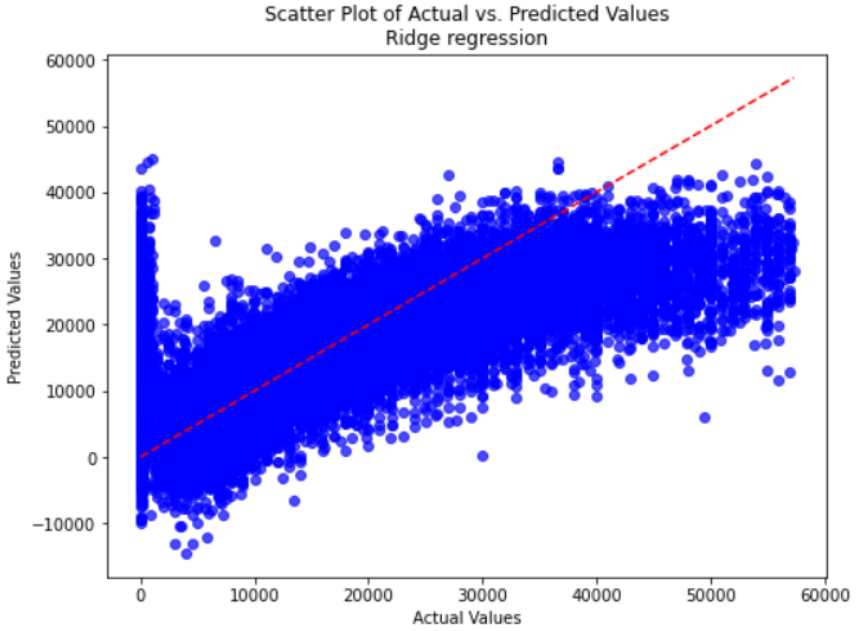
Mean absolute error(MAE):

MAE of 6,706.76 means that the average prediction by the model is about $6,706.76 away from the actual price. So, this model is not precise enough for pricing used cars.

Root Mean Squared Error (RMSE): RMSE value of 9,466.82 also indicates large variances in the prediction errors.

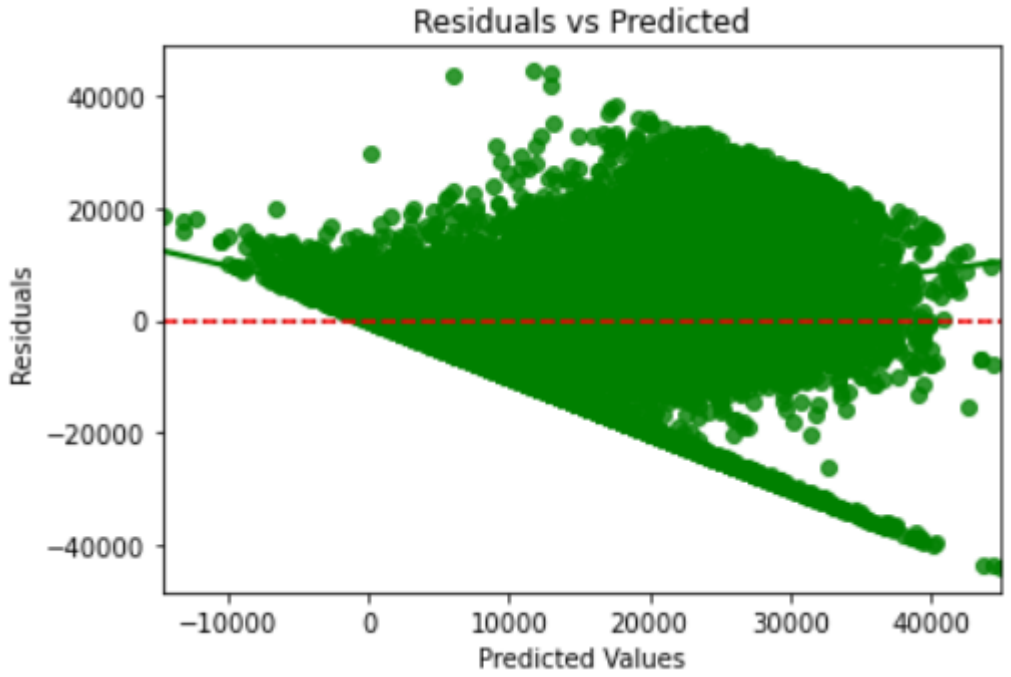
2) Ridge regression

- Why did we choose this algorithm?
    - Upon reviewing the outcomes of our initial linear regression model, it became apparent that the performance did not meet our desired standards. In response, we opted to transition to Ridge regression, a well-regarded alternative to linear regression. Ridge regression is renowned for its efficacy in mitigating issues related to model overfitting and multicollinearity, while simultaneously enhancing the model's generalization capabilities
    - We assume that this approach will provide more stable and reliable predictions for vehicle prices.
- How did we train the model for our problem?
    - To set the hyperparameter alpha for our regression model, we employed a trial-and-error approach. We initiated the process with a relatively high value, such as 19, to enforce strong regularisation and reduce the risk of overfitting. Ultimately, we settled on the value of 20.336 as it consistently delivered the most favorable results among all other values considered.
    - We utilized the same training dataset that was previously employed for Linear Regression.



Observation:

- In this plot, we observe a similar pattern as seen in the linear regression scatter plot. The threshold values are also nearly identical.
- The model consistently delivers accurate predictions for price values above 1000 and below 35,000. However, beyond this threshold, the model's predictive performance diminishes, resulting in less accurate predictions as prices increase



Observation: Here also we can have the same observation as we got for the linear regression residual plot. The residual plot reveals a linear pattern among the data points centered around

the horizontal line. This indicates that the model is not adequately capturing the relationships between the variables.

Effectiveness of the algorithm/model using below evaluation metrics:

R2 score: 0.4319
Mean squared error: 89744847.83
Mean absolute error: 6709.35
Root mean squared error: 9473.38

R-squared (R^2) Score:
Similar to linear regression, the model explains 43.19% of the variance in the car prices. This model also has captured some of the relationships between the features and the car prices but more than half of the variance is still unexplained by the model.

Mean Squared Error(MSE):
MSE value for ridge regression is also very high like linear regression. The model has large errors in predicting the car prices and so this model will significantly impact decision-making processes based on these predictions.

