Analysis of Used car dataset using Machine Learning Models in Spark CLI

Contributed by Jo In Kang, Heta Parekh, Priya Ramdas
Email id: kjikji956@gmail.com, hparekh2@calstatela.edu, pramdas2@calstatela.edu

Abstract: With the growing mobility among cities, not everyone can afford a brand-new car. Today, most of the population has started to purchase used cars, and many platforms offer excellent services to customers. Cargurus is one of the platforms to provide a preference-based automobile as on the needs and demands of consumers. This project uses the Cargurus dataset to perform prediction analysis and measure the accuracy analysis using machine learning models in the Spark environment.

1. Introduction

Cargurus is a well-known automotive shopping website headquartered in Cambridge, Massachusetts, where they are assisting millions of users in comparing the local listings of the user as well as new cars and seller details information. Also, dealers advertise on this site, and CarGurus connect with dealers and customers by displaying the inventory on the site. Each buyer has different expectations or features when purchasing a used car. The key factors present in the data are accident history, body type, price range, feature list, seller rating, etc. In this project, we have used linear regression, random forest regression, Gradient boost tree, and recommendation models to understand the predictive patterns in terms of the price and seller rating, which would help the users to make informative decisions before customers purchase used cars from the CarGurus. To get accurate results-big data tools, methods, and various technologies are implemented in this term project. The dataset was collected as the used car dataset from Kaggle, and it consists of 3 million car records from 2005 to September 2020. The initial models were built on data bricks, and later, the code was used to run the models with the framework of Hadoop in the Spark CLI shell.

2. Related Work

A related study and analysis "*Statistical Analysis on second-hand vehicle sales using Big data Technologies and Linear Regression*"[1] was performed in 2019 where they used Hadoop for handling the data and Python Programming language for performing regression analysis and building a model that predicts the price of used cars. Their dataset consists of 1.7 million records of car sales data from the year 1900 to 2019. Their study mainly centres on trends of used car sales over the years, State-wise distribution of car sales, and which manufacturer is the most trusted in the second-hand vehicle sales market. In contrast, we have used price and seller rating as the significant outcome to analyze the data and get valuable insights.

Related analysis on "*Used car price analysis*" [2] was a part of the Capstone project performed in the Jupyter Notebook. The primary goal of their project was to build a model that decides if the asking price for a particular car is acceptable, given the information provided in the listing. And they have provided code that will be useful in finding recommendations for related vehicles with a lower price, lower miles, and one that is slightly more expensive.

Another analysis has been achieved by Luc Frachon [3] who served in automotive industry for 12 years. The dataset they used for analysis was found on Kaggle, the well-known Machine Learning Competition website. His analysis was aimed on Used Cars Dataset to foresee the price for the used cars by comparing features like date of manufacture, milage on meter. Analysis has the brand category variable that went out to be useful as well as it presents a significant association to price. The brand and model variables show us a lot about the German market, one of the most high-end markets in Europe. Using the exploratory analysis, assessment constructed a linear model that explains 76.5% of the variance in the data. This analysis evaluated the car makers, how much time it held to go in combined days range to 0-91 ,91-182,182-365,365-above.

3. Specifications

File Size	9.29 GB
No. of files	1
File format	CSV
Years analyzed	2006-2020
Country	USA
No. of records	3.0 Million

Table 1 Data Specification

Cluster Version	Hadoop 3.2.1-amazon-3.1
Number of Nodes	5
Memory size	30874 KB
CPU	8 OCPUs
CPU Speed	2.20 GHz
HDFS capacity	147 GB
Storage	481 GB

Table 2 H/W Specification

4. Background

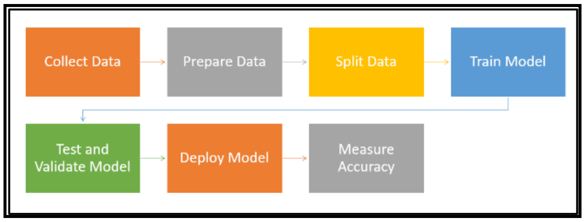


Fig 1.1 Workflow Architecture

The above process listed in Fig 1.1 are the steps to build our machine learning models. The first step is to collect the data and find the right source of the problem to define measurable and quantifiable objectives to achieve at the end of this project. The data preparation stage consisted of transforming the data into a structured format and converting it to desired datatypes, cleaning the missing values, removing duplicate data, and filtering unwanted rows and columns.

