Car Price Prediction

Project Report

Supriya Parandha, Binu Singh, M Samarasimha Reddy

DST-CIMS (Statistics & Computing), Banaras Hindu University, Varanasi, India.

Abstract

Approximately 40 million used vehicles are sold each year. Effective pricing strategies can help any company to efficiently sell its products in a competitive market and making profit. In the automotive sector, pricing analytics play an essential role for both companies and individuals to assess the market price of a vehicle before putting it on sale or buying it. And, the rise of used cars sales is exponentially increasing. Car sellers sometimes take advantage of this scenario by listing unrealistic prices owing to the demand.

Therefore, arises a need for a model that can assign a price for a vehicle by evaluating its features taking the prices of other cars into consideration. In this Notebook, we use supervised learning methods to predict the prices of used cars. The model has been chosen after careful exploratory data analysis to determine the impact of each feature on price.

So, we propose a methodology using Machine Learning models to predict the prices of used cars given the features. The price is estimated based on the number of features as mentioned above.

Keywords

Car price prediction, Classifiers (K-Nearest Neighbor, Decision Trees, Random Forest, Logistic Regression and Support Vector), Machine Learning, Missing Values and Outliers.

Introduction

The prices of new cars in the industry is fixed by the manufacturer with some additional costs incurred by the Government in the form of taxes. So customers buying a new car can be assured of the money they invest to be worthy. But due to the increased price of new cars and the incapability of customers to buy new cars due to the lack of funds, used cars sales are on a global increase. Predicting the prices of used cars is an interesting and much-needed problem to be addressed. Customers can be widely exploited by fixing unrealistic prices for the used cars and many falls into this trap. Therefore, rises an absolute necessity of a used car price prediction system to effectively determine the worthiness of the car using a variety of features. Due to the adverse pricing of cars and the nomadic nature of people in developed countries, the cars are mostly bought on a lease basis, where there is an agreement between the buyer and seller. These cars upon completion of the agreement are resold. So reselling has become an essential part of today's world.

Given the description of used cars, the prediction of used cars is not an easy task. There are a variety of features of a car like the age of the car, its make, the origin of the car (the original country of the manufacturer), its mileage (the number of mildes it has run) and its horsepower. Due to rising fuel prices, fuel economy is also of prime importance. Other factors such as the type of fuel it uses, style, braking system, the volume of its cylinders (measured in cc), acceleration, the number of doors, safety index, size, weight, height, paint color, consumer reviews, prestigious awards won by the car manufacturer.

Other options such as sound system, air conditioner, power steering, cosmic wheels, GPS navigator all may influence the price as well.

Objective

To build a supervised machine learning model for forecasting value of a vehicle based on multiple attributes

The system that is being built must be feature based i.e. feature wise prediction must be possible

Providing graphical comparisons to provide a better view.

Literature review

In this chapter, we discuss various applications and methods which inspired us to build our project. We did a background survey regarding the basic ideas of our project and used those ideas for the collection of information like the technological stack, algorithms, and short comings of our project which led us to build a better project.

CARS24

Cars24 is a web platform where seller can sell their used car. It is an Indian Start-up with a simplified user interface which asks seller parameters like car model, kilometers traveled, year of registration and vehicle type (petrol, diesel)[1]. These allow the web model to run certain algorithms on given parameters and predict the price.

GET VEHICLE PRICE

Get Vehicle Price is an android app which works on similar parameters as of Cars24. This app predicts vehicle prices on various parameter like Fiscal power, horsepower, kilometers traveled. This app uses a machine learning approach to predict the price of a car, bike, electric vehicle and hybrid vehicle. This app can predict the price of any vehicle because of the smartly optimized algorithm.

CARWALE

CarWale app is one of the top-rated car apps in India for new and used car research. It provides accurate on-road prices of cars, genuine user and expert reviews. It can also compare different cars with the car comparison tool. this app also helps you to connect with your nearest car dealers for the best offers available.

CARTRADE

CarTrade is web and Android platform where user can research New Cars in India by exploring Car Prices, Car Specs, Images, Mileage, Reviews, and Car Comparisons. On this app one can Sell Used Car to genuine buyers with ease. One can list their used car for sale along with the details like image, model, and year of purchase and kilometers so that it is displayed to lakhs of interested car buyers in their city. User can read user reviews and expert car reviews with images that help in finalizing a new car buying decision

TECHNOLOGY USED

Python was the major technology used for the implementation of machine learning concepts the reason being that there are numerous inbuilt methods in the form of packaged libraries present in python. Following are prominent libraries/tools we used in our project.

NUMPY

NumPy is a general-purpose array-processing package. it provides a high-performance multidimensional array object and tools for working with these arrays. It is the fundamental package for scientific computing with Python. Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

SCIPY

SciPy is a free and open-source Python library used for scientific computing and technical computing. SciPy contains modules for optimization, linear algebra, integration, interpolation, special functions, FFT, signal and image processing, ODE solvers and other tasks common in science and engineering. SciPy builds on the NumPy array object and is part of the NumPy stack which includes tools like Matplotlib, pandas, and SymPy, and an expanding set of scientific computing libraries. This NumPy stack has similar users to other applications such as MATLAB, GNU Octave, and Scilab. The NumPy stack is also sometimes referred to as the SciPy stack. The SciPy library is currently distributed under the BSD license, and its development is sponsored and supported by an open community of developers. It is also supported by NumFOCUS, a community foundation for supporting reproducible and accessible science.

SCIKIT-LEARN

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. The library is built.

JUPYTER NOTEBOOK

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. It includes data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more. The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. It includes data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

IMPLEMENTATION

We have approached the project in a step by step manner. For this project, we are using the dataset on used car sales which is available on Kaggle. The features available in this dataset are name, year, selling_price, km_driven, fuel, seller_type, transmission, owner in which selling_price is target variable and remaining were independent variable.

So, after viewing the dataset we firstly imported dataset into Notebook (Jupyter notebook) which we preferred for this work. For that we used pandas library function provided by python. As the dataset is Raw data which needs some mining in it to analyse in effective way. So, for that we firstly check the null cells by using python function (.isna().sum()) and then compensating this with overall average of that particular attribute. After that in order to make this data with different format usable for our algorithms, categorical data was converted into separated indicator data, which expands the number of features in this dataset. And then to getting more clarity about data we needed some Visualizing through bar plot, scatterplot, quality correlation matrix, Correlation matrix etc.

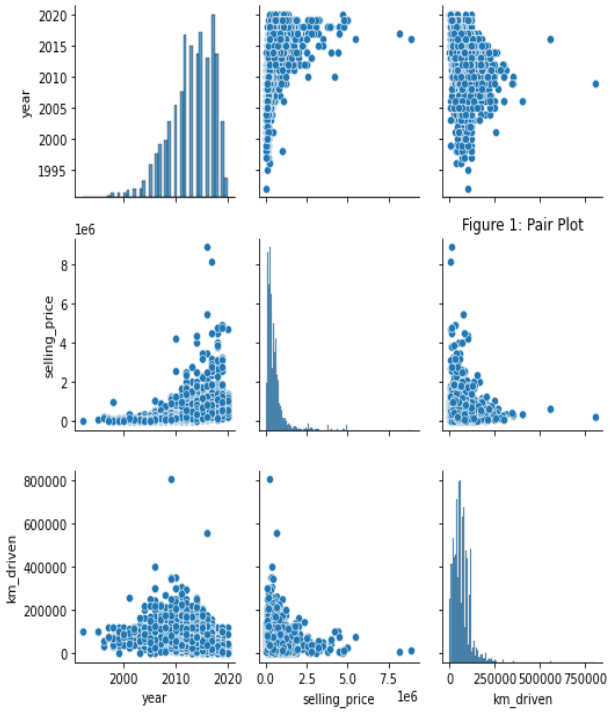
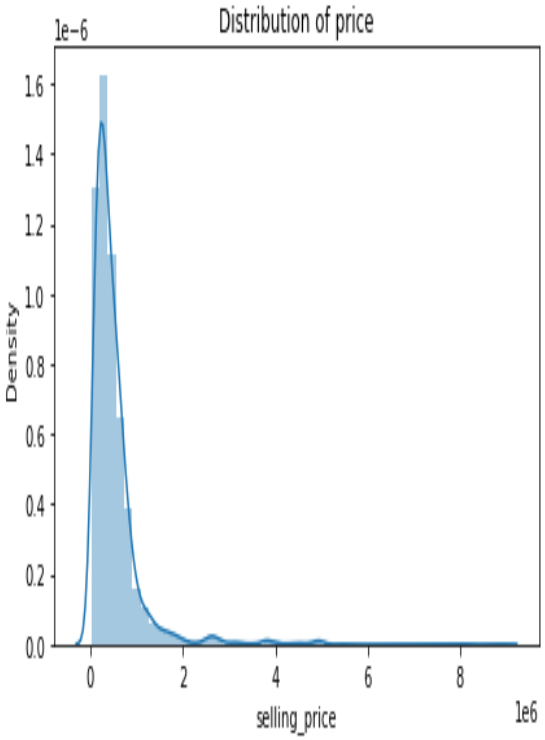
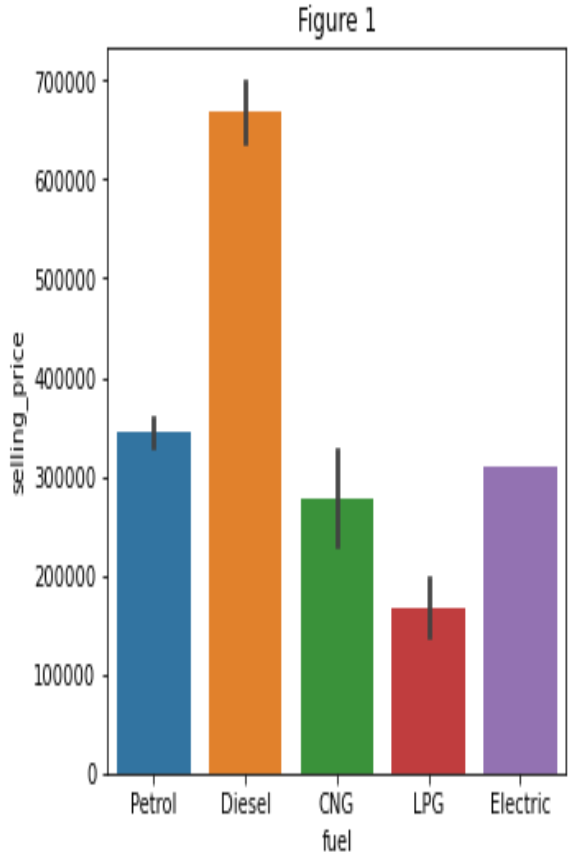


Figure 1: Pair Plot

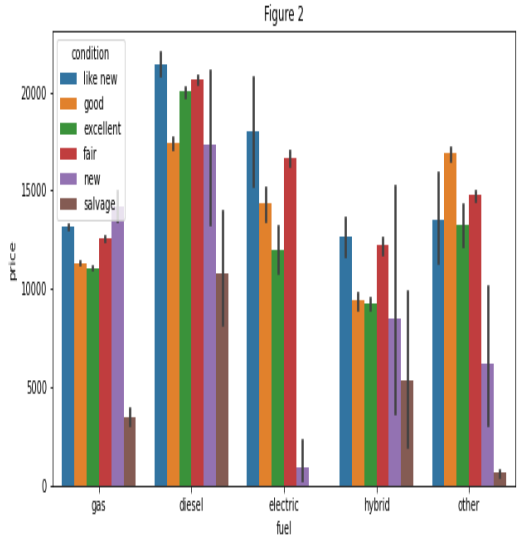
From the pair plot, we can't conclude anything. There is no correlation between the variables.



From the distplot, we can conclude that initially, the price is increasing rapidly but after a particular point, the price starts decreasing.

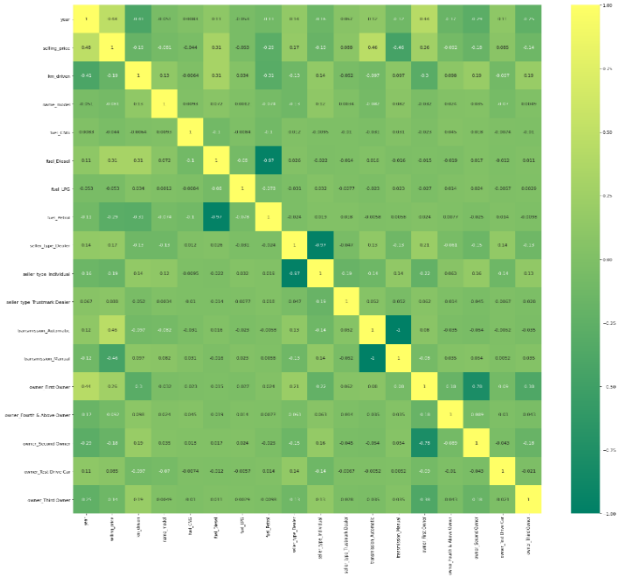


From figure 1, we analyze that the car price of the diesel variant is high then the price of the electric variant comes. Hybrid variant cars have the lowest price.



From figure 2, we analyze that the car price of the respective fuel also depends upon the condition of the car.

Above and below pair are gives pair plot before and after encoding categorical features.



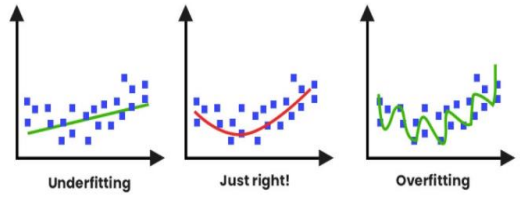
Training and Testing

Next we had splitted our dataset into training, testing data with a 70:30 split ratio. The splitting was done by picking at random which results in a balance between the training data and testing data amongst the whole dataset. This is done to avoid overfitting and enhance generalization.

Models and Results

1.Linear regression

For regression models, we try to solve the following problem: given a processed list of features for a car, we would like to predict its potential sale price. Linear regression is the first step stone in the field of machine learning and is a supervised learning approach that is used to predict a quantitative response from a predictor variable by use of statistical approach. So created variables in our model are year,km_driven,name_model, fuel_cng, diesel, petrol, sellr_type_dealer, individual, trustmarkdealer, trasmissio_automatic, manual, owner_first, second, third, fourth & above. Our linear model was trained on 70% training data and then the test data was predicted. The performance was measured on R^2 of the predicted results and the actual results and our baseline model generated R^2 of 0.5830 or 58.30%.



After using linear regression model as the baseline model, techniques that improve the performance of ordinary linear regression models. The most common techniques are LASSO regularization (L1 Regularization) and Ridge Regularization (L2 Regularization). Simply Regularization is the process of adding information to prevent over-fitting. The over-fitting problem occurs when the error of the model is minimum in the training phase, but the performance of the model with testing data points is poor. That means that the model is not generalized and cannot be used in production.

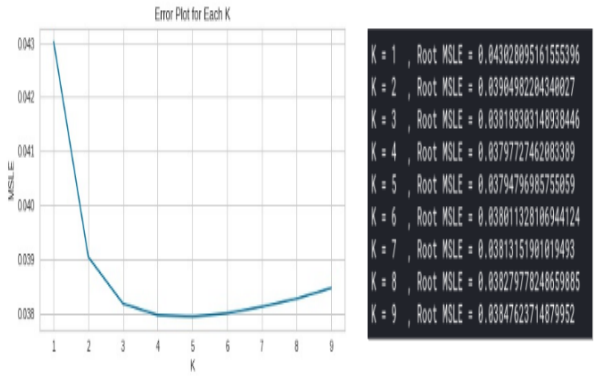
Linear regression with Lasso at alpha=100 and tolerance 0.1 generated R^2 0.5930.

Other than lasso regularizer, we also applied ridge regularizer in our linear regression model, which generate a R^2 of 0.5960. This R^2 is slightly better than our baseline model.

4)KNeighbors Regressor: Regression-based on k-nearest neighbors.

The target is predicted by local interpolation of the targets associated with the nearest neighbours the training set. *k*-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until function evaluation. From the below

figure, for k=5 KNN give the least error. So dataset is trained using n_neighbors=5 and metric='euclidean'.

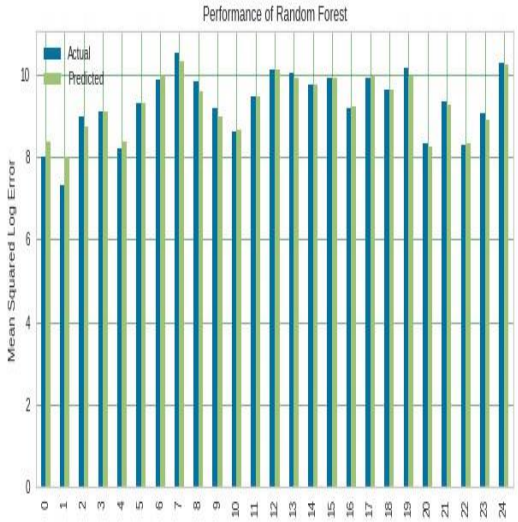


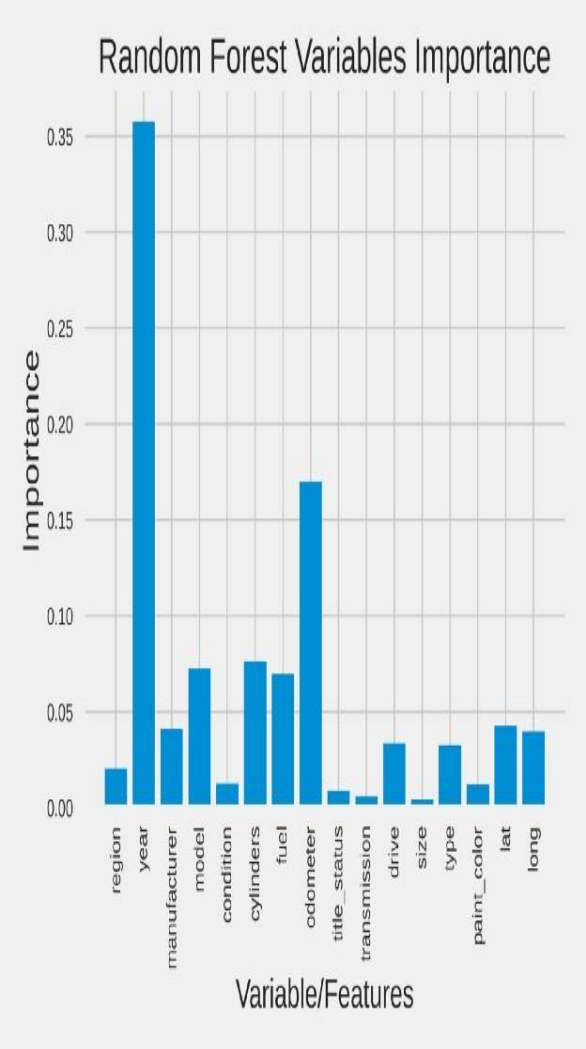
The result of knn obtained was R^2 of 76.4381%, which tells the performance of KNN is better and error is decreasing with increased accuracy**.**

5)Random Forest:

The random forest is a classification algorithm consisting of many decision trees. It uses bagging and feature randomness when building each tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree. ***Read More***

In our model, 180 decisions are created with max_features 0.5





This is the simple bar plot which illustrates that ***year*** is the most important feature of a car and the Odometer variable and then other.
The performance of the Random forest is better and accuracy is 87.59% which is increased by approx. 10% which is good. Since the random forest is using bagging when building each tree so next Bagging Regressor will be performed.

6) Bagging Regressor:

A Bagging regressor is an ensemble meta-estimator that fits base regressors each on random subsets of the original dataset and then aggregates their predictions (either by voting or by averaging) to form a final prediction. Such a meta-estimator can typically be used as a way to reduce the variance of a black-box estimator (e.g., a decision tree), by introducing randomization into its construction procedure and then making an ensemble out of it.

In our model, DecisionTreeRegressor is used as the estimator with max_depth=20 which creates 50 decision trees and the result is R^2 of 76.890%.

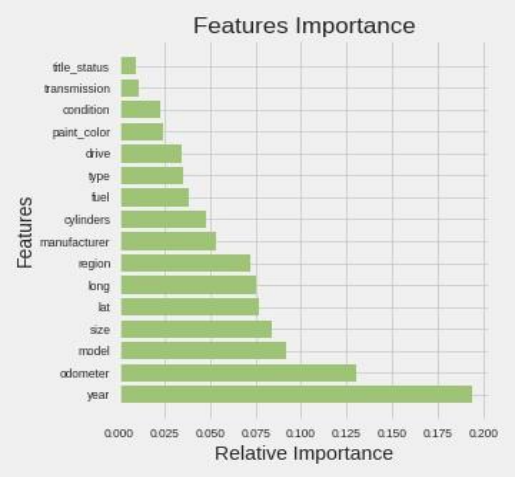
The performance of Random Forest is much better than Bagging regressor

The key difference between Random forest and Bagging: The fundamental difference is that in **Random forests**, only a subset of features are selected at random out of the total **and** the best split feature from the subset is used to split each node in a tree, unlike in **bagging** where all features are considered for splitting a node.

7) Adaboost regressor:

AdaBoost can be used to boost the performance of any machine learning algorithm.

Adaboost helps you combine multiple “weak classifiers” into a single “strong classifier”.

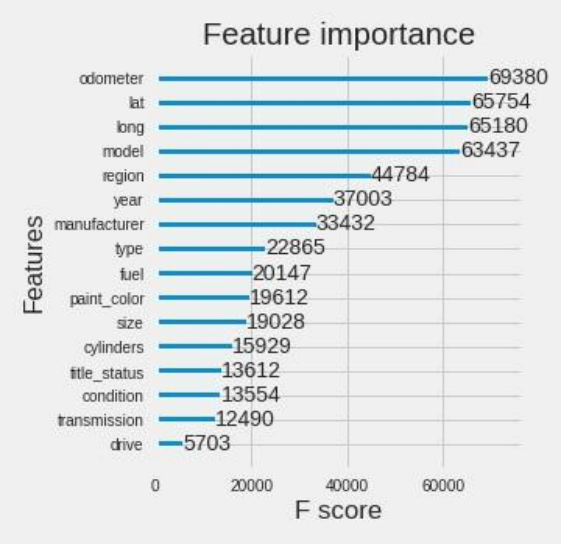


This is the simple bar plot which illustrates that ***year*** is the most important feature of a car and then ***odometer*** variable and then model, etc.

In our model, DecisionTreeRegressor is used as an estimator with 24 max_depth and creates 200 trees & learning the model with 0.6 learning_rate and result is R^2 of 86.4084%.

8) XGBoost: XGBoost stands for eXtreme Gradient Boosting

XGBoost is an ensemble learning method.XGBoost is an implementation of gradient boosted decision trees designed for speed and performance. The beauty of this powerful algorithm lies in its scalability, which drives fast learning through parallel and distributed computing and offers efficient memory usage.



This is the simple bar plot in descending of importance which illustrates that which **feature/variable** is an important feature of a car is more important. According to XGBoost, **Odometer** is an important feature whereas from the previous models **year** is an important feature. In this model,200 decision trees are created of 24 max depth and the model is learning the parameter with a 0.4 learning rate and accuracy of R^2 of 89.6623%.

Comparison of the performance of the models:

Result of Models:

Model	MSLE	RMSLE	Accuracy
Linear regression	0.00243399	0.04933557	59.3051%
Ridge regression:	0.00243399	0.04933553	59.3051%
Lasso regression	0.00243400	0.04933566	59.305%
KNN	0.00144004	0.03794796	76.4681%
Random Forest	0.00077811	0.00077811	87.5979%
Bagging Regressor	0.00143192	0.03784080	76.809%
Adaboost Regressor	0.00084475	0.02906475	86.4084%
XGBoost Regressor	0.00065047	0.02550431	89.6623%

Conclusion:

By performing different ML models, we aim to get a better result or less error with max accuracy. Our purpose was to predict the price of the used cars having 25 predictors and 509577 data entries.

Initially, data cleaning is performed to remove the null values and outliers from the dataset then ML models are implemented to predict the price of cars.

Next, with the help of data visualization features were explored deeply. The relation between the features is examined.

From the below table, it can be concluded that XGBoost is the best model for the prediction for used car prices. XGBoost as a regression model gave the best MSLE and RMSLE values.

FUTURE WORKS

Keeping the current model as a baseline, we intend to use some advanced techniques algorithms to predict car price as our future work. We intend to develop a fully automatic. Interactive system that contains a repository of used-cars with their prices. This enables a user to know the price of a similar car using a recommendation engine, which we would work in the future.

ACKNOWLEDGEMENT

We sincerely acknowledge the help and guidance of Dr. Jyoti Singh Kirar, Assistant Professor, DST-CIMS

BHU, without whose guidance the Project Report entitled “Car Price Prediction” would not have been possible.

REFERENCES

[1] Regression algorithms “analytics Vidhya” https://www.analyticsvidhya.com/blog/2021/05/5-regression-algorithms-you-should-know-introductory-guide/

[2] Popular Regression algorithms https://www.jigsawacademy.com/popular-regression-algorithms-ml/

[3] Linear Regression “Geeks for Geeks” https://www.geeksforgeeks.org/ml-linear-regression/

[4] Commonly used regression algorithms https://towardsdatascience.com/7-of-the-most-commonly-used-regression-algorithms-and-how-to-choose-the-right-one-fc3c8890f9e3

[5] Oreilly hands on Regression https://www.oreilly.com/library/view/hands-on-machine-learning/9781491962282/ch04.html