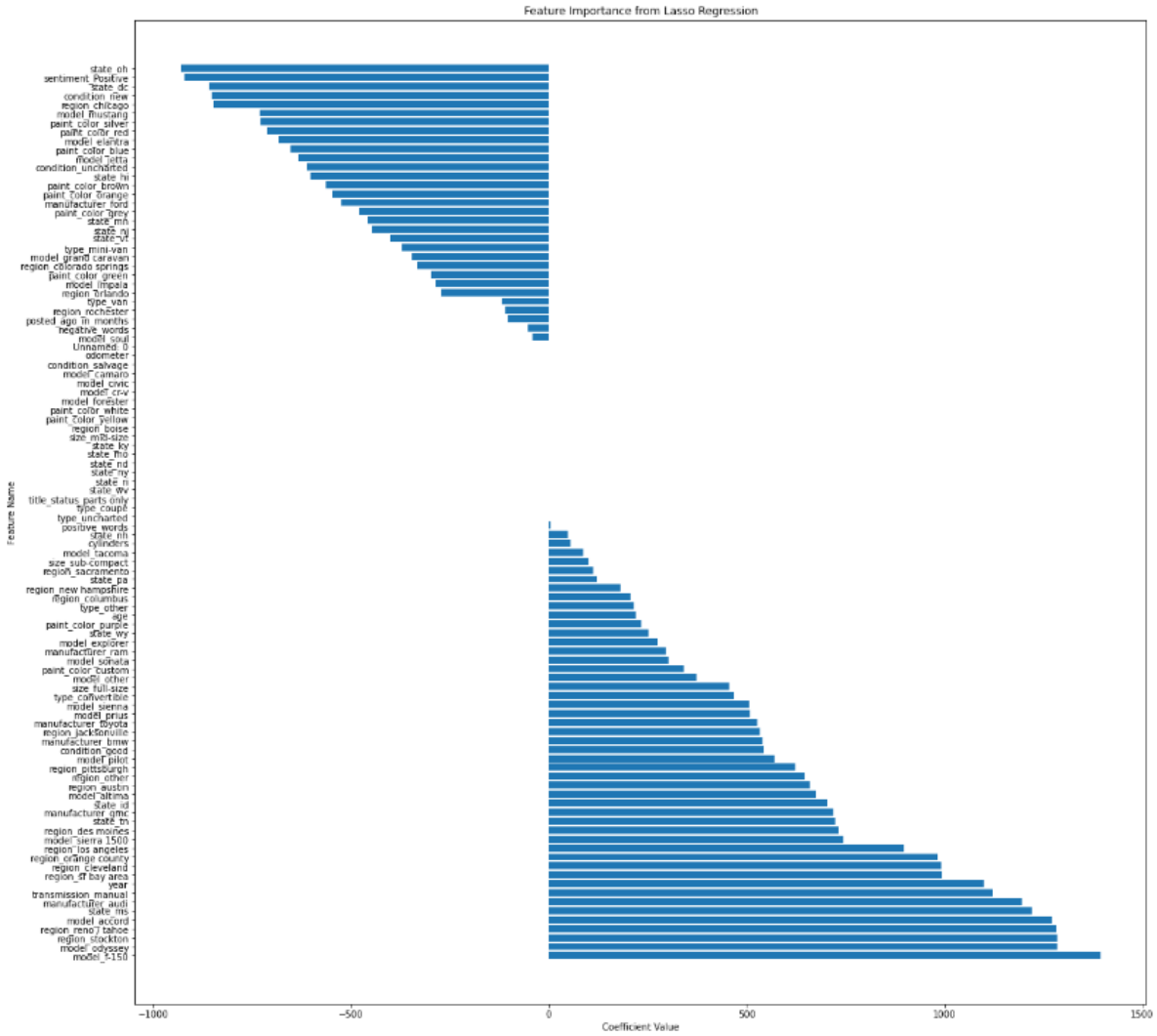
Mean absolute error(MAE):
MAE value for ridge regression is almost the same as that we noticed in Linear regression. So, this model is also not precise enough for pricing used cars.

Root Mean Squared Error (RMSE): Similar to linear regression, this model also has a considerable spread in the residuals.

3) Lasso Regression

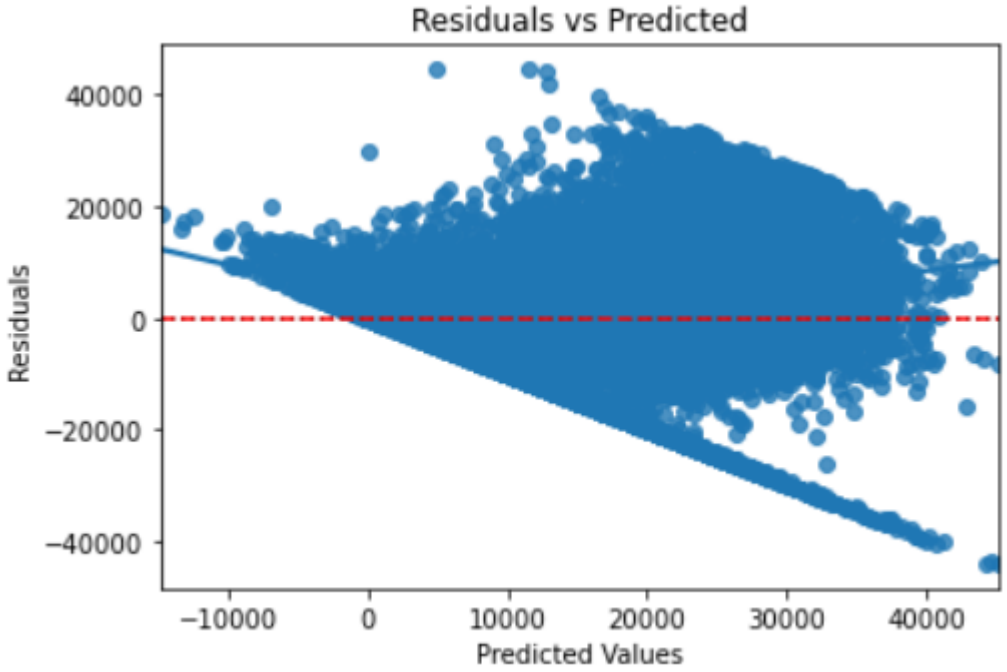
- Why did we choose this algorithm?
    - Lasso regression is a variation of linear regression, similar to Ridge regression. We considered it as a potential model for predicting car prices.
    - Lasso regression has an improved ability to handle high-dimensional data, which is particularly relevant for our dataset, which has over 200 features after preprocessing.
    - Lasso regression offers enhanced model interpretability and can strike a balance between bias and variance.
    - Taking these factors into consideration, we concluded that Lasso regression is indeed a suitable model for our predictive task.
- How did we train the Knn regressor for our problem?
    - To set the hyperparameter alpha for the lasso regression model, we employed a trial-and-error approach. Ultimately, we settled on the value of 0.0001 as it consistently delivered the most favorable results among all other values considered.

- We utilized the same training dataset that was previously employed for Linear Regression.



Observation:

- From the above plot, we can see that coefficient value is positive for certain regions like region_losangeles, region_austin, region_pittsburg, and other features like transmission_manual, model, condition_good, positive_words, etc. This means that these features are important and it indicates that as the feature value increases, the dependent variable(used car price) also increases
- And some of the features like condtion_new, paint_color_*, model_ford, model_jetta etc have negative coefficient value, indicating that as the feature value increases, the car price value decreases.
- For certain features, coefficient value is zero, which means those features are not important in predicting the car price.



Observation: This is also the same as that we have in Linear and Ridge regressions, because the results for these three linear regressions are almost the same.

Effectiveness of the algorithm/model using below evaluation metrics:

R2 score: 0.4327
Mean squared error: 89622337.84
Mean absolute error: 6706.70
Root mean squared error: 9466.91

We can clearly see that the above metrics are very close to that of the linear regression model. So this model is also not the right fit for used car price prediction.

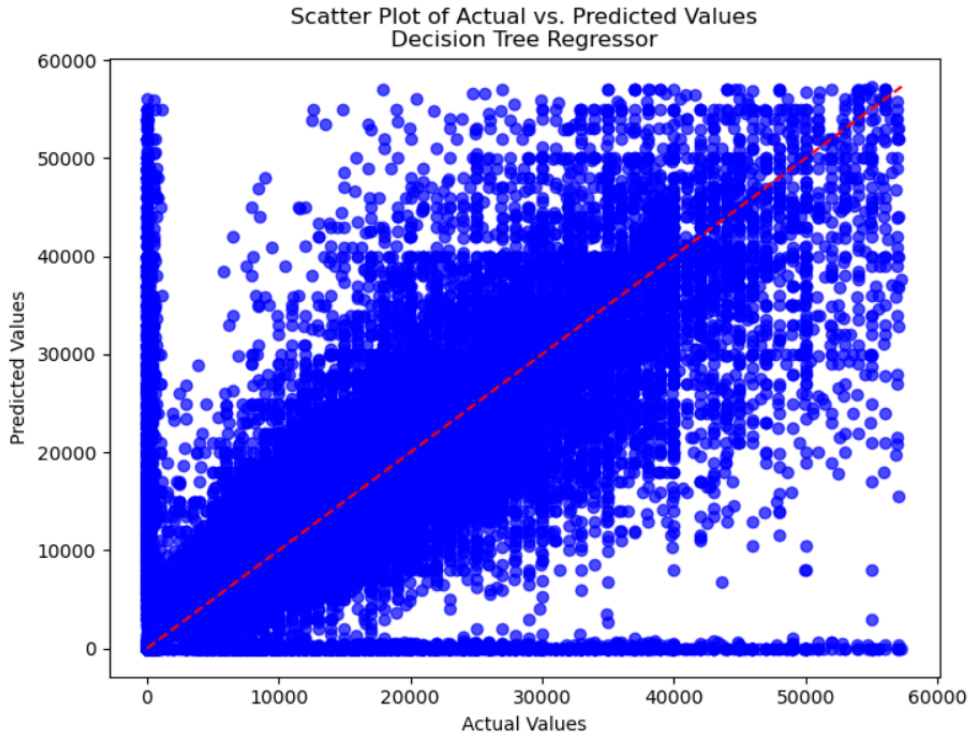
We can conclude that Linear Regression, Ridge Regression, and Lasso Regression are not good enough models to predict the price of used cars. This could be due to non-linear relationships that these regressions cannot capture or important variables that are not included in the model. So we wanted to try models like KNN, Decision Tree, Random Forest, etc which capture non-linear relationships.

4) Decision Tree Regressor

- Why did we choose this algorithm?
    - Decision tree regressor is a versatile and interpretable model technique that is very effective in dealing with non-linear, complex, and mixed datatypes.
    - Since our experiences with various linear regression models and their variations yielded suboptimal performance, we are exploring the potential of the Decision Tree Regressor to provide improved results

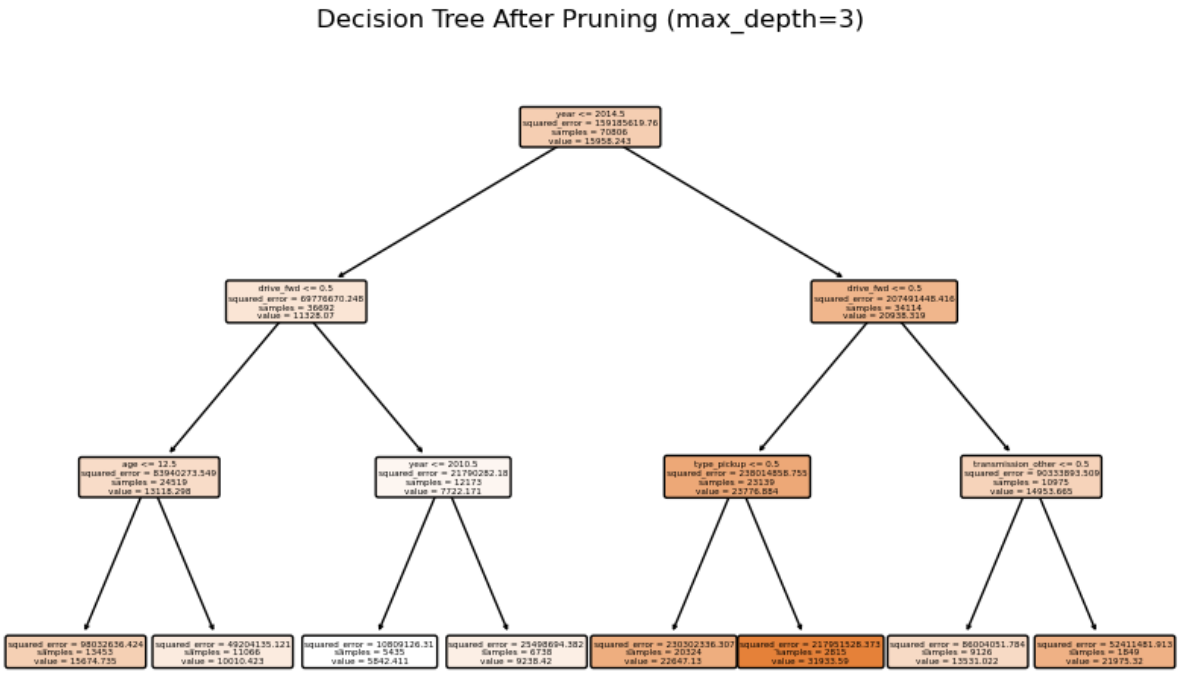
- Despite our exploratory data analysis (EDA) hinting at potential linear patterns, the anticipated benefits did not materialize as expected. Therefore, we are considering this model as a means to gain deeper insights into our data and its potential treatment.

- How did we train the model for our problem?
  - We utilized the same training dataset that was previously employed for other models. We didn't use any hyperparameters for this model.



Observation:

- In this plot, it is evident that the model performs poorly in predicting values, as the data points are scattered across the plot. This highlights the inefficiency of the model in accurately predicting values.
- Even though there is a significant number of data points in the diagonal part of the graph, ranging from 1000 to 40000, the presence of other points that deviate from the actual values contributes to an overall poor accuracy of the model.



Observation: The above decision tree depicts the structure of the decision tree for making decision analysis. This is a pruned decision tree with a max_depth of 3. The root decision was made by year first

Effectiveness of the algorithm/model using below evaluation metrics:

R2_score: 0.3083
Mean squared error: 109275909.36
Mean absolute error: 6402.73
Root mean squared error: 10453.51

R-squared (R^2) Score:

The R-squared value suggests that only 30.83% of the variance in car prices can be explained by the model, which means it does not capture a large portion of the variability in car prices.

Mean Squared Error (MSE):

MSE value is quite high, indicating that the model is not capturing the necessary patterns in the data or overfitting to the training data leading to large errors in the prediction.