Later, split the cleaned data into training and testing sets. The training set is the set that the models learn from, whereas the testing set is used to check the model's accuracy after training. Training the models is one of the essential steps in any machine learning algorithm as it passes the prepared data to find patterns and make predictions. Finally, testing and validating the model's type depends on the requirements and expected outcomes. And hence, the accuracy of the data can be measured through Coefficient of Determination (R2) and Root Mean Square Error (RMSE).

5. Machine Learning Models

i. Linear Regression Model:

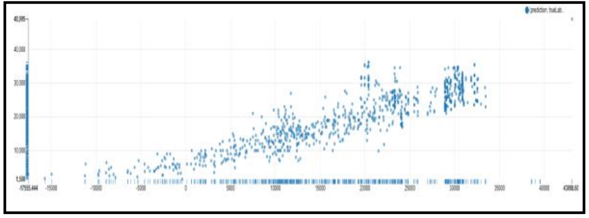


Fig 2.1 Linear Regression Plot

Linear Regression is a model that combines a numerical set of input values to find the predicted outcome of the input values. In the figure 2.1, we have done the prediction of price while taking features column as the input. The feature column consists of all the car features i.e. mileage, engine displacement, number of accidents, seller rating and horsepower generated by the car. The graph above shows the labels scattered majorly in the center giving an approximate diagonal line with the performance metric of R^2 = 0.735 and RMSE of 7766.5. The R^2 value here represents smaller differences between the observed data and the predicted values.

ii. Random Forest Model:

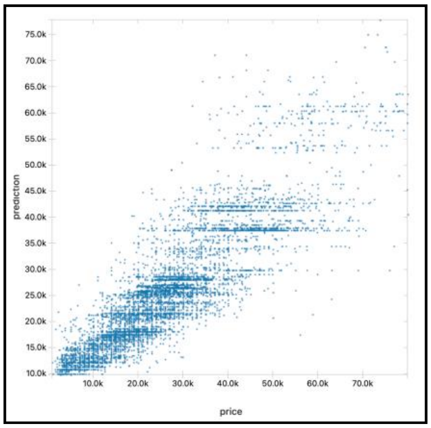
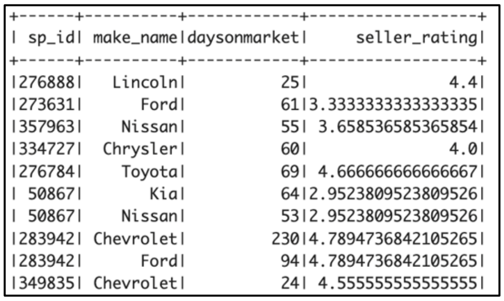


Fig 2.2 Random Regression Model

Random Forest Regression is a supervised learning algorithm that uses the ensemble learning method for regression, The ensemble learning method is a technique that combines the prediction from multiple machine learning algorithms to make a more accurate prediction then a single model. For our regression, we set the number of trees to 10 and number of levels (depth) for the decision tree to 5. As a result, we got RMSE of 6319.9 and R^2 of 0.825; compared to the linear regression model, this model's accuracy has been improved.

iii. Recommendation Model:



```
+------+----------+------------+------------------+
| sp_id| make_name|daysonmarket|     seller_rating|
+------+----------+------------+------------------+
|276888|   Lincoln|          25|               4.4|
|273631|      Ford|          61|3.3333333333333335|
|357963|    Nissan|          55| 3.658536585365854|
|334727|  Chrysler|          60|               4.0|
|276784|    Toyota|          69| 4.666666666666667|
| 50867|       Kia|          64|2.9523809523809526|
| 50867|    Nissan|          53|2.9523809523809526|
|283942| Chevrolet|         230|4.7894736842105265|
|283942|      Ford|          94|4.7894736842105265|
|349835| Chevrolet|          24| 4.555555555555555|
```

Fig 2.3: Recommendation Model Outcome

A recommender model, or a recommendation system (sometimes replacing 'system' with a synonym such as platform or engine), is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item. For our recommendation model, we have used parameter combination and train validation to make data more general and as a result, we got RMSE: 0.44 and R2: 0.69.

iv. Gradient Boost Tree (GBT) Model:

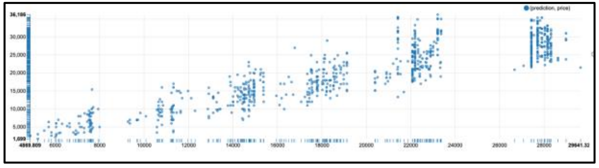


Fig 2.4 Gradient Boost Tree Model

Gradient Boosted DecisionTree(GBT)[3] is a forward learning ensemble method that obtains predictive results through incrementally improved estimation. This machine learning model is used specifically for regression and classification tasks. Compared to a normal Decision Tree, Boosting trees increase accuracy and reduce speed. The GBT method generalizes tree boosting to minimize this problem. We put the hyperparameter in the GBTRegressor model of pyspark maxIter=10, maxDepth=10 for training. As a result, the run-time was: R^2 was 0.847 and the RMSE was 5912.7. This model produces improved accuracy, although the learning time is longer than the above models.

v. GBT with Train Validation and Cross Validation Split:

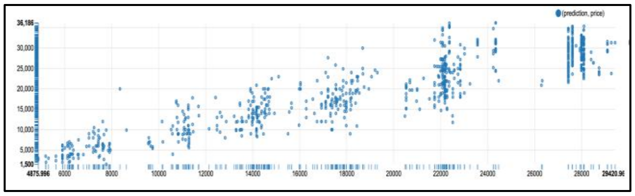


Fig 2.5 GBT with Cross Validation Split

Train Validation Split is Validation for hyper-parameter tuning. Randomly splits the input dataset into train and validation sets and uses evaluation metric on the validation set to select the best model. Like CrossValidator, but only splits the set once. R^2 for Cross Validation was 0.847 whereas for Train Validation split was 0.848.

vi. Factorized Machine Learning Model (FM)

It is a general supervised learning model that can be used to apply for both classification and regression tasks. It is an extensive linear regression model which is designed to find outcomes between features within a large dimensional dataset. The RMSE and R^2 for the respective factorized model is 8910.7 and 0.651 respectively.

6. Conclusion

We analyze the results of deep learning prediction models through big data. We compared R^2 (R-squared values), RMSE (Root Mean Square Error), and Runtime. R^2 is a commonly used performance metric for regression analysis. It is expressed as a value between 0 and 1, and it is judged that the closer it is to 1, the higher the accuracy. RMSE is a commonly used measure when dealing with the difference between the estimated value or the value predicted by the model and the value observed in the real environment. The below table summarizes the results of the experiment. GBT Model has better results in this model with a comparatively short runtime.

"Price" Prediction model	R^2	RMSE	Runtime(sec)
Factorization Machines learning	0.55	13117	4875.29
Linear Regression	0.736	7766.5	20.91
Random forest	0.825	6319.9	40.03
GBT /w CV	0.847	5913.6	355.01
GBT	0.847	5912.2	76.11
GBT /w TV	0.848	5892.6	144.27

"Seller Rating" Prediction model	R^2	RMSE	Run Time
Recommendation Model	0.69	0.44	49.56

Table 3 R^2, RMSE and Runtime of the experiment

References:

[1] https://www.slideshare.net/YashIyengar/big-data-analysis-of-second-hand-car-sales
[2] https://github.com/clrife/CarPriceAnalysis
https://github.com/Heta-Parekh/MachineLearningModels
https://www.kaggle.com/ananaymital/us-used-cars-dataset
[3]Mason, L., Baxter, J., Bartlett, P. and Frean, M. (2000). Boosting algorithms as gradient descent. In *Advances in Neural Information Processing Systems* **12** 512--518. MIT Press, Cambridge, MA.