Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

A Mini-Project Report On

"Used Car Auction Prices Prediction"

Submitted By

Harshal Chaudhari
PRN: 1132220562

Khushboo Chaudhari
PRN: 1132220968

Rutuja Ghagare
PRN: 1132220564

F.Y. M.Sc. (Data Science and Big Data Analytics)

School of Computer Science
Faculty of Engineering and Technology
Dr. Vishwanath Karad MIT – World Peace University
Pune - 411038
Academic Year 2022-2023
Nov - 2022

Dr. Vishwanath Karad MIT WORLD PEACE UNIVERSITY, PUNE
SCHOOL OF COMPUTER SCIENCE

Certificate

This is to certify that

Harshal Chaudhari
PRN: 1132220562
Khushboo Chaudhari
PRN: 1132220968
Rutuja Ghagare
PRN: 1132220564

Of **M.Sc. (Data Science and Big Data Analytics)** successfully completed his/her Mini-Project in

"Used Car Auction Prices Prediction"

to our satisfaction and submitted the same during the academic year 2021 - 2022 towards the partial fulfilment of degree of **Master of Science in Data Science and Big Data Analytics** of Dr Vishwanath Karad MIT World Peace University under the School of Computer Science, MIT WPU, Pune.

Prof. Dr. Shubhalaxmi Joshi	**Prof. Surabhi Thatte**	**Prof. Surabhi Thatte**
Associate Dean	**Program Head**	**Assistant Professor**
Faculty of Science	**School of Computer Science**	**School of Computer Science**
MITWPU	**MIT WPU**	**MIT WPU**

ACKNOWLEDGEMENT

In the accomplishment of this project, I would like to express my special thanks of gratitude to my teachers **Prof. Project Mentor**, School of Computer Science, Dr. Vishwanath Karad MIT World Peace University whose valuable guidance has been the ones that helped me patch this project and make it full proof success. His/Her suggestions and instructions have served as the major contributor towards the completion of the project.

As we were working in a group, I would like to thank my group members for their fabulous support throughout the completion of the project. We learned a lot of things during this period, as it was hard to work in this time of adversity; we were in touch with each other throughout the period and shared everything which was important from the aspect of our project. As this project was completed by staying at home, I would also like to thank our families for their cooperation and for providing facilities to us.

Harshal Chaudhari
PRN: 1132220562
Khushboo Chaudhari
PRN: 1132220968
Rutuja Ghagare
PRN: 1132220564

Contents

Introduction

Predicting the price of used cars in both an important and interesting problem. In many developed countries, it is common to lease a car rather than buying it outright. A lease is a binding contract between a buyer and a seller.

Predicting the selling price of a car is not a simple task. It is trite knowledge that the value of used cars depends on a number of parameter. The most important ones are usually the odometer, its condition, make (and model), colour, interior, body, transmission etc.

As we can see, the price depends on a large number of parameters. Unfortunately, information about all these parameters are not always available and the buyer must make the decision to purchase at a certain price based on few factors only.

In this project we are predicting how does each parameter affects on the value of selling price.

Domain Name : Used Car Auction Price Prediction.

Motivation :

There can be several motivations for creating a used car price prediction project report. Some of the common reasons are:

1. To help buyers and sellers: Used car price prediction can help buyers and sellers to determine the fair market value of a car, which can make the buying and selling process more efficient and transparent.
2. To improve the car industry: Used car price prediction models can help car manufacturers and dealers to optimize their inventory management, pricing strategies, and marketing campaigns. This information can lead to better profits, reduced inventory costs, and better customer satisfaction.
3. To assist financial institutions: Financial institutions often use car price prediction models to determine the loan amount they can offer for a used car. This information can help them make better lending decisions and reduce their risks.
4. To advance machine learning: Used car price prediction is a challenging task that requires advanced machine learning algorithms and techniques. Developing such models can help advance the field of machine learning and improve our understanding of complex data analysis problems.

Problem Statement:

The goal is to predict how does each of the parameters like odometer, transmission affect the value of selling price of the car using machine learning algorithms.

Literature Survey:

s.no	Paper Title	Publication Year	Author Name	Outcome/Accuracy	Limitations/ Advantages
1.	Car Price Prediction using Machine Learning Techniques	1 Feb 2019	Enis Gegic, Becir Isakovic, Dino Kečo, Zerina Mašetić, Jasmin Kevrić	To build a model for predicting the price of used cars in Bosnia and Herzegovina, we applied three machine learning techniques (Artificial Neural Network, Support Vector Machine and Random Forest). However, the mentioned techniques were applied to work as an ensemble. The data used for the prediction was collected from the web portal autopijaca.ba using web scraper that was written in PHP programming language. The final prediction model was integrated into Java application. Furthermore, the model was evaluated using test data and the accuracy of 87.38% was obtained.	The drawback of the proposed system is that it consumes much more computational resources than single machine learning algorithm
2.	OLD CAR PRICE PREDICTION WITH MACHINE LEARNING	March 2021	Prashant Gajera , Akshay Gondaliya, Jenish Kavathiya	Total five supervised machine learning models where used, and their root mean squared error are, KNN Regressor: 7771.09, Linear Regression: 6846.23, XG Boost: 3980.77, Random Forest: 3702.34 and Decision Tree Regressor: 5590.43. Out of all Random Forest has lowest RMSE, and performed well with highest Rsquared value: 0.93.	The limitation of this research is a smaller number of records of old car. There was a relatively small dataset for making a strong inference because number of observations was only 92386. Gathering more data can yield more robust predictions. Secondly, there could be more features that can be good predictors.
3.	Car Price Prediction Using Machine Learning	May 2019	Ashish Chandak, Prajwal Ganorkar, Shyam	successfully implemented the machine learning	Limited features where used for the preparation of the model because of

			Sharma, Ayushi Bagmar, Soumya Tiwari	algorithmic paradigms using prominent algorithms from libraries in python.	which price for various car might turn out to be wrong.
4.	Used Cars Price Prediction using Supervised Learning Techniques	December 2019	Pattabiraman Venkatasubbu, Mukkesh Ganesh	The prediction error rate of all the models was well under the accepted 5% of error.	further analysis, the mean error of the regression tree model was found to be more than the mean error rate of the multiple regression and lasso regression models. Even though for some seeds the regression tree has better accuracy, its error rates are higher for the rest. This has been confirmed by performing an ANOVA.
5.	Used Car Price Prediction Using Regression Analysis	2017	S.S.Ghodake,P.S. Despande	The study used dataset of used car prices and car features to develop a regression model to predict car prices. The authors used various regression techniques, including multiple linear regression, polynomial regression, and support vector regression, and found that the support vector regression model performed the best, with an accuracy rate of 86%.	The study is based on a limited sample size of used cars, and the data was collected from a single source, which may not be representative of the entire used car market.
6.	CS 229 Project Report: Predicting Used Car Prices		Kshitij Kumbar, Pranav Gadre, Varun Nayak	The study used different models for the prediction of the used car price and the for the conversion of the non-numeric value to numeric values use of technology of one-hot vector was made use of.	The time taken by each of the model for the process was very high

Solution Design

Solution Approach:

Predicting auction prices for used cars can be a challenging task, as it depends on a variety of factors such as the odometer, its condition, make (and model), colour, interior, body, transmission of the vehicle. However, a few approaches that can be used for predicting used car auction prices are:

Regression Models: One of the most common approaches is to use regression models like Linear regression, Decision Tree, Random Forest. In these models, the features of the car such as the odometer, its condition, make (and model), colour, interior, body, transmission, etc. are used to predict the auction price of the car.

Machine Learning: Another approach is to use machine learning algorithms such as random forest, gradient boosting. In these models, the features of the car are used to train a model that can predict the auction price of the car.

Data Preprocessing: Before applying any machine learning algorithm, it is essential to preprocess the data. This can involve tasks such as data cleaning, normalization, feature selection, and feature engineering. This helps in improving the accuracy of the prediction.

Comparative Analysis: It is also helpful to compare the auction prices of similar cars that have been sold in the past. This can provide insights into the factors that affect the auction price of a car.

Technology Stack:

Encoding:

Encoding refers to the process of transforming data from its raw form into a format that can be effectively used by algorithms for training and prediction.

This transformation is necessary because machine learning algorithms generally require numerical data as input, whereas many real-world datasets contain a mix of categorical, text, and continuous variables.

1) One-hot encoding: This is a technique used to represent categorical variables as binary vectors. Each category is assigned a unique integer value, and then converted into a binary vector where each element represents the presence or absence of a category.
2) Label Encoder: **Label Encoding** refers to converting the labels into a numeric form so as to convert them into the machine-readable form. Machine learning algorithms can then decide in a better way how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.

Scalars: Scalars are often used in machine learning for various purposes, such as representing the output of a neural network or the value of a loss function. For example, the output of a binary classification model could be a scalar value between 0 and 1, representing the probability that the input belongs to the positive class.

Scalars can be manipulated using mathematical operations, such as addition, subtraction, multiplication, and division, just like any other numerical value. In Python, scalars can be represented using basic data types such as integers or floating-point numbers.

The Three scalars which will be made use of is:

1) StandardScalar: **StandardScaler** follows **Standard Normal Distribution (SND).** Therefore, it makes *mean = 0* and scales the data to unit variance.

2) MinMaxScaler: **MinMaxScaler** scales all the data features in the range *[0, 1]* or else in the range *[-1, 1]* if there are negative values in the dataset. This scaling compresses all the inliers in the narrow range *[0, 0.005]*.

3) RobustScaler: By using **RobustScaler()**, we can remove the outliers and then use either StandardScaler or MinMaxScaler for preprocessing the dataset.

Some models that we are going to used for the prediction of the car prices are:

1) K Neighbour's Regressor
2) Decision Tree Regressor
3) Random Forest Regressor
4) Extra Trees Regressor
5) Gradient Boosting Regressor
6) LGBM Regressor
7) XG Boost Regressor
8) Extra Tree Regressor

Some of the Python models that we are going to use are:

1) Numpy
2) Pandas
3) Sklearn
4) MatPlotLib
5) Seaborn

Design Model:

Designing a model for used car price prediction involves several steps, including data collection, data preprocessing, feature selection and engineering, model selection, training and evaluation, and deployment.

Here is a high-level overview of the steps involved in designing a model for used car price prediction:

1. Data Collection: Collect data on used car auctions from reliable sources such as auction websites or car dealerships. This data should include features such as the make and model of the car, mileage, age, location, and condition of the vehicle.
2. Data Preprocessing: Preprocess the data to remove any missing values or outliers. This may involve data cleaning, normalization, and feature scaling.
3. Feature Selection and Engineering: Select the most relevant features for the model and engineer new features if necessary. For example, you can calculate the average annual mileage of a car by dividing the total mileage by the age of the car.
4. Model Selection: Choose a suitable model for the prediction task. Some popular models for used car price prediction include linear regression, decision trees, random forests.
5. Training and Evaluation: Train the selected model on a portion of the data and evaluate its performance on a separate testing dataset. Evaluate the model using appropriate metrics such as mean squared error, mean absolute error, or R-squared.
6. Hyperparameter Tuning: Fine-tune the hyperparameters of the model to improve its performance on the testing dataset. This may involve techniques such as cross-validation or grid search.
7. Deployment: Once the model is trained and validated, deploy it to a production environment. This may involve integrating the model with a web application or API so that users can input the relevant features and receive a prediction of the used car auction price.

Solution Implementation and Results

Obtaining Data:

We have collected data from online source.
i.e https://www.kaggle.com/datasets/tunguz/used-car-auction-prices

Total Observations: 558838
EXPLORATORY DATA ANALYSIS
Condition:
Below is the density graph of the continuous column condition, from the graph we can that the condition column is not following any specific distribution accurately and is also not skewed in any direction, this graph tells us that most of the car are condition was either less than 2.3 or greater than 3 while car having condition between 2 and 3 were less.

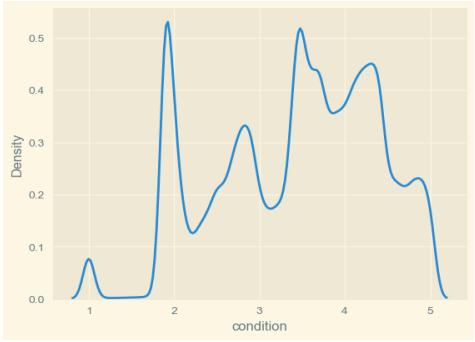

Figure 1 Density plot of condition of the cars.

As this graph is not following any distribution, it should be converted into a normalized or standardized form while building up the model.

Selling Price:
Below is the graphical representation of the continuous variable selling price,
In the below shown density plot we see that graph is positively skewed i.e. most of the car that were sold were sold at a price of less then <40000 dollars.

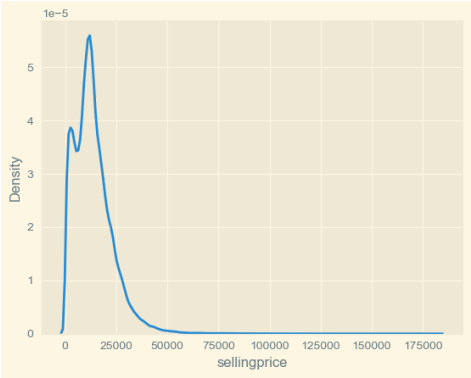

Figure 2 Selling price of the cars.

Below is the density plot of the continuous numerical column odometer,in the odometer density function we see that the density plot is positively skewed stating that most of the car were driven less 400000 miles .

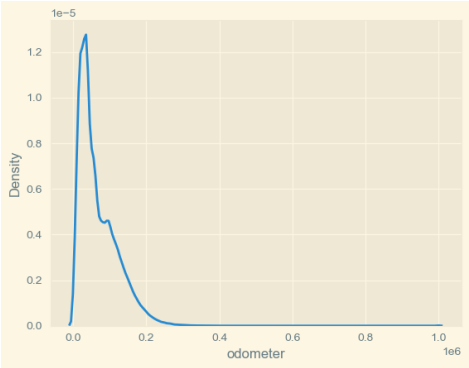

Figure 3 Density plot of the Odometer.

As the odometer density plot show that the odometer column is positively skewed, thus while building up the predictive model it will add some biasness to the model thus to overcome this problem scaling techniques such as normalization and standardization will be made use of.

Models:

Below is the bar chart representation of the categorical column model contain only the top 10 occurring model(values) as representing all the made the representation difficult to understand.In the bar chart we can clearly see that the most occuring model was altima followed by f-150,fusion,camry,escape,focus,accord,3-series,grandcaravan and impala.

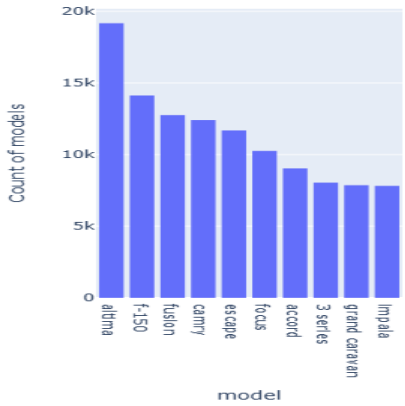

Figure 4 Bar chart of top 10 model(values).

EDA of expensive car:

Dataset named as expensive car was made containing all the cars that were sold above the price of 100000 dollars.

Most of the cars that were sold at a high cost were manufactured after the year 2007, also the cars that were sold at a high cost were made form a renowned brands such as ferrari,bentley,audi etc.

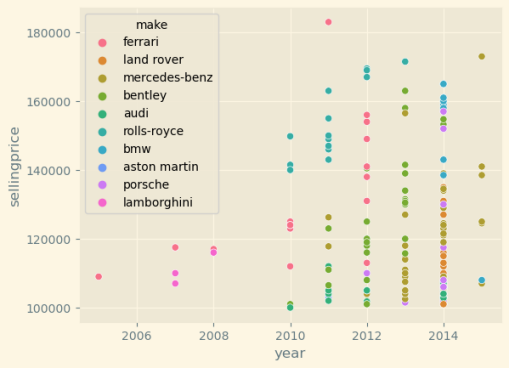

Figure 5: Year vs selling Price of expensive Cars.

Car that was sold at a very high cost were also in a good condition as depicted in the scatter plot figure 6.

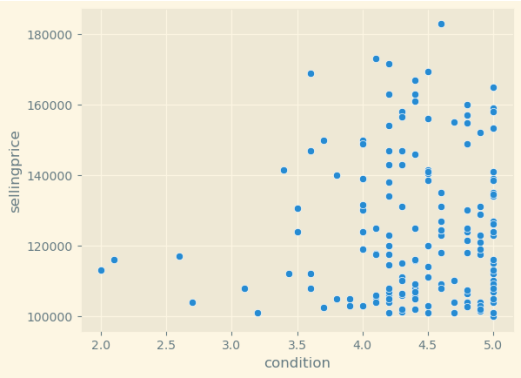

Figure 6: Condition vs selling Price.

Odometer of such types of cars were also less than 40000 miles driven shown in figure 7.

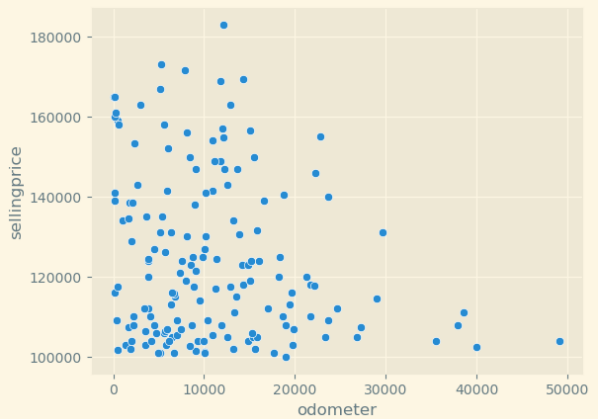

Figure 7: Odometer vs selling Price of expensive cars.

Cheap Sold Cars:

Dataset of cars sold at a price of less than 1000 dollars was created and eda for the same dataset was performed

Odometer of cars that were sold at a cheap price is high, shown in figure 8.

The cars that were sold at cheap price ,there manufacturing dates were spread 1995 to 2010.shown in the figure 9.

The condition of the cars sold at lower price was also highly spread shown in figure 10.

This three graph tells us that any one parameter was not highly able to explain the reason for there cheap price at thus all the parameter were checked while buying such cars.

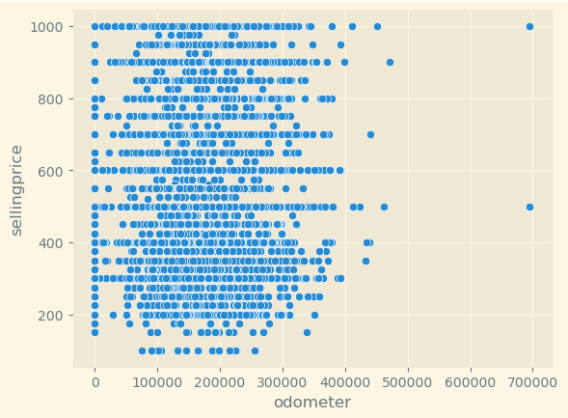

Figure 8: Odometer vs Selling Price of cheap Sold cars

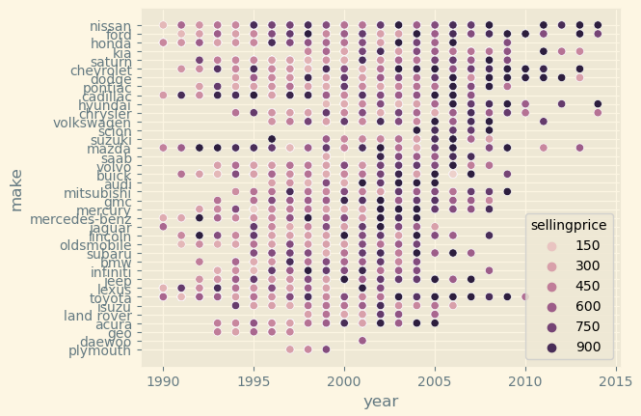

Figure 9: year vs make of the cars

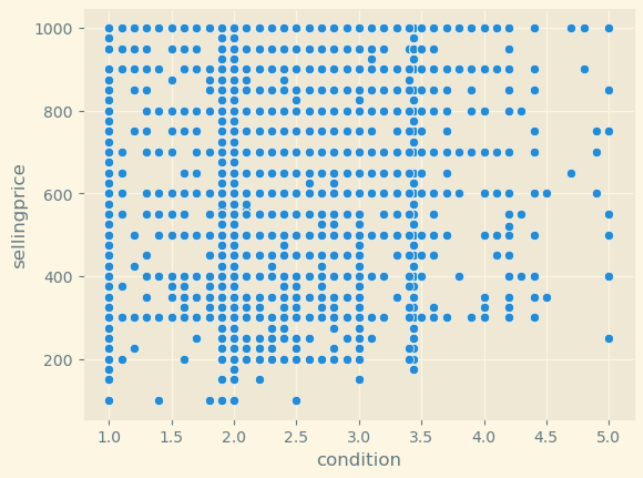

Figure 10: condition vs selling Price of the cars.

Data Pre-Processing

Note-While loading the data inside the python complier we faced an issue thus to resolve those issue initial preprocessing was done on excel sheet

In the Excel sheet we arranged all data that was mismatches inside the column properly.

There are two types of Columns inside the data

1)Categorical

2) Numerical

These two types of data need to be handled in two different ways, Thus

1) Categorical Columns

   In the handling of the categorical columns, we found that some of the same values was depicted in capital as well as small letters. Thus, to overcome this problem all the values were reduced into lower case.

   i) VIN-A VIN, or vehicle identification number, is a unique identifying code given to a vehicle when it's manufactured. Thus, vin cannot be duplicate, while checking the same inside the data we found some duplicity, this duplicity was removed and then the column VIN was also removed from the dataset.

   ii) Model, Make, Trim, Body, Color, Interior-all the null values where were dropped as adding up new values were too difficult as the selling price was different varying in large range for all the null values.

2) Numerical columns

1)BoxPlot

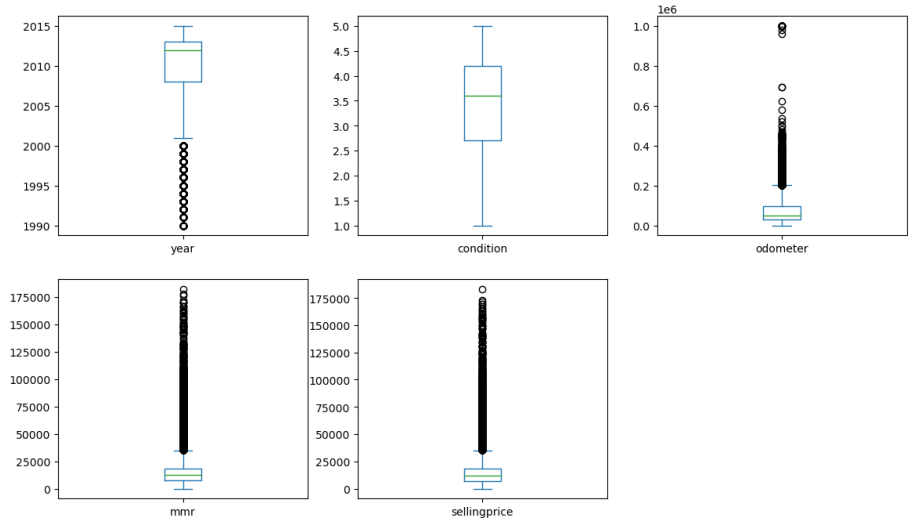

Figure 11:Box plot of numerical features

i) Odometer -Initially all the null values of the odometer were filled with the median values of the odometer column itself. In the box plot we can clearly see that there are some outliers which needs to be check,while checking the outliers we found that there are several value above 900000 mile, a car with such a driven history is definitely less like to be sold thus a second hand car dealer will also not purchase such type of car, thus this values were replaced by the 99 percentile of the odometer column value.

ii) MMR-(MMR or Manheim Market Report (MMR) is the premier tool for wholesale vehicle valuations, searching millions of transactions for industry-average pricing.).Thus, While Implementing the model this column was removed.

iii) Selling Price-Single value of the selling price is further away from the other values in the boxplot graph, while checking this outlier we found that the selling price of this outlier was 230000 while the MMR value was 22800Thus, this value was replaced with the median of the selling price. While the selling price of the cars as 1 dollar was also replaced with the median.

Feature Engineering

1) Label Encoding: **Label Encoding** refers to converting the labels into a numeric form so as to convert them into the machine-readable form. Machine learning algorithms can then decide in a better way how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.

As the dataset which we are making use of for the analysis purpose consists of categorical column more than the numerical columns ,thus the label encoding become a very necessary part for our dataset.

All the categorical values were encoded to numeric form using sklearn module preprocessing.

Feature Selection:

	F_value	p_value
year	281136.032190	0
make	2244.701098	0
model	15.919135	6.611574e-05
trim	494.578028	1.611999e-109
body	6302.874231	0
transmission	1216.976663	2.486399e-266
state	38.141341	6.584803e-10
condition	209693.831749	0
odometer	287098.498082	0
color	754.932680	4.429914e-166
interior	9304.191804	0
seller	271.732944	4.923494e-61

F-value:

The F_value measures the strength of the linear relationship between a feature and the traget variable.A higher “F_value” indicates a stronger linear relationship between the feautre and the target variable.It is a ratio two variance :the variance between the means of the groups formed by dividing the samples based on the feature’s values and the variance within the groups.

p_value:

The p_value measures the probability of observing a test statistic as extreme as the one computed from the data,assuming the null hypothesis is true.A smaller ‘p-value’ indicates that the relationship between the featues and the traget varibale is more statically significant.

We made use of the f_regression to find the f_value and calculate the p_value,it was found that the model and state was less significant for prediction of the values.

MODELLING
Modeling Overview

Figure 12: Overview of the Modelling procedure

The overall procedure followed to obtain the Final Model was data preprocessing ,EDA ,feature engineering ,feature selection ,model fitting and evaluation. The provided data is first cleaned and transformed using Feature Engineering. We then split the data into Train set and Test set (for Model Evaluation). Using RMSLE as our evaluation metric, we compare various models and select the regression algorithm based on the lowest RMSLE on the Test data. The final model used for submission is then obtained by again training the selected Regression Algorithm on the entire Input Data set.

Train/Test Split

The method to split the provided data into training and testing data set.We made use of the Train_Test_split module form the Sklearn library in python.Using the train and test data was split into 80 and 20 % respectively.

1.Training Set

i) This set is used to train various model and obtain the best set of hyperparameters for these models. We use RandomSearchCV to tune the hyperparameters using this training set.

1. Test Set

i) This is used to evaluate all our models. The model with the best test score is finally chosen for submission.

Regression Algorithms Summary

Below is a list of models tried out and a brief description of the algorithm.

Category	Modeling Algorithm	Details
Linear	Linear Regression	Single model for prediction of the used car prices. Label Encoding Features
Non-Linear	Decision Tree	Single model for prediction of the used car prices. Label Encoding Features
Ensemble	Random Forest	Single Model for prediction of the used car prices. Label Encoding Features
Ensemble	Gradient Boost	Single Model for prediction of the used car prices. Label Encoding Features
Ensemble	LGBM	Single Model for prediction of the used car prices. Label Encoding Features
Ensemble	Extra Trees	Single Model for prediction of the used car prices. Label Encoding Features
Ensemble	XGBoost	Single Model for prediction of the used car prices. Label Encoding Features

Boosting:

Boosting is a widely used ensemble learning technique in machine learning that combines multiple weak models to create a strong model. The idea behind boosting is to iteratively train models on the same data and give more weight to misclassified samples in each iteration, thereby focusing the model's attention on the difficult examples. Boosting can be used with any learning algorithm that can handle weighted training data, such as decision trees, neural networks, and support vector machines. Some popular boosting algorithms include AdaBoost, Gradient Boosting, and XGBoost, which have been shown to be highly effective in many machine learning tasks, including classification, regression, and ranking. Boosting has several advantages over other ensemble techniques, including its ability to

handle complex data and its ability to automatically tune model parameters. However, it can also be sensitive to noisy data and overfitting, and requires careful tuning of its hyperparameters to achieve optimal performance.

Bagging:

Bagging, short for Bootstrap Aggregating, is a powerful technique in machine learning that is used to improve the accuracy and stability of statistical learning algorithms. It works by training multiple models on different random subsets of the original training data and then combining the predictions of all these models to obtain a final prediction. This approach helps to reduce the variance in the model by introducing diversity, which can lead to better generalization and lower prediction errors. Bagging is widely used in many areas of machine learning, including classification, regression, and clustering, and has been shown to be highly effective in improving the performance of many different types of models.

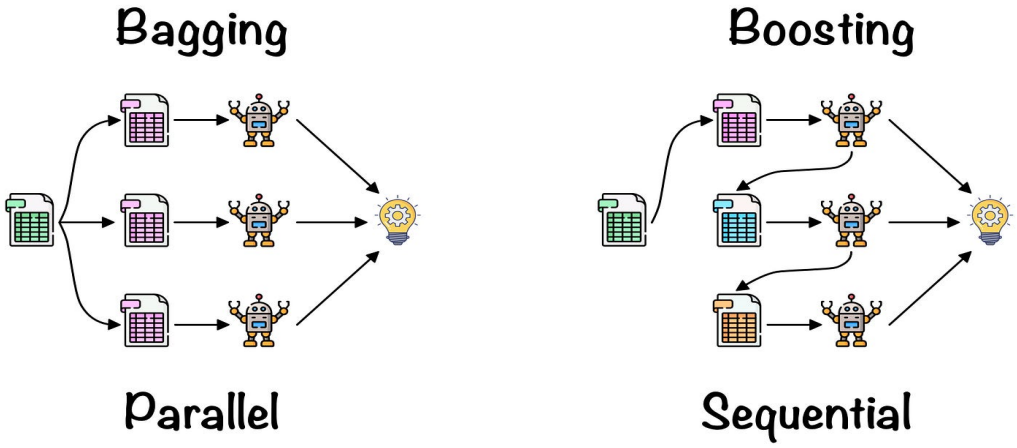

Figure 13:Bagging and Boosting

Linear Regression:

Linear Regression is a fundamental machine learning algorithm used for predicting a continuous variable. It works by fitting a linear function to the input data, where the function maps the input features to the target variable. Linear Regression assumes that there is a linear relationship between the input features and the target variable, and tries to find the coefficients of the linear function that minimize the

residual error between the predicted and actual values of the target variable. Linear Regression is widely used in various fields, such as finance, marketing, and social sciences, due to its simplicity and interpretability. However, Linear Regression can be limited by its assumption of linearity, and may not perform well when the relationship between the input features and the target variable is non-linear.

Decision Tree Regression:

Decision Tree Regression is a machine learning algorithm used for predicting continuous variables. The algorithm works by recursively partitioning the input data based on the values of the features, and then fitting a regression model to each resulting subset of the data. The regression model is typically a linear or polynomial function of the features, which is used to predict the target variable for new input data. Decision Tree Regression is known for its simplicity and interpretability, as it can be easily visualized and understood by non-experts. It is also computationally efficient, and can handle both numerical and categorical data.

Random Forest Regression:

Random Forest Regression is a machine learning technique that combines multiple decision trees to create a strong regression model that can predict the target variable. Random Forest Regression works by creating a large number of decision trees, each trained on a random subset of the features and training data, and then averaging their predictions to obtain a final prediction. Random Forest Regression is known for its ability to handle high-dimensional data and noisy data, as well as its ability to capture non-linear relationships between the features and the target variable. It has been shown to be highly effective in many machine learning tasks, including regression, classification, and feature selection. Random Forest

Regression also incorporates several advanced features, such as out-of-bag error estimation and variable importance ranking, which can help to improve model accuracy and interpretability.

Gradient Boosting Regression:

Gradient Boosting Regression is a powerful machine learning technique that combines multiple weak regression models to create a strong model that can predict the target variable. Gradient Boosting Regression works by iteratively adding weak regression models to the ensemble, with each model being trained to minimize the error of the previous models. The models are combined by taking a weighted average of their predictions. Gradient Boosting Regression is based on the gradient descent optimization algorithm, which updates the model parameters in the direction of the negative gradient of the loss function. By using this approach, Gradient Boosting Regression can effectively handle complex non-linear relationships between the features and the target variable and has been shown to be highly effective in many machine learning tasks, including regression, classification, and ranking. Some popular Gradient Boosting Regression algorithms include XGBoost, LightGBM, and CatBoost, which have become popular choices in many machine learning applications.

LightGBM Regression:

LightGBM (LGBM) regression is a powerful and efficient gradient boosting framework used for machine learning tasks such as regression, classification, and ranking. LGBM regression works by creating an ensemble of weak regression models, typically decision trees, and then training them sequentially to minimize the loss function. LGBM is known for its speed and scalability, as it can handle large datasets with high-dimensional features in a relatively short amount of time. LGBM regression has several advanced features such as early stopping, regularization, and parallel processing, which make it a popular choice for many machine learning practitioners. One of the key features of LGBM regression is its

ability to handle categorical features, which are commonly found in real-world datasets. LGBM regression uses a combination of histogram-based and leaf-wise algorithms to optimize the split points in decision trees, which helps to reduce overfitting and improve accuracy.

Extra Tree Regression:
Extra Tree Regression is a powerful machine learning technique that is used for predicting continuous numerical values. It is a type of decision tree ensemble algorithm, similar to Random Forest Regression, but with some key differences. In Extra Tree Regression, the algorithm builds a large number of decision trees by randomly selecting subsets of features and splitting points, and then averages their predictions to obtain a final prediction. The algorithm also incorporates randomness into the selection of splitting points, which helps to reduce overfitting and improve generalization performance. Extra Tree Regression has several advantages over other regression techniques, including its ability to handle high-dimensional data and noisy data, its computational efficiency, and its flexibility in handling different types of data.

XGBoost Regression:
XGBoost Regression is a powerful technique for predicting continuous numerical values, also known as regression problems. XGBoost stands for Extreme Gradient Boosting, and it is a widely used ensemble learning algorithm that combines multiple weak learners to create a strong model. In XGBoost regression, the algorithm builds a series of decision trees by recursively partitioning the training data into smaller and smaller subsets, and then combines their predictions to obtain a final prediction. The algorithm also incorporates regularization techniques to prevent overfitting and improve generalization performance. XGBoost has several advantages over other regression techniques, including its ability to handle missing data, its scalability to large datasets, and its ability to capture non-linear relationships between the features and the target variable.

Evaluation Metric-RMSE

Root mean square error (RMSE) is a popular metric used to measure the difference between predicted and actual values in regression analysis. It is the square root of the average squared difference between the predicted and actual values.
RMSE provides a measure of the spread of the residuals (the difference between the predicted and actual values) around the regression line. A smaller RMSE value indicates that the predicted values are closer to the actual values, while a larger RMSE value indicates a larger spread of residuals and therefore poorer performance of the model.
RMSE is commonly used in machine learning and data science to evaluate the accuracy of regression models, and it can be useful in comparing the performance of different models. However, it should be used in conjunction with other evaluation metrics to get a more comprehensive view of model performance.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

where,
predicted_i = The predicted value for the ith observation.
actual_i = The observed(actual) value for the ith observation
N = Total number of observations.

For the evaluation of the model which we will be building for our dataset,we have made use of this evaluation metrics.
Result:

Linear Regression:

For Linear Regression, we developed a single model for prediction of the used car price, we also made use of the Label Encoding for converting the categorical features.

R^2	0.449884619572
Adjusted R^2	0.44498230917

RMSE for Test dataset	**7178.477909633718**
RMSE for Train Dataset	**7141.9754136**
Total Train Test time	**0.368435144442**

The RMSE obtained for this model on the test dataset is 7178.477909633718 and RMSE obtained for this model on the train dataset is 7141.9754136, as they are almost same, the model is definitely not an overfit model.
Below is the graphical representation of Predicted vs actual values and the fitted Values vs the residual values. From the figure 15 we can clear see that the Residual value follows a pattern and the from the figure 14 we can also see that there are some values which are predicted as negative.As the dataset contained some of the Categorical values which were label encoded ,this values would be the reason for such error.

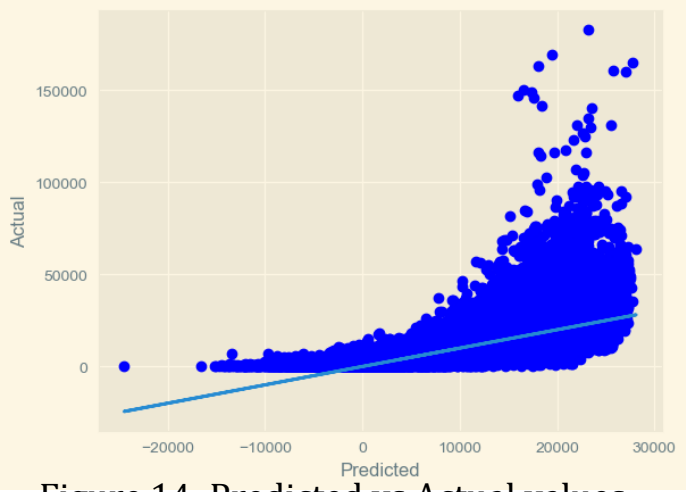

Figure 14: Predicted vs Actual values

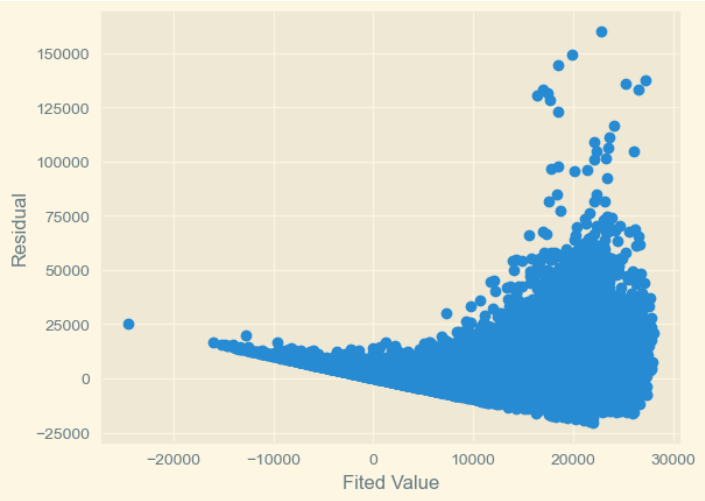

Figure 15: Fitted Values vs residual values

Decision Tree:

Since Decision tree Follows a non-linear approach for building model, we expect it to handle categorical features like make,model,body gracefully.In decision tree we use a single model for prediction of the car prices were all the categorical features were encoded using label encoding.

R^2	**0.90154779339**
Adjusted R^2	**0.90153678196**
RMSE for Test dataset	**3036.814062254**
RMSE for Train Dataset	**0.0**
Total Train Test time	**3.48999**

The RMSE for test dataset is 3036.81406 and the RMSE for 0.0 thus decision tree overvall showed some signs of overfit but while making use of cross validation the mean value of cross validation was around 0.9015477 depicting no such issues. Thus overall the accuracy of the decision was around 0.9015477.

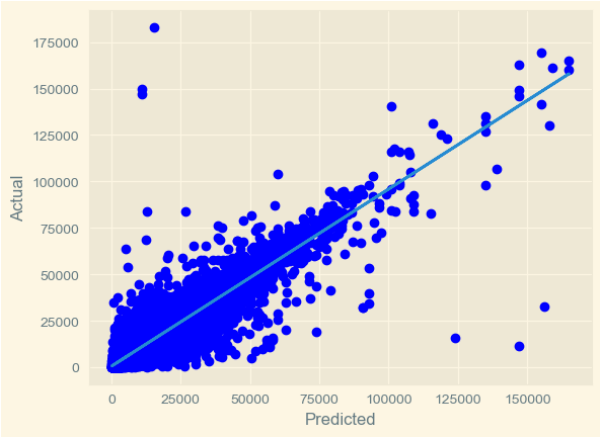

Figure 16: Predicted vs Actual Value

Figure 17: Fitted value vs Residual

Random Forest:

Since random forest is an ensemble technique making use of decision trees for predicting the value,we expect it to also handle categorical features gracefully. In random forest ,we use a single model technique for predict of the used car prices,were the categorical features were encoded using label encoding.

R^2	**0.950238**
Adjusted R^2	**0.950232**
RMSE for Test dataset	**2159.007444**
RMSE for Train Dataset	**841.4594**
Total	**109.09216737747**

Train Test time	

The accuracy for the model is 0.950238
The RMSE for the test dataset was found 2159.007 and the RMSE for the train dataset was found 841.459 after making use of cross-validation the mean of the accuracy was 0.950564 depicting that the model is not an overfitting. As the random forest makes use of more than one decision tree, the accuracy of the model was more than the decision tree.

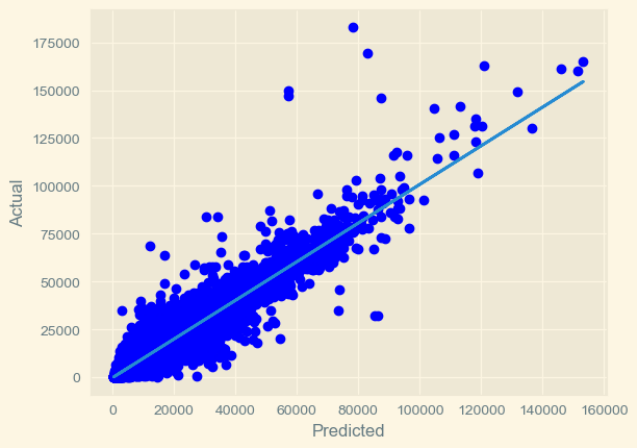

Figure18: Predicted vs Actual Values

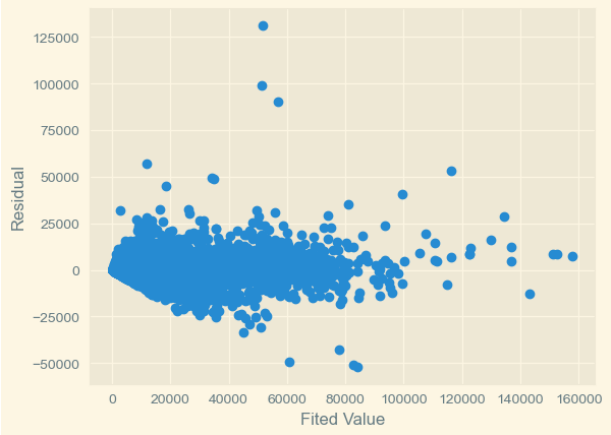

Figure 19: Fitted Values vs Residual

Gradient Boosting:
base model for gradient boosting depends on the nature of the problem at hand, as well as the trade-offs between accuracy, interpretability, and computational efficiency. In general, decision tree models are popular choices for gradient boosting due to their flexibility and ability to capture non-linear relationships between features and the target variable.

R^2	0.723692951

Adjusted R^2	0.723662948932
RMSE for Test dataset	5087.465256149773
RMSE for Train Dataset	4997.717185633473
Total Train Test time	71.8997099399

The RMSE value for the test dataset of the model is 5087.46525614 and the RMSE value for the train dataset of the model is 4997.717185. The total train test time of the model is 71.8997099399. After cross validation the mean accuracy of the model was found to be 0.729322229144. As the accuracy of the model was around 72% to increase the accuracy of the model hyperparameter tunning was done.

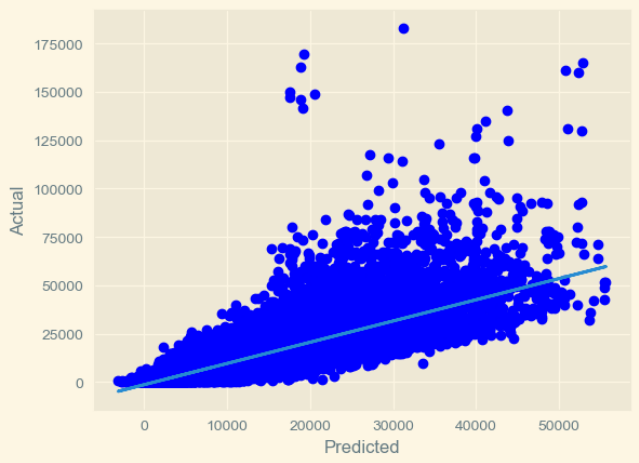

Figure 20: Predicted vs Actual Values

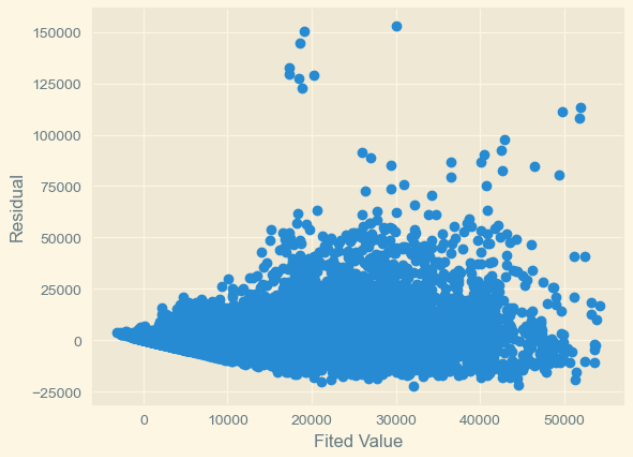

Figure 21: Fitted Values vs Residual

LGBM:

R^2	0.8958796
Adjusted R^2	0.895868038
RMSE for Test dataset	3123.008833271517
RMSE for Train Dataset	2998.313668324
Total Train Test time	2.12155805

The accuracy of the model was 0.8958796%
The RMSE for the test dataset is 3123.0088332715 and the RMSE for the train dataset is 2998.313668324 saying that the model was fitting properly and also after cross validation the mean of the accuracy was 0.896996513.

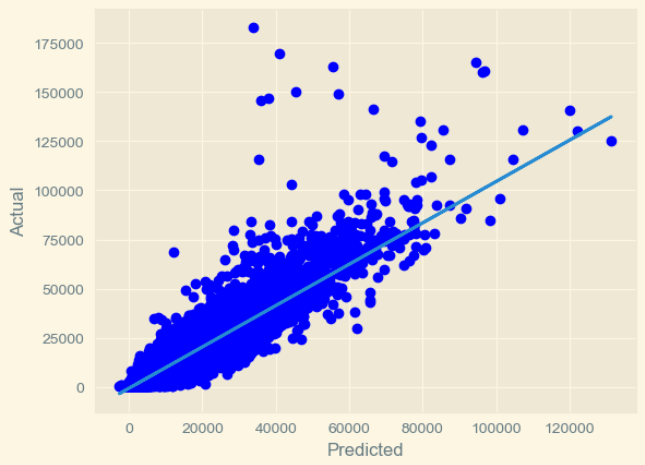

Figure 22: Predicted vs Actual Values

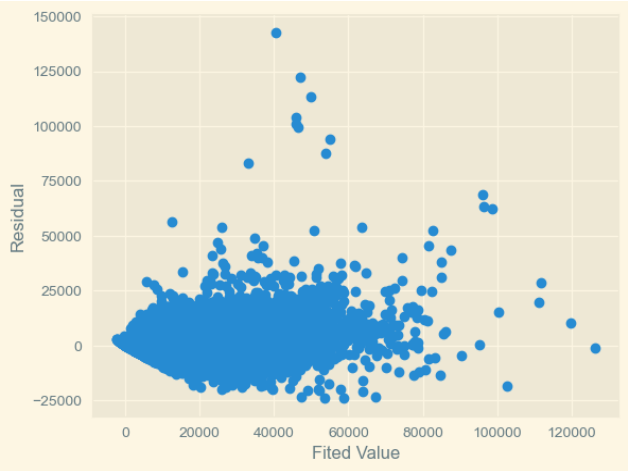

Figure 23: Fitted Values vs Residual

Extra Tree:

R^2	**0.944640192**
Adjusted R^2	**0.9446340011**
RMSE for Test dataset	**2277.2083032193**
RMSE for Train Dataset	**2.962170553879**
Total Train Test time	**153.4322686195**

The RMSE for the test dataset is 2277.2083032 and the RMSE for the train dataset is 2. 962170553879. The total train and test time taken by the model is 153. 4322686195. After cross validation the mean accuracy of the model was found to be 0.942731763. As the model makes use of multiple decision tree thus the total time taken by the model to test is also high.

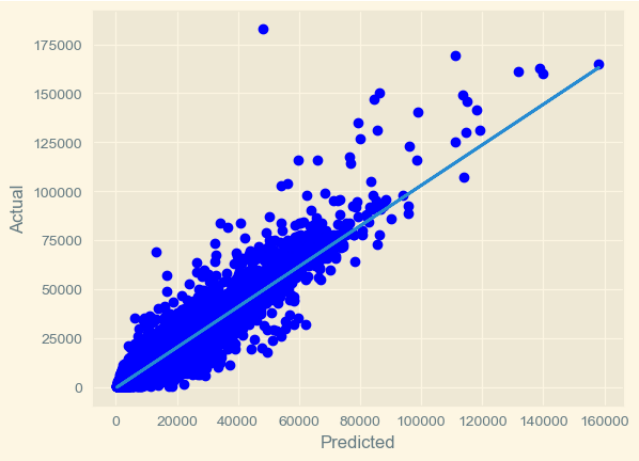

Figure 24: Predicted vs Actual Values

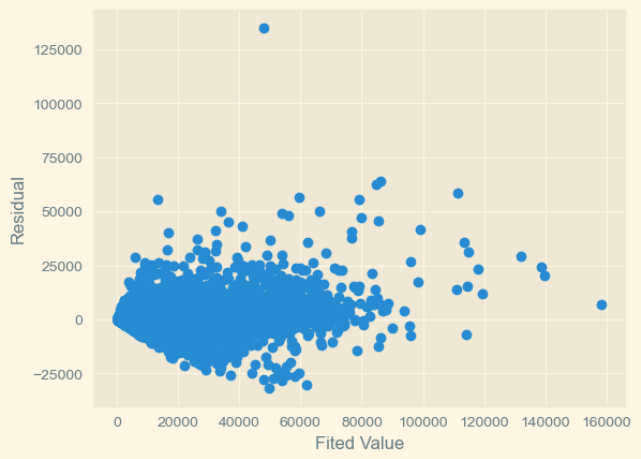

Figure 25: Fitted Values vs Residual

XGBoost:

R^2	**0.935652523**
Adjusted R^2	**0.9356453268**
RMSE for Test dataset	**2455.11157367**
RMSE for Train Dataset	**2286.506309720**
Total Train Test time	**21.26089525222**

The RMSE for the test dataset is 2455.1115736 and the RMSE for the train dataset is 2286.50630972. The total train and test time taken by the model is 21.26089.

After the cross validation the mean accuracy of the model was found to be 0.9364558698.

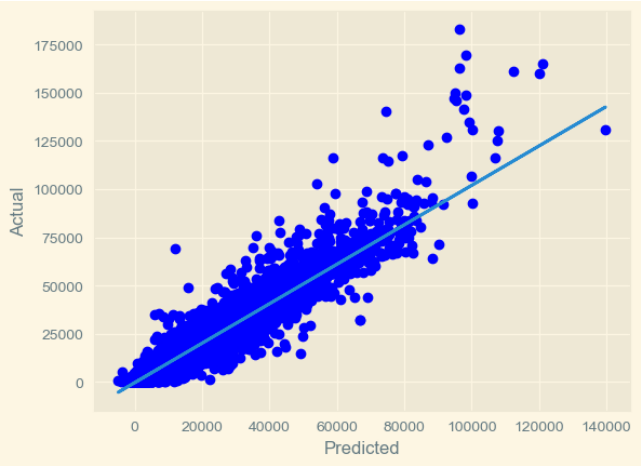

Figure 26: Predicted vs Actual Values

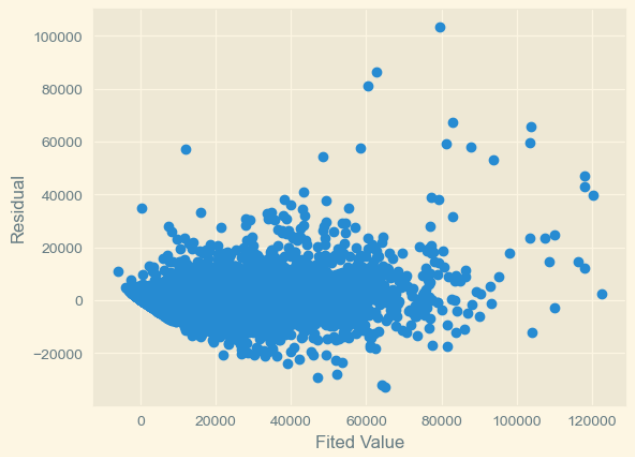

Figure 27: Fitted Values vs Residual

HyperParameter Tunning:

Hyperparameter tuning is an essential process in machine learning model development. Hyperparameters are model parameters that are set before the training process and can significantly affect the model's performance. Tuning these hyperparameters involves selecting the optimal values that lead to the best model performance.

There are various methods for hyperparameter tuning, including grid search, random search, and Bayesian optimization. These methods allow us to search for the best combination of hyperparameters within a given range or distribution.
It's important to note that hyperparameter tuning is an iterative process and requires a lot of computational resources. It involves testing different combinations of hyperparameters, training and evaluating the model, and selecting the optimal combination.
In conclusion, hyperparameter tuning is a critical step in machine learning model development that can significantly improve model performance. It's essential to choose the right hyperparameters and use an appropriate tuning method to ensure the best possible model performance.

We made use of hyperparameter tunning to improve the accuracy of the model (Gradient boosting Regression). As the data is extremely large making use of GridSearchCV was time consuming thus, we made use of the function RandomSearchCV.
The HyperParameters which we tried optimized in GradientBoosting, along with their values are.

```
n_estimators = [150,200,500],
max_depth = [2,4,6],
learning_rate = [0.01,0.1,1],
loss= ['squared_error','huber','quantile']
```

The optimal parameter that was obtained form the RandomSearchCV was N_estimators=500, max_depth=6, loss=huber,learning_rate=0.1
The accuracy obtain is 0.9286.

Conclusion:

Without knowing the specifics of the used car auction price prediction model in question, it is difficult to provide a conclusive statement. However, in general, a good conclusion to draw from a used car auction price prediction model would be:

The model can provide valuable insights into the potential market value of used cars at auction. By analyzing historical data, such as make and model, body ,transmission, condition, odometer and other factors, the model can generate predictions of the likely selling price of a given car. These predictions can be useful to both buyers and sellers, allowing them to make informed decisions about selling strategies.
However, it's important to keep in mind that no model can provide perfect predictions. There are many factors that can influence the final sale price of a used car, such as the specific conditions of the auction and the overall demand for a particular vehicle.

Future Work:

1) In future we can explore the use of other algorithms such as support vector machines to see if they can improve the accuracy of your predictions.
2) We can consider adding more data to your dataset, such as the mileage, the number of previous owners and other features that might impact the price of a used car. The more data you have, the better we model can learn the patterns and make accurate predictions.
3) we can consider incorporating external data sources such as economic indicators, gas prices, or weather data to see if they have any correlation with the auction price of used cars. This can help we build a more robust and accurate model.

References:

https://www.knowledgehut.com/blog/data-science/linear-regression-for-machine-learning
https://www.geeksforgeeks.org/ml-label-encoding-of-datasets-in-python/
TEMJournalFebruary2019_113_118.pdf
(PDF) Used Cars Price Prediction using Supervised Learning Techniques (researchgate.net)
1628083284.pdf (irjmets.com)
(PDF) Car Price Prediction Using Machine Learning (researchgate.net)
26612934.pdf (stanford.edu)