Mean Absolute Error (MAE):

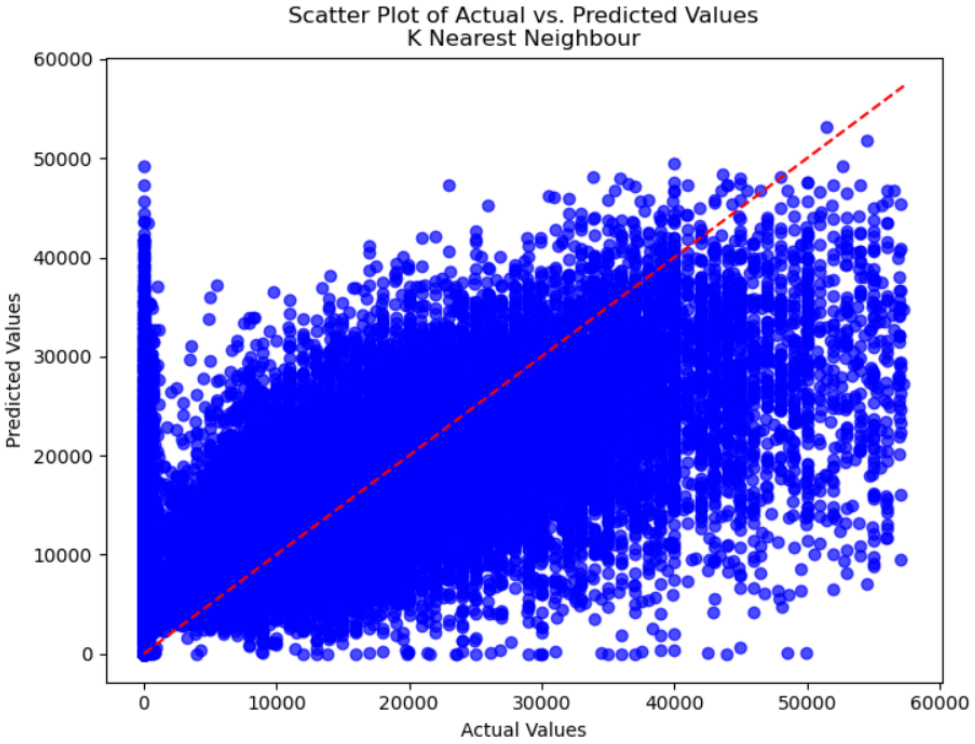
The model's predictions are approximately $6,402.73 away from the actual prices. This suggests that the predictions are inaccurate for some cars.

Root Mean Squared Error (RMSE):

RMSE value is also high, indicating significant variability in the individual errors of the model's predictions.

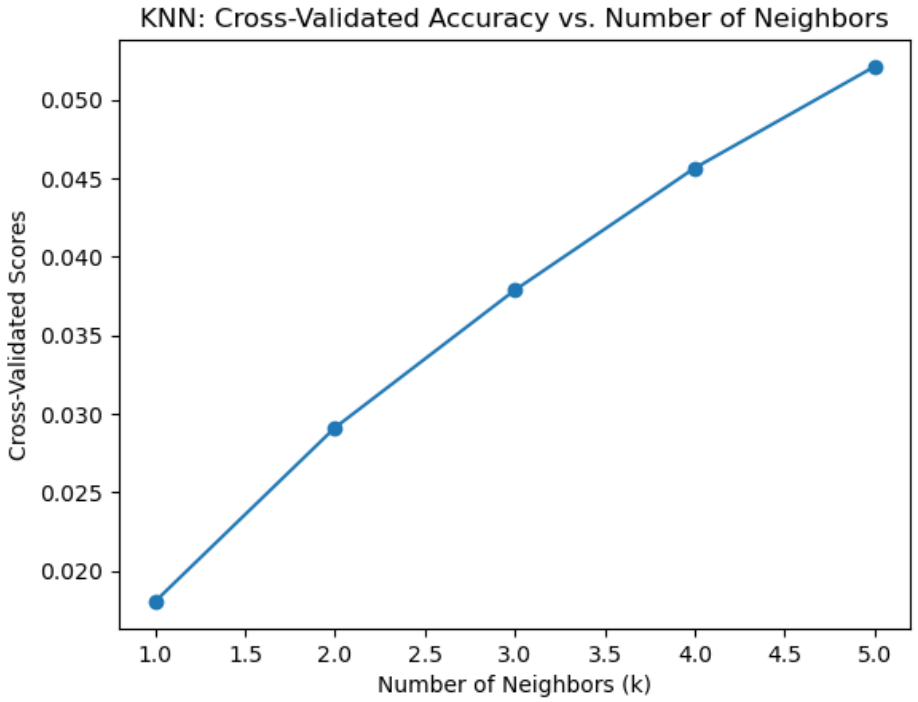
5) K Nearest Neighbors Regression:

- Why did we choose this algorithm?
    - KNN algorithm is well suited for problems which are having similarities within the features. Assuming the cars we are going to predict are similar, we are opting to train the data with the KNN regressor algorithm.
    - KNN provides flexibility towards non-linearity. It does not assume any functional relationship between the price feature and the other dependent features.
- How did we train the Knn regressor for our problem?
    - Choosing the K value: Trying with different k values, we have chosen k as 5 as it is neither too small to result in noisy data nor too large to overtrain the data. Choosing the optimal k value would be a better balance between bias and variance.



- The above scatter plot tells us that a considerable proportion of points aligns along the diagonal which means car price prediction is accurate for prices ranging from $10000 to $40000 but there are numerous points that are deviating from the diagonal line meaning the overall accuracy is low.

Elbow Graph:



Observation: From the above graph, we can see that as the k value increases, the cross validated score is also increasing, suggesting that with an increasing number of neighbours, predictive accuracy will improve.
But the cross validated scores are low for all k values, meaning the model's price prediction is not accurate enough.

Effectiveness of the algorithm/model using below evaluation metrics:

R2_score: 0.3951
Mean squared error: 95564661.71
Mean absolute error: 7085.96
Root mean squared error: 9775.72

R-squared (R^2) Score:
0.3951 R2 score suggests that the KNN model explains approximately 39.51% of the variance in the used car prices. A significant portion of the variance in the data is not captured by the model, which means the KNN regressor is less effective at capturing the variance in car prices.

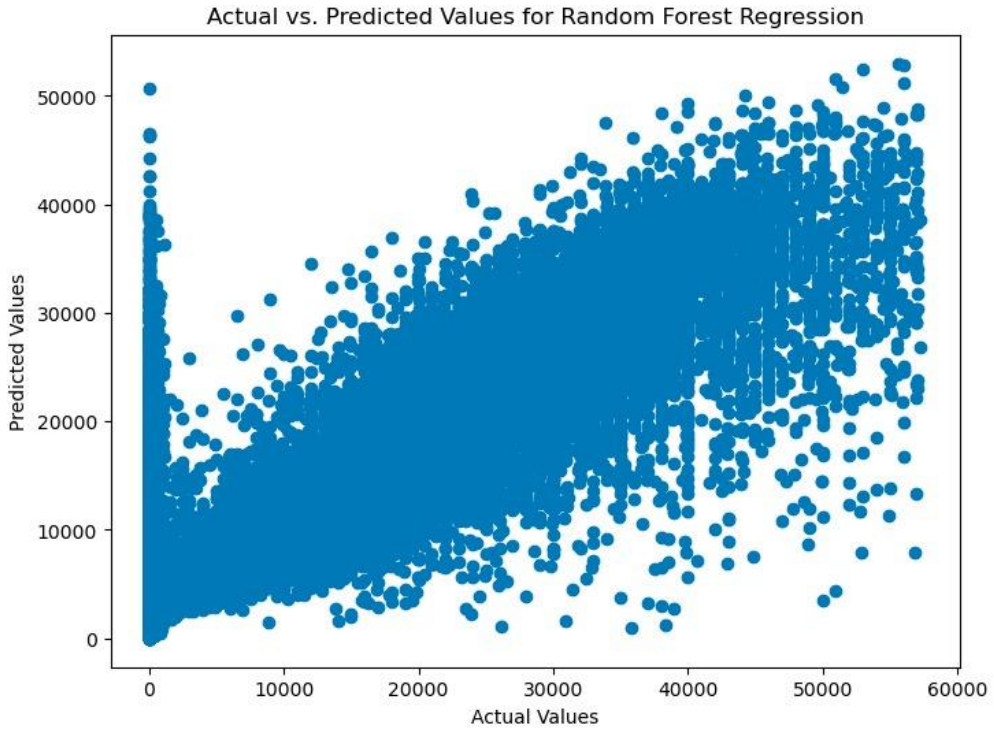
Mean Squared Error (MSE):
Higher MSE value indicates that KNN model is making large errors on average in its predictions.

Mean Absolute Error (MAE):
MAE value is also relatively high, $7,085.96 off from the actual car prices, which implies that KNN model is less accurate in car price prediction.

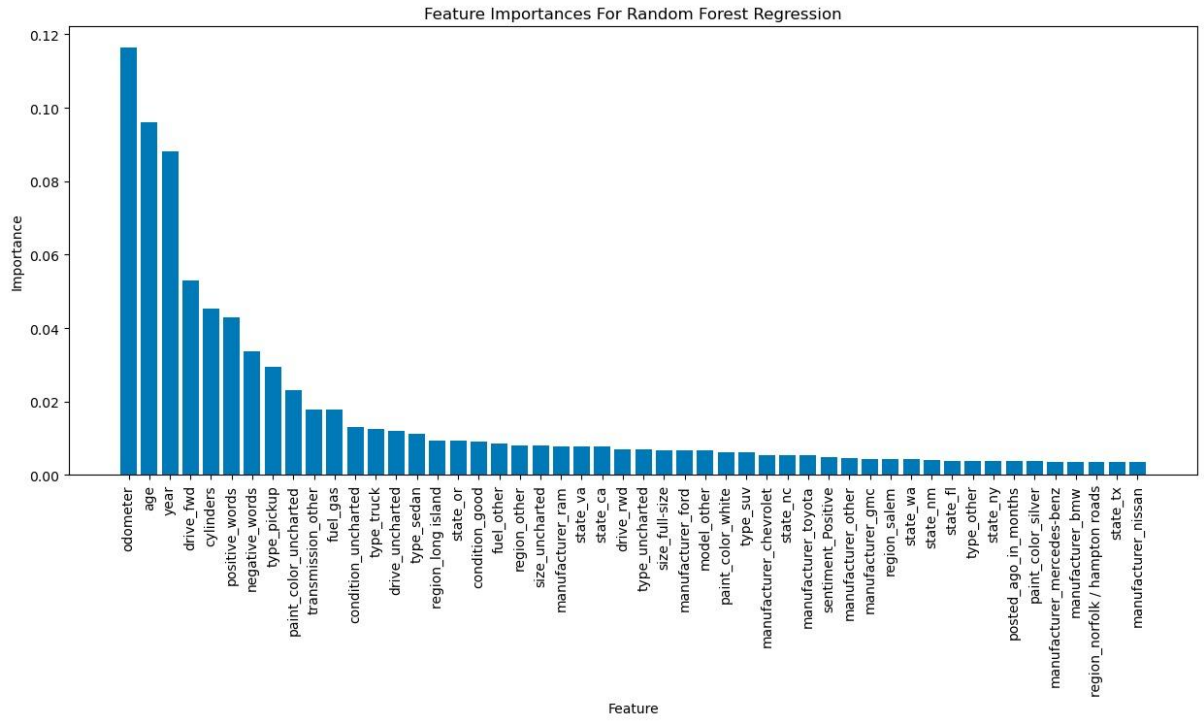
Root Mean Squared Error (RMSE):
RMSE is the square root of the MSE. RMSE value of 9,775.72 shows that KNN model has greater variability in the prediction errors.

6) Random Forest Regression:

- Why did we choose this algorithm?
    - Random forest tries to identify complex non-linear patterns in the data, which may be suitable for our used car price prediction problem, having a variety of features.
    - Random forest regression is an ensemble learning method, which combines multiple decision trees and provides a generalized machine learning model that handles robust data.
    - Random Forest also provides feature importances, which will help in selecting features that have a significant impact on car prices.
- How did we train the random forest regressor for our problem?
    - Choosing Hyperparameters: With trial and error method, we have chosen the following hyperparameters for our model.
        - N_estimators: 150
        - Min_samples_leaf: 1
        - Oob_score: True
        - Max_features: 0.3



Observation: The above scatter plot depicts the relation between the actual values and the predicted values, estimated by the random forest regressor.



Observation: The bar plot above depicts the information about features that are ranged from most significant feature to the least significant feature. Age, year, cylinders, positive_words and negative_words are derived features from the data, which are contributing as most useful features for our used car price prediction model.

Effectiveness of the algorithm/model using below evaluation metrics:

R2_score: 0.658
Mean absolute error: 4768.08
Mean squared error: 55160878.59
Root mean squared error: 7427.04

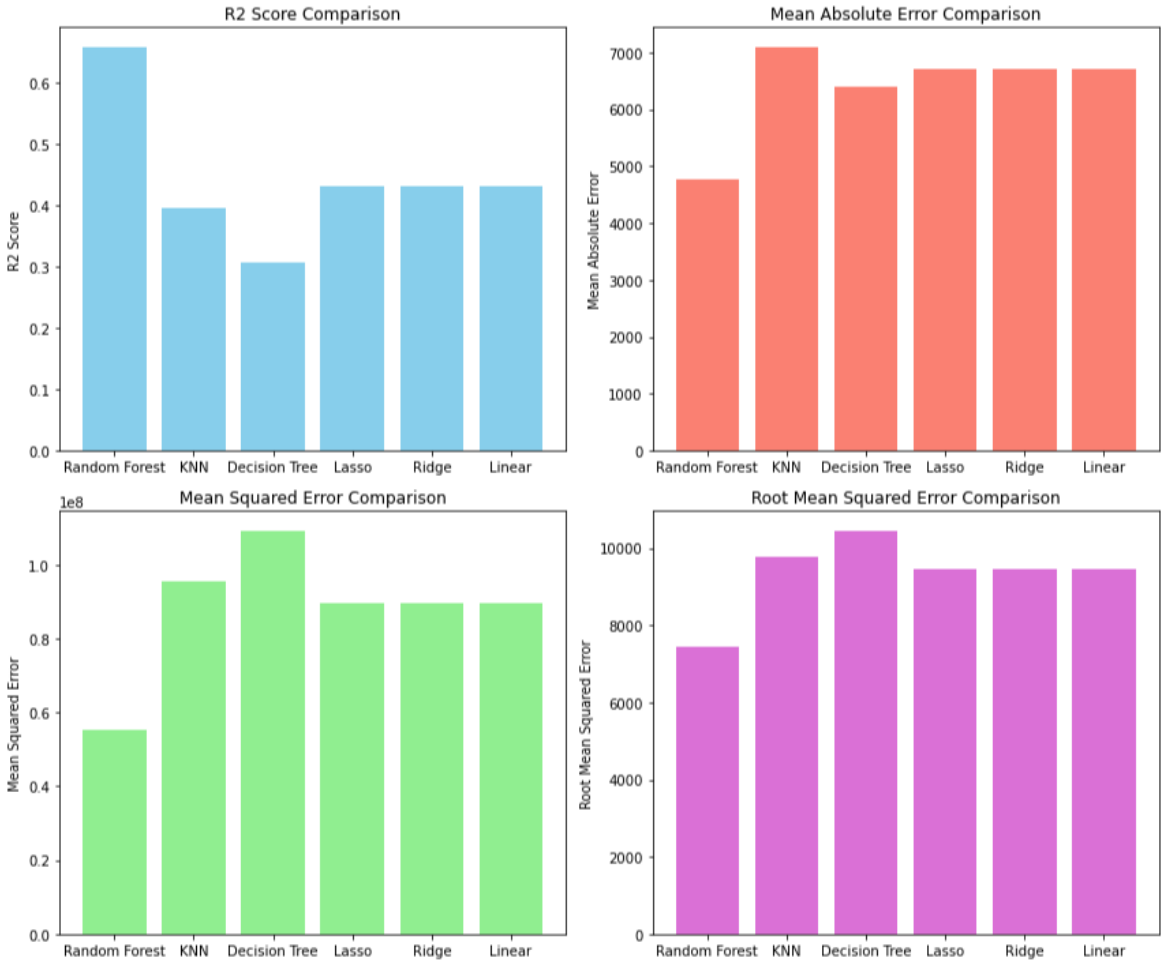
R-squared (R^2) Score:
This model captures 65.8% of the variance in the used car prices. This is a good improvement over the previously discussed models. A higher R-square value indicates that it is a better fit to the data.

Mean Absolute Error (MAE):
MAE value is lower compared to other models that we discussed. This indicates that the predictions made by Random Forest are closer to the actual values on average.

Mean Squared Error (MSE):
MSE value by this model suggests that it has fewer large errors in its predictions.

Root Mean Squared Error (RMSE):
RMSE of 7,427.04 is also an improvement over the other models, indicating that the Random Forest is generally making better predictions with less variation in error.

After applying six different models on our dataset, we have compared all the six models through a comparison plot. Please see below for the visualisation.



- When compared to other algorithms, random forest regressor, having the ability to capture complex non-linear patterns within the features, performed better in terms of r2 score, mean absolute error, and mean squared error.
- The feature importance provided by the model helped in understanding the significance of each feature in predicting used car prices.
- The model has performed better for the testing data, which means the model can generalize patterns well.
- In addition, random forest regressors can handle a large number of features, which is suitable for our data.
- Hence, we will use the Random Forest algorithm to predict used car prices.

Phase 3

Instructions on how to run preprocessing code:

Firstly, to run the code developed in phase 2, we need a dataset which includes all the following feature columns.

Vehicle manufacturer (Ford, Chevrolet, …)

Region (Columbus, Houston, …)

Vehicle condition (Excellent, Good, …)

Fuel type (Gas, Diesel, Hybrid, …)

Title status (Clean, Salvage, …)

Transmission (Automatic, manual, …)

Vehicle size (Full-size, mid-size, compact, …)

Paint color (red, grey, black, …)

Vehicle Type (sedan, suv, pickup, …)

Model of the vehicle (F-150, Silverado 1500, …)

State (ca, fl, az, …)

Drive options (4wd, fwd, rwd, …)

Vehicle manufactured year (1980 to 2023)

No of cylinders (3 to 8)

Odometer reading

Posting date of the used vehicle

Description of the vehicle

Deploying the data product

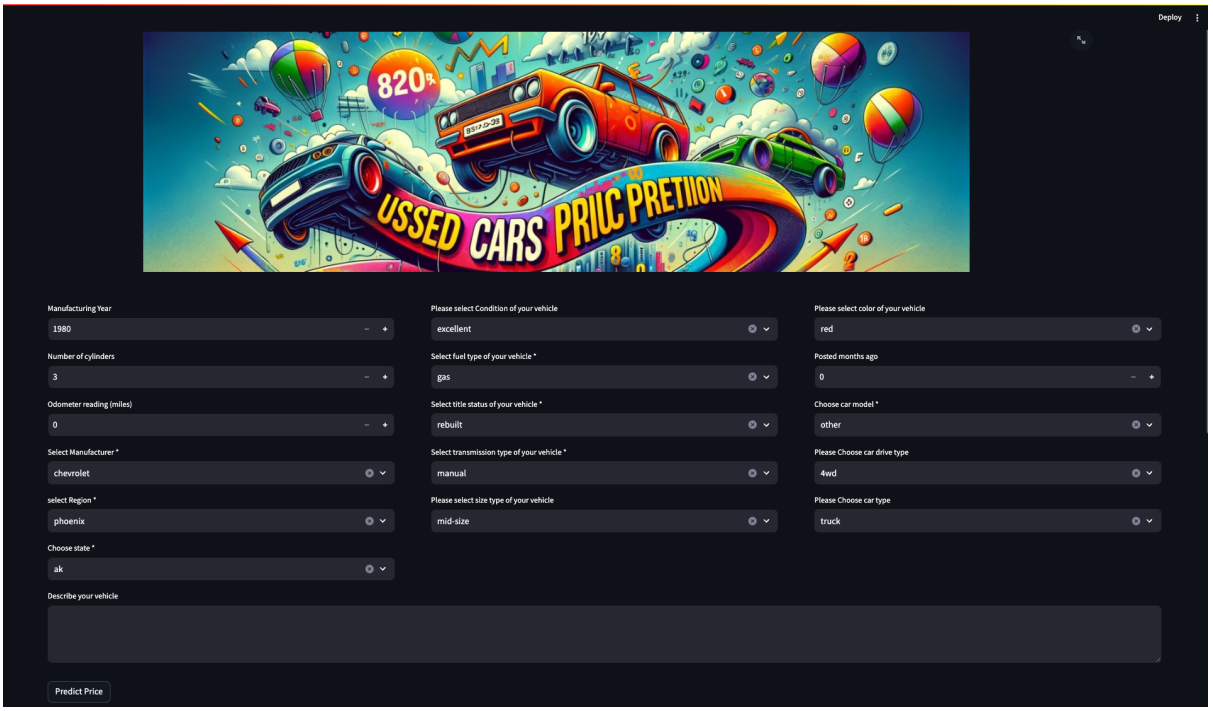
We have used streamlit to build the application. If the library is not installed previously, then please install it using the '*pip install streamlit*' command.

Streamlit Application:

The trained Random Forest classification model was integrated into the streamlit application using python's streamlit library.

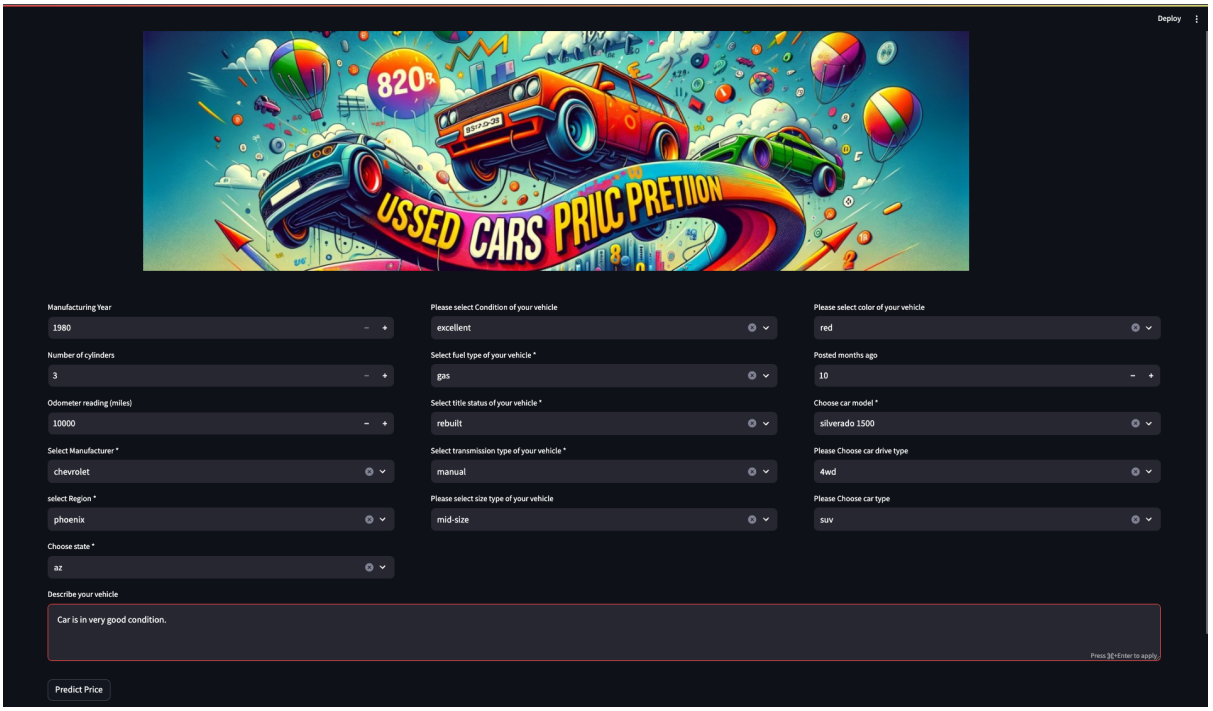
How to run the application:

1. Clone the source repository from the github: https://github.com/gokulpulikkal/UsedCarPricePrediction
2. Change the directory to repository directory.
3. Install all the required libraries from the requirements.txt file.
4. Run the application using command: '*streamlit run webApp.py*'
5. This will open a web application on the user's default browser or you may get a localhost address from which you can access this application, which will look something like the screenshot below.

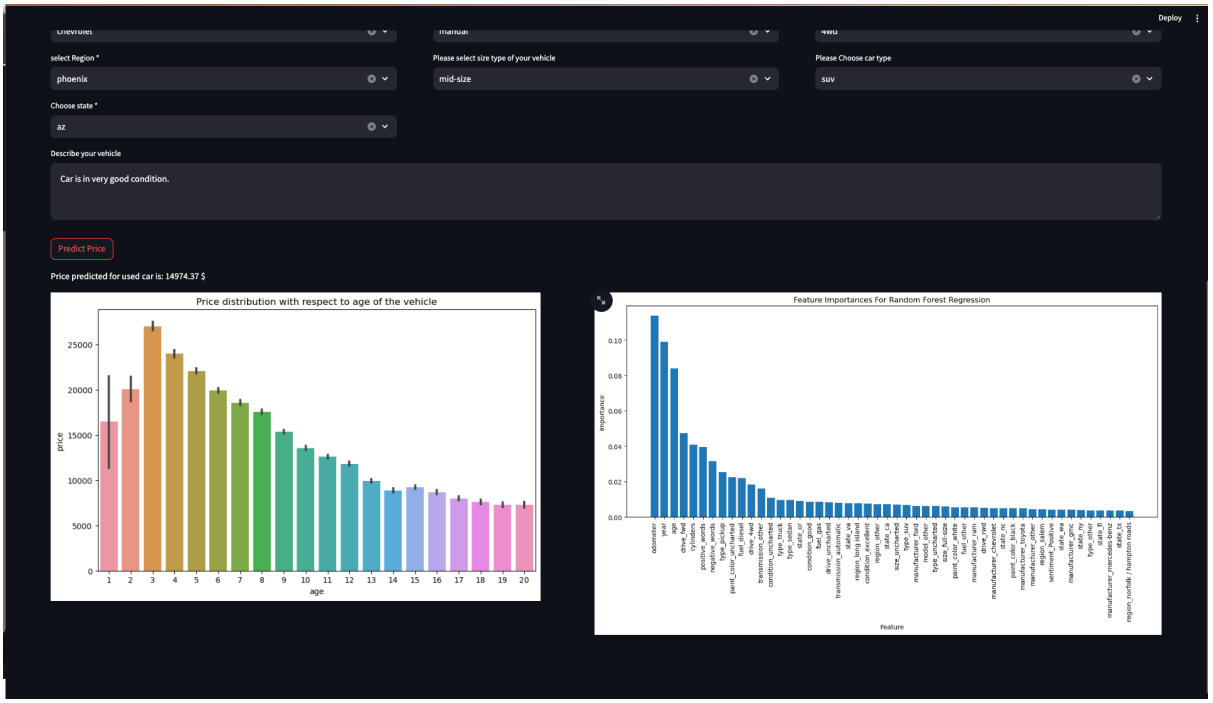


The GUI gives an option to the user to input all the required attributes mentioned in the above section.

6. Filling vehicle data in input fields for prediction:

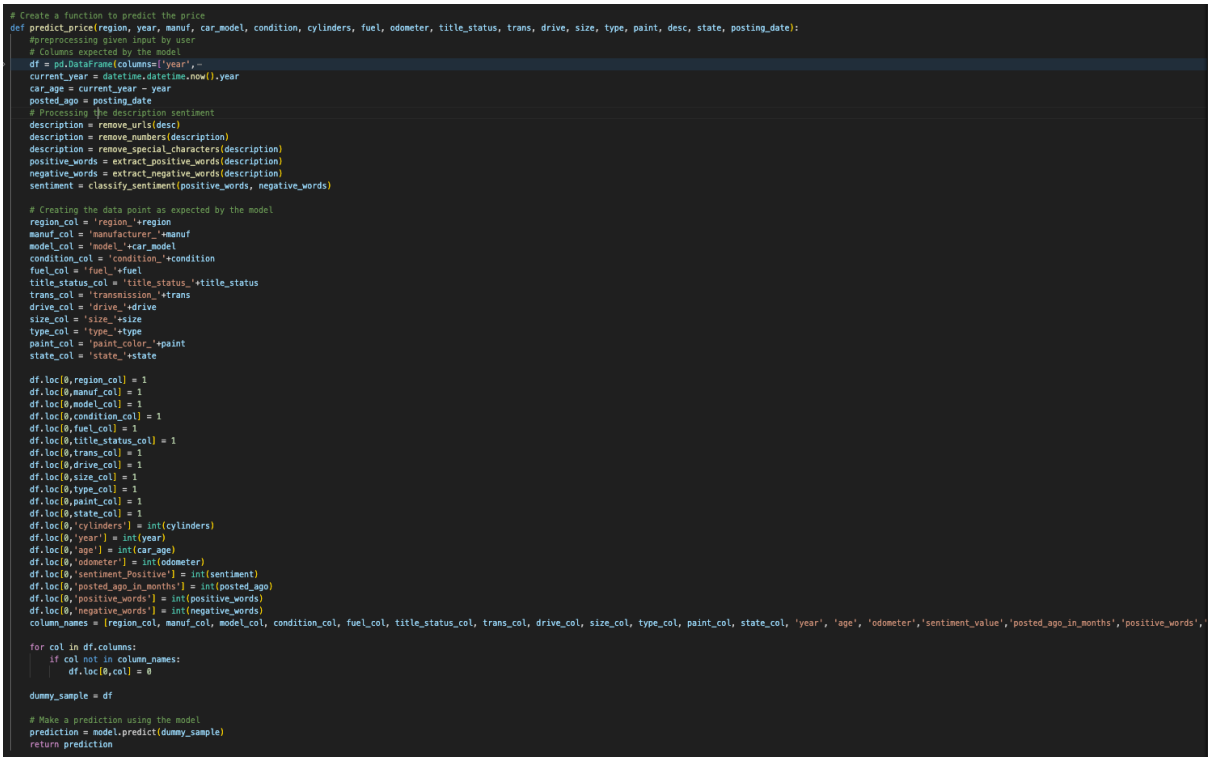


7. Then click on the Predict Price button to trigger the prediction operation.



As a result we can view the approximate predicted price along with the visualizations of feature importance and how price ranges with change in the age of a car.

Code Sample for phase 3:



```python
# Create a function to predict the price
def predict_price(region, year, manuf, car_model, condition, cylinders, fuel, odometer, title_status, trans, drive, size, type, paint, desc, state, posting_date):
    #preprocessing given input by user
    # Columns expected by the model
    df = pd.DataFrame(columns=['year',-
    current_year = datetime.datetime.now().year
    car_age = current_year - year
    posted_ago = posting_date
    # Processing the description sentiment
    description = remove_urls(desc)
    description = remove_numbers(description)
    description = remove_special_characters(description)
    positive_words = extract_positive_words(description)
    negative_words = extract_negative_words(description)
    sentiment = classify_sentiment(positive_words, negative_words)

    # Creating the data point as expected by the model
    region_col = 'region_'+region
    manuf_col = 'manufacturer_'+manuf
    model_col = 'model_'+car_model
    condition_col = 'condition_'+condition
    fuel_col = 'fuel_'+fuel
    title_status_col = 'title_status_'+title_status
    trans_col = 'transmission_'+trans
    drive_col = 'drive_'+drive
    size_col = 'size_'+size
    type_col = 'type_'+type
    paint_col = 'paint_color_'+paint
    state_col = 'state_'+state

    df.loc[0,region_col] = 1
    df.loc[0,manuf_col] = 1
    df.loc[0,model_col] = 1
    df.loc[0,condition_col] = 1
    df.loc[0,fuel_col] = 1
    df.loc[0,title_status_col] = 1
    df.loc[0,trans_col] = 1
    df.loc[0,drive_col] = 1
    df.loc[0,size_col] = 1
    df.loc[0,type_col] = 1
    df.loc[0,paint_col] = 1
    df.loc[0,state_col] = 1
    df.loc[0,'cylinders'] = int(cylinders)
    df.loc[0,'year'] = int(year)
    df.loc[0,'age'] = int(car_age)
    df.loc[0,'odometer'] = int(odometer)
    df.loc[0,'sentiment_Positive'] = int(sentiment)
    df.loc[0,'posted_ago_in_months'] = int(posted_ago)
    df.loc[0,'positive_words'] = int(positive_words)
    df.loc[0,'negative_words'] = int(negative_words)
    column_names = [region_col, manuf_col, model_col, condition_col, fuel_col, title_status_col, trans_col, drive_col, size_col, type_col, paint_col, state_col, 'year', 'age', 'odometer','sentiment_value','posted_ago_in_months','positive_words',

    for col in df.columns:
        if col not in column_names:
            df.loc[0,col] = 0

    dummy_sample = df

    # Make a prediction using the model
    prediction = model.predict(dummy_sample)
    return prediction
```

The code above shows the predict_price method which preprocesses the input given by the user and loads it to the machine learning model and returns the predicted price of the used vehicle.

What model did we use for our product? Why did we use it?

In Phase 2, we have trained our dataset with 6 different models:

Linear Regression
Ridge Regression
Lasso Regression
Decision Tree
KNN
Random Forest

After analysing evaluation metrics from these 6 models, we have selected the Random Forest model for our product.

We tuned below parameters for the Random Forest model
N_estimators: 150
Min_samples_leaf: 1
Oob_score: True
Max_features: 0.3

The reason we used Random Forest model is: compared to other models, random forest regressor has the ability to capture complex non-linear patterns within the features, performed better in terms of r2 score, mean absolute error, and mean squared error.

The feature importance provided by the model helped in understanding the significance of each feature in predicting used car prices.

The model has performed better for the testing data, which means the model can generalize patterns well.

In addition, random forest regressors can handle a large number of features, which is suitable for our data.

So, we ended up using the Random Forest model for our product.

Recommendations and Insights:

With our product, users gain transparency into the factors affecting used car prices. By understanding the impact of age, manufacturer, model, size, type, etc., they can make more informed decisions when buying or selling a used car.

Users can use this as a reference point for negotiation between buyers and sellers. This can help both parties reach a fair price that reflects the car's true value.

Also, underestimation or overestimation of a used car's price can be avoided which saves money for the users.

Ideas to extend the project:
Develop functionalities to analyze market trends based on real-time data, providing insights into future price fluctuations.

Create a mobile app for easier accessibility and on-the-go price estimations

Incorporate large datasets from various sources to enhance the model's ability to learn and adapt to changing market conditions.

Integrate the prediction model with online data marketplaces for used cars to provide real-time price estimates directly on listings.

Reference :

1. https://www.mygreatlearning.com/blog/understanding-of-lasso-regression/#:~:text=Lasso%20regression%20is%20a%20regularization,i.e.%20models%20with%20fewer%20parameters).

2. https://scikit-learn.org/stable/modules/tree.html#:~:text=Decision%20Trees%20(DTs)%20are%20a,as%20a%20piecewise%20constant%20approximation.
3. https://scikit-learn.org/stable/supervised_learning.html#supervised-learning
4. https://www.mygreatlearning.com/blog/what-is-ridge-regression/#:~:text=Ridge%20regression%20is%20a%20model,away%20from%20the%20actual%20values.