FLIP ROