An Automated Car Price Prediction System Using Effective Machine Learning Techniques

Santosh Kumar Satapathy
Assistant Professor, Department of ICT
Pandit Deendayal Energy University,
Gandhinagar, INDIA
Satapathy.3336@gmail.com

Rutvikraj Vala
Student, B.Tech., ICT,
Pandit Deendayal Energy University,
Gandhinagar, INDIA
rutvikraj.vict18@sot.pdpu.ac.in

Shiv Virpariya
Student, B.Tech., ICT,
Pandit Deendayal Energy University,
Gandhinagar, INDIA
shiv.vict18@sot.pdpu.ac.in

2022 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES) | 978-1-6654-8004-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/CISES54857.2022.9844350

***Abstract*— This research focuses on Building a mathematical model that could predict the price of a second-hand car based on its current features. Determining the price of a used automobile is a difficult task because several factors like Current Mileage, Current Condition, Make, Year, etc., can influence the prediction prices of an automobile. And, from the perspective of a person who sells, it becomes a dilemma to predict the price of a second-hand car accurately. Thus, the point of interest of this challenge is in growing gadgets, studying models that can correctly expect the price of a used car primarily based on its capabilities. Due to this, in turn, a consumer can make a much more informed purchase. Therefore, We will be implementing and examining various Machine Learning Techniques with Data Analysis to Provide an Accurate and Easy to use solution.**

***Index Terms*—Machine Learning, Random Forest, XGBoost, Price Prediction, Supervised Learning, Gradient Boosting Regressor.**

I. Introduction

The manufacturer fixes the cost of new cars within the industry at an extra cost incurred by the government within a range of taxes which is also increasing daily. Therefore, people who purchase a brand-new car are assured of the value they are getting to qualify. The increasing prices of new vehicles day by day and the financial incapability of the people to buy them due to the pandemic caused second-hand car sales value to increase by a significant amount in a brief period. The second-hand car market in India accounts for about millions of vehicles per year. It is estimated to reach 50 billion USD by 2026, registering a Compound Annual Growth Rate of 15% between 2021-2026. Thus, we thought there is a desire for a second-hand car price prediction system to successfully determine a car's worth with its current features like Kilometers Travelled, Current Condition of the car, No. of Owners. Seats, Current Mileage, Power, etc... ". The growth of online markets such as CarDheko, Carwale, Quikr & Cars24, and others have necessitated a greater understanding of the patterns and instances that determine the value of a pre-owned vehicle on the market for both the client and the merchant" [7]. While some of above mentioned or other websites provide this service, their prediction method might not be the simplest. Besides, different techniques may contribute to predicting the worth of a second-hand car's actual value. Therefore, it's necessary to understand their market price for buying and selling. And thereby, this project aims to supply an answer that's accurate enough and simply available, keeping in mind all the present features of the car, as mentioned above. To achieve the goals mentioned above, we will use some Ensemble Machine Learning Techniques to predict the price accurately and integrate it with the Fronted of the website.

II. Literature Review

Going through the research part, we found that a few of the researchers implemented various Machine Learning Models, which gives us a better vision of our research and how we should select the best machine learning models for better accuracy and eventually do a better research project. The article [1] gives a good representation of how we can pre-process our data and remove some null values that help train machine learning more robustly and make it less keen to errors and uses models like KNN Classification & Regression Trees. And it is well observed that the buying year was slightly more dominant than the results observed in research. KNN, Classification & Regression Trees are compared on two different makes of cars. The outcomes show a direct proportionality between price; kilometers are driven & buying year, and an inverse proportionality between buying year and price.

This [2] research work presents a system that has been applied to predict the fair price for any second-hand car. The authors used ensemble machine learning techniques, namely Random Forest Algorithm and XGBoost.

The techniques are compared with each other to quantify an optimal one. The final performance results show that XGBoost performs better than that random forest algorithm. RMSE value of random forest recorded 3.44 on the other side XGBoost displayed 0.53 of RMSE value. So, taking into account the facts above, they can conclude that Boosting outperforms Random Forest Regressor to reduce the error rate and accuracy based on data from their region.

In [3], the authors focused more on the older regression methods like Multiple Regression, Lasso Regressor & overfitting of models. The error rate of all the models was well observed, under 5%. One more observation, the MSE of the Regression model was higher than the MSE of the Multiple & Lasso Regression models. Also, as the authors say," To get even more accurate models, we can also choose more advanced machine learning algorithms such as random forests. This ensemble learning algorithm creates multiple decision/regression trees.", which provides us with a better view of what to and how to implement.

Here in [4], the paper did a good amount of research on machine learning models like how to categorize your problem, whether it's a classification or regression problem, and how we can select a model best suitable for our purpose. The prediction part used two machine learning algorithms, i.e., Random Forest and Extra Tress Regressor. The prediction of this model is further compared with the test data set created by picking random values from the original data set. It shows that it's pretty accurate to the original data set. These two algorithms are exact and fast in prediction, irrespective of the size of the data set.

Going through the research [5], they trained on the KNN algorithm and got an accuracy of around 85%. In contrast, the accuracy of the Linear Regression algorithm is about 70% based on the data set collected from Kaggle. Also, these models are validated with 5 and 10 folds by applying K Fold Method.

Gazing at other articles like [6], the random forest has a steady but non-ideal effect on the costing model for a specific car. Still, it has considerable advantages over the more general linear regression when compared to linear regression. This shows that random forest is the best approach for dealing with complex models with many variables and samples. In paper [7], we discovered that Random Forest is the best performing model, with an accuracy of 86%. Also, the [8] report proposed an ensemble machine learning method by collecting different types of machine learning models like SVM, Random Forest Regressor & ANN. They build the model based on the data collected from a web crawler for price prediction of second-hand cars in Herzegovina and Bosnia. The accuracy of their final model is observed at around 87%.

This article [9] provides a price prediction system for cars. They used the supervised machine learning technique for the prediction, which uses many linear regression models, having observed a precision of 98 %. There are many independent variables in many linear regression models, but only one dependent variable whose values are compared to determine the accuracy of findings.

Finally, after reviewing the prior studies, we better understand how we will approach finding a better answer to our project's machine learning section. Aside from that, we noticed that the number of factors used in some of the studies is quite limited because we can't determine the price of a used car solely on 3-4 parameters such as previous owners, current price, distance traveled, and so on. As a result, we will attempt to use various elements to forecast the car's pricing.

III. METHODOLOGY

The methodology we will use is to gather data from the internet with a sufficient number of variables, analyze and process it for Machine Learning Models, and then apply the models to the processed data, picking the best and most accurate model available. Figure 1 shows the description of the methodology we will use for this study.

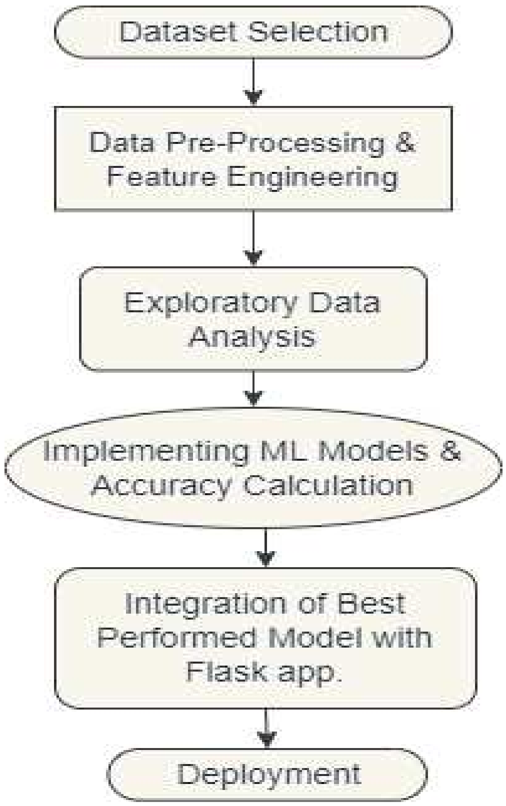

Fig. 1. Proposed Layout Diagram

	Name	Location	Year	Kilometers_Dri	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	New_Price	Price
0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First	26.6 km/kg	998 CC	58.16 bhp	5		1.75
1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67 kmpl	1582 CC	126.2 bhp	5		12.5
2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	18.2 kmpl	1199 CC	88.7 bhp	5	8.61 Lakh	4.5
3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77 kmpl	1248 CC	88.76 bhp	7		6
4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	15.2 kmpl	1968 CC	140.8 bhp	5		17.74
5	Hyundai EON LPG Era Plus Option	Hyderabad	2012	75000	LPG	Manual	First	21.1 km/kg	814 CC	55.2 bhp	5		2.35
6	Nissan Micra Diesel XV	Jaipur	2013	86999	Diesel	Manual	First	23.08 kmpl	1461 CC	63.1 bhp	5		3.5
7	Toyota Innova Crysta 2.8 GX AT 8S	Mumbai	2016	36000	Diesel	Automatic	First	11.36 kmpl	2755 CC	171.5 bhp	8	21 Lakh	17.5
8	Volkswagen Vento Diesel Comfortline	Pune	2013	64430	Diesel	Manual	First	20.54 kmpl	1598 CC	103.6 bhp	5		5.2
9	Tata Indica Vista Quadrajet LS	Chennai	2012	65932	Diesel	Manual	Second	22.3 kmpl	1248 CC	74 bhp	5		1.95
10	Maruti Ciaz Zeta	Kochi	2018	25692	Petrol	Manual	First	21.56 kmpl	1462 CC	103.25 bhp	5	10.65 Lakh	9.95

Fig. 2. Sample Dataset

A. Dataset

The used car data set that we will be using is from Kaggle. As mentioned above, our literature review's conclusion contains 12 factors which you can see in the screenshot of the sample dataset in Fig. 2. The description of the factors mentioned above are :

- **Name**: Gives the car's full name with model build and company's name.
- **Location**: Gives the place at which the car was driven.
- **Year**: Gives the year in which the vehicle is bought.
- **Kilometers Driven**: Provides the total no. of KM traveled by car till time.
- **Fuel Type**: Gives an idea about the car's fuel type from CNG, Diesel, or Petrol.
- **Transmission**: Gives if the vehicle has an Automatic or Manual transmission type.
- **Owner Type**: Provides previous no. of owners of the car.
- **Mileage, Engine & Power**: Few of the car's internal build factors will effect on car's price.
- **New Price**: Gives the current showroom price of the vehicle if available.
- **Price**: Is the deciding factor on which we will get the amount of accuracy of our implemented Models.

B. Data PreProcessing

Data pre-processing and analyzing is the initial stage before training on any model. As a result, we processed every column individually, checking for any null values in the data and estimating how much use each column is.

Going to the first column is the Index for each data point, and hence we can simply drop it. Moving to **the Name** column, we see several cars in the dataset, some with a count higher than 1. Sometimes, a vehicle's resale value also depends on the car's manufacturer. Hence, We extracted the manufacturer from this column and added it to the dataset. We will simply drop it to the Location column as the location won't matter in deciding the price. From the **Year** column, we will extract how a car is by subtracting the Year column value from the current year. These are categorical columns, moving to **Fuel Type, Transmission & Owner Type** columns. Thus, they will be converted to dummy variables before being used. The **Kilometers Driven** simply contains numerical values, which will be help full. **The mileage** column defines the car's mileage so that we will be extracting the numerical value out of each string and saving it. The **Engine** values are defined in 'CC,' so we removed 'CC' from the data. Similarly, **Power** has bhp, so we will be releasing 'bhp' from it. Also, as there are missing values in **Engine, Power & Seats**, we will be replacing them with the mean. Coming to **the New Price** column, as most of its values are missing, we will also be dropping it.

As the last step after processing columns, we create dummy columns for categorical columns and scale them to the data.

C. Exploratory Data Analysis

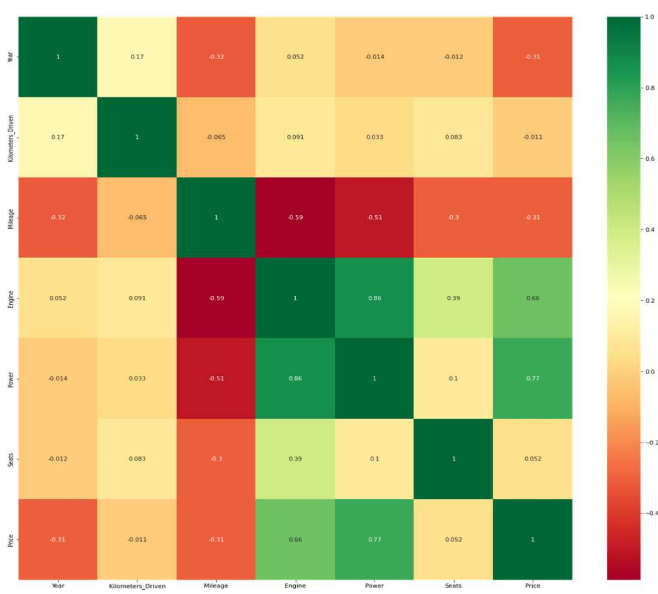

Fig. 3. Heat Map Plotting

Now, after the Data is processed, we will be moving to the analyzing part, from which we get a better understanding of how a factor affects the prediction price of the car and

How much of it co-relates to the other factors in the data. As you can see in Figure 3, we initiated our analysis by displaying the Heat-map Representation of our data collection. As shown in Fig. 3, the Heat Map provides a broad picture of how the variables in the given dataset are correlated. The dark green color implies that variables on both the x-axis and the y-axis show Direct Proportionality, while the dark red color indicates Inverse Proportionality.

As shown in Fig. 3, the variable Year has a negative proportionality to Mileage & Price, implying that the more the driven car, the Mileage & Price will be lower and vice versa. Also, the Engine exhibits a greenish color towards Price & Power, suggesting that good engine performance will lead to a higher price and more Power consumption for the car. The Graph in Fig. 4 is plotted between Fuel Types.

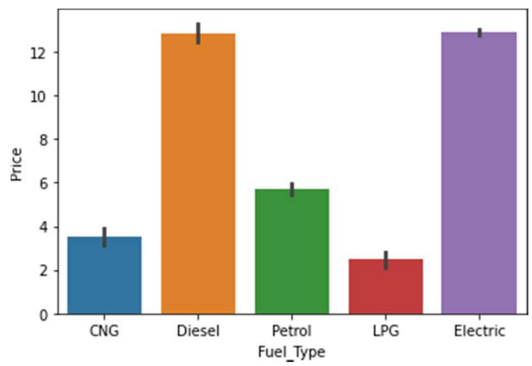

Fig. 4. Fuel Type VS Price Plotting

And price. And as shown in the graph, the cost of the Diesel & Electric car is the highest followed by Petrol, CNG & LPG. This is the same as when you buy a new car, the price of the Diesel model is the highest than the cost of the Petrol model, and also the price of an electric vehicle is usually higher than non-electric ones. Plotting the graph of Year Vs. Price, we get

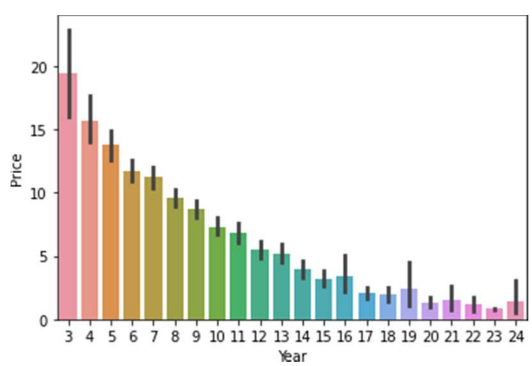

Fig. 5. Year Vs. Price Plotting

An insight into how much older cars exists in the database. And From the Fig. 5, we can see the most number of vehicles are used only for 3-5 years, and the price of these cars is also the highest, which is clear. Also, it shows that people mostly spend less money on more used vehicles. Instead, they prefer a less used car with more money. In the

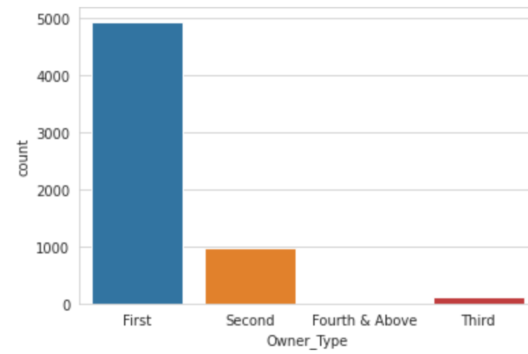

Fig. 6. Owner Type Count Plotting

Owner Type Count graph Fig. 6 shows that most of the cars are only used by First Owner and very few by Two or more owners, which accounts for most people buying a Second-Hand car.

Also, from the upcoming scatter plot, we can see most of the used cars have a Mileage range of around 10-25, which is quite close to What would be in according Indian roads, and some Zero values which were not present in the data we replaced with dummies. Above all, the analysis part.

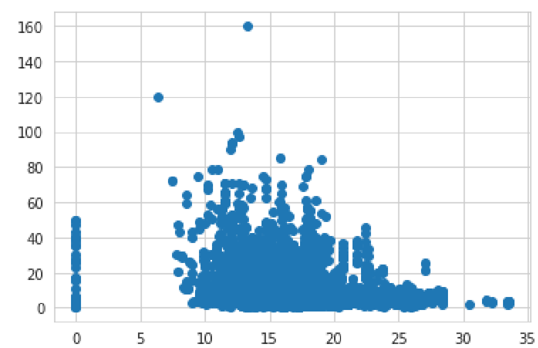

Fig. 7. Mileage Vs. Price Plotting

We know that the data we are using is quite like that of the Indian car market, and all the above plots show resemblance to the Indian market. Thus justifying its cause for the same.

D. Selection & Implementing Machine Learning Models

As we have to predict the car price of user input, i.e., it is a regression problem, we used many different algorithms like linear regression, lasso and ridge regression, random forest, and we have also used Gradient Boosting XGBoost.

Python Libraries used in this project are:

1) NumPy - Full form of NumPy is Numerical Python. NumPy is mainly used for working with complex data structures like arrays. We can use this library freely as

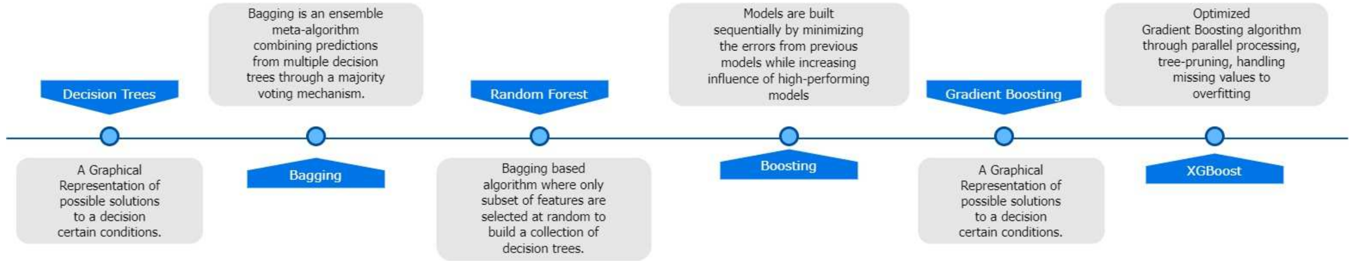

Fig. 8. Different Types of Ensemble Techniques

It is open source. Arrays in NumPy work much faster compared to lists of python.

2) Pandas is a python library used to analyze the data. We also use pandas for importing datasets, writing, checking data types, handling missing values, etc.

3) Scikit Learn - This library is used for getting inbuilt code written for our machine learning algorithms like regression, classification, etc. We directly import them into our code while working with it. NumPy, SciPy, and Matplotlib were used to create this library.

4) Matplotlib - This library is used for visualizing the data. The plot, which is a part of the library, provides users with various fields that we use to understand our data better. Hard-looking data can be understood easily by using this library.

Different Machine Learning Algorithms used are:

1)Linear Regression: The linear Regression algorithm is based on Supervised learning. There are two types of features in it, one is the independent feature, and the other is the dependent feature. We want to find the dependent feature's value based on the independent feature's values. The algorithm focuses on finding the relationship between both parts. Price is the dependent feature in our work, and others are the autonomous feature. We try to find the statistical relationship between both parts. We then give different independent features to see what outcome our model gives. The main aim of this algorithm is to find the best fit line which has the slightest error in it. Error is the distance between actual and predicted values. We use mean-squared error to calculate the error of the model.

2) Ridge and Lasso Regression: We often see over-fitting using linear regression, and to overcome that, we use Ridge or Lasso regression. It also aims to reduce multicollinearity, i.e., correlation among independent features. Lasso regression helps us in feature selection, and Ridge regression helps in reducing multicollinearity. So, the ridge is a kind of advanced algorithm compared to Linear regression.

Another algorithm we have used in this work is the Ensemble technique. They combine many base models to create one best prediction model. This technique is separated into two components Bagging and Boosting Techniques.

Understanding the decision tree is most important before going to models like random forest and bagging techniques. This type of algorithm is also known.

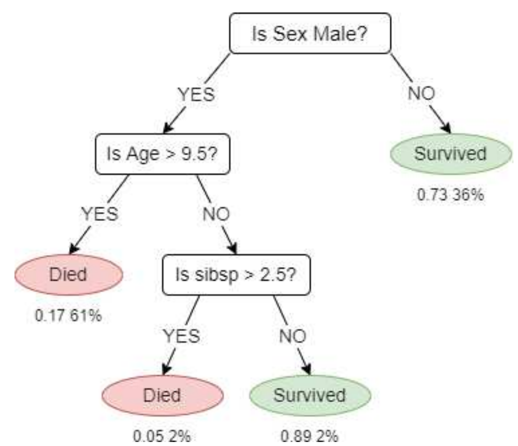

Fig. 9. Demo of Decision Tree

As for the Classification and Regression Tree. Above is an example of a decision tree built from a titanic dataset. The decision tree has a root node which we can find using Information Gain and Entropy. The answer of the root node will decide the internal node/branch in the tree. A leaf node will not be further divided in the tree anymore. In our case, survived or died is the leaf node. So the main aim of the decision tree is to know what to choose as the root node and what features to take for splitting the tree further, and to know when to stop breaking the tree further.

3) Random Forest: Random Forest is another" Tree"-based algorithm that makes decisions based on the quality aspects of multiple Decision Trees. As forest comprises many trees and uses many decision trees in this algorithm, We got its name, Random Forest.

Also, the word random is justified because we have randomly constructed decision trees in our algorithm. Overfitting is a subject of the dispute regarding the Decision Tree method. This problem can be addressed by using the Random Forest Regression instead of the Decision Tree Regression. Furthermore, the Random Forest approach is master and more reliable than existing regression models.

Let's understand the working of the Random Forest Algorithm:

- Step-1: n number of records is taken from the dataset with m number of records.
- Step-2: singular decision trees are made for each sample
- Step-3: Output is generated by the decision tree.
- Step-4: final output is done based on the majority outcome of individual decision trees.

The following algorithm which we will use is gradient boosting. It runs several models parallelly. Every model reduces the residue/error, and so, at last, we get the model with the slightest mistake. So we can get a better accuracy of the model based on different machine learning techniques.

4) XGBoost: XGBoost stands for eXtreme Gradient Boosting. It is a distributed gradient boosting framework that has been optimized for efficiency, flexibility, and portability using gradient enabling as an implementation. XGBoost is a parallel tree-growing algorithm that solves a wide range of data science issues quickly, effectively, and accurately. XGBoost has higher execution speed and model performance compared to gradient boosting. Fig. 8 shows different Ensemble techniques, like how different algorithms are made from the average decision tree-like bagging and growing techniques. If we have more trees in boosting, it might take the model to overfit, so we must see how and what range we have to use of our hyper-parameters.

E. Performance Analysis

This section covers the performance of different algorithms on our dataset. We started with pre-processing and data cleaning on the dataset to make it worthwhile for our model. We also removed all null values by filling them up with appropriate values using a measure of central tendency.

The dataset is split into 85% training and 15% for testing. From the exploratory data analysis part, we created many different diagrams to understand the relation of independent features with our pendant features. From the heat map, we found out how our numerical columns like Mileage, Seat, etc., features are correlated with our Price columns and with what amount are they correlated. Then we did some barplot, counterplots, and scatterplots to understand the input data better. The first model which we used is Linear Regression.

Model	R2	RMSE
Linear	62.00	6.9673
Ridge	61.84	2.6422
RFR	90.00	3.5625
GBR	90.29	3.5202
XGBoost	92.70	2.6395

Fig. 10. Applied Model Accuracy & Error Rates

As the problem is of Regression type, our first thought is Linear Regression. Using Scikit Learn, we created a model of Linear Regression and passed our training dataset to it. For accuracy, we used R-squared values. The linear model gave us an accuracy of 61.99%, which was very low.

So the next model we used was Ridge Regression because it can reduce the model complexity issue we face in linear regression. Using Scikit Learn, we created a Ridge model and utilized the parameter alpha. As the value of alpha increases, the model complexity reduces. We have also used RandomizedSearchCV, which is a hyper-parameter tuning method. The outcome of this model was 62% which was the same as linear regression. The third model we choose is Random Forest which is an ensemble-bagging technique. It makes multiple decision trees, and the majority of the output of the trees is selected as the final decision. We also used many hyperparameters in Random Forest, which are listed below:

- **n_estimators**: The number of trees you want to build before computing maximal vote or forecasting mean values
- **max_depth**: Most number of levels in a tree.
- **min_sample split**: The smallest number of leaves needed to separate an internal node.
- **min_sample leaf**: The smallest number of samples must be present at the leaf node.
- **max_features**: Selection of attributes to be considered at each split.

We also used the RandomizedSearchCV method for tuning. This model gave an accuracy of 90.17%, which was the highest among all three. We also have used cross-validation in hyperparameters which will further divide training data into k different parts, of which k-1 will be taken for training, and the last part will be used for testing. It helps increase the stability of output values meaning output value fluctuation will be less when two nearer inputs are given. Our next idea was to use Boosting techniques. So we used the Gradient Boosting Regressor to see what accuracy it provides. Again used Scikit to learn to import Gradient.

Boosting Regressor with the same hyperparameters of Random Forest and one extra parameter Learning Rate added. Again we used RandomizedSearchCV and fitted the model. This model gave used the best accuracy of 91.35%. This accuracy was good enough, but we wanted to get more, so we used the updated Gradient boosting technique called XGBoost.

XGBoost is a more generalized form of Gradient Boosting Regressor. It also uses L1 and L2 regularization, which helps in improving model generalization capabilities. XGBoost gives higher performance than Gradient Boosting Regressor, and the training is also high-speed and can be paralleled between clusters. We used hyper-parameters like max depth, reg alpha, min child weight, n estimators, and learning rate. Using boost, we imported XGBRegressor, used the tuning method, and fitted and trained the training data. The outcome was the best among them all, and we got an accuracy of 92.66% on the test data.

F. Integrating Machine Learning Models

Once we've generated our optimal model, we'll need to integrate it with the responsive web-based application. We'll transform the model to a '.pkl' file, which is created using Python's pickle module, allowing objects to be serialized to files on record and afterward deserialized back into the implementation at program execution. As a result, our model could be reused for flask integration.Flask is a python based micro web framework used to build and integrate small applications with less complexity. We need a template file containing the front end of the web application for the flask app, which is made with HTML, CSS & JavaScript.We tested the system on our local machine once it was merged with the Flask app and looked for any faults or defects. After satisfying the app's functionality, we pushed the code to GitHub and then deployed it onto the Heroku platform by connection to the GitHub repository. Heroku is a cloud service that allows developers to publish, run, and administer applications.

IV. CONCLUSIONS

To conclude, our work provided an accurate solution to the problem statement of Second-Hand Car Price Prediction. After running different machine learning algorithms and ensemble techniques, we observed that XGBoost was the best and most consistent model, with an accuracy of roughly 92% and an error rate of 2.64. The model forecasts that the original pricing will be well within the appropriate range. The other ensemble algorithms also performed well, with the Random Forest Regressor achieving 90% with a 3.56 error rate and the Gradient Boosting Regressor scoring about 91% with a 3.52 error rate. The previously described aims are also very well met by the freely accessible Web Application and extremely easy to use and excellent User Interface.

REFERENCES

[1] A. Chandak, P. Ganorkar, S. Sharma, A. Bagmar, and S. Tiwari, "Car Price Prediction Using Machine Learning," International Journal of Computer Sciences and Engineering, vol. 7, no. 5, pp. 444–450, 2019.

[2] C. Longani, S. Prasad Potharaju, and S. Deore, "Price Prediction for Pre-Owned Cars Using Ensemble Machine Learning Techniques," Recent Trends in Intensive Computing, 2021.

[3] P. Venkatasubbu and M. Ganesh, "Used Cars Price Prediction using Supervised Learning Techniques," International Journal of Engineering and Advanced Technology, vol. 9, no. 1S3, pp. 216–223, 2019.

[4] A. Pandey, V. Rastogi, and S. Singh, "Car's Selling Price Prediction using Random Forest Machine Learning Algorithm," SSRN Electronic Journal, 2020.

[5] K. Samruddhi and R. Ashok Kumar, "Used Car Price Prediction using K-Nearest Neighbor Based Model," International Journal of Innovative Research in Applied Sciences and Engineering, vol. 4, no. 3, pp. 686–689, 2020.

[6] C. Chen, L. Hao, and C. Xu, "Comparative analysis of used car price evaluation models," AIP Conference Proceedings, May 2017.

[7] B.Lavanya, Sk.Reshma, N.Nikitha, and M.Namitha, "Vehicle resale price prediction using machine learning," UGC Care Group I, 2021.

[8] E. Gegic, B. Isakovic, D. Keco, Z. Masetic, and J. Kevric, "Car price prediction using machine learaning techniques," TEM Journal, February 2019.

[9] K. Noor and S. Jan, "Vehicle Price Prediction System using Machine Learning Techniques," International Journal of Computer Applications,vol. 167, no. 9, pp. 27–31, 2017.

[10] N. Pal, D. Sundararaman, P. Arora, P. Kohli, and S. S. Palakurthy,"How much is my car worth? a methodology for predicting used cars prices using random forest," Future of Information and Communications Conference (FICC), 2018.

[11] F. R. Amik, A. Lanard, A. Ismat, and S. Momen, "Application of Machine Learning Techniques to Predict the Price of Pre-Owned Cars in Bangladesh," Information, vol. 12, no. 12, p. 514, 2021.

[12] E. Gokce, "Predicting Used Car Prices with Machine Learning Techniques," 2021. [Online]. Available: https://towardsdatascience.com/predicting-used-car-prices-with-machine-learning-techniques-8a9d8313952

[13] A. Chandak, P. Ganorkar, S. Sharma, A. Bagmar, and S. Tiwari, "Car Price Prediction Using Machine Learning," International Journal of Computer Sciences and Engineering, vol. 7, no. 5, pp. 444–450, 2019.

[14] C. V. Narayana, C. L. Likhitha, S. Bademiya, and K. Kusumanjali,"Machine Learning Techniques To Predict The Price Of Used Cars:Predictive Analytics in Retail Business," 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), 2021.

[15] S. S, "Build and deploy a car prediction system," 06 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/05/build-and-deploy-a-car-price-prediction-system/

[16] G. S. k. S.E. Viswapriya, D.S.S Sharma, "Vehicle Price Prediction using SVM Techniques," International Journal of Innovative Technology and Exploring Engineering, vol. 9, no. 8, pp. 398–401, 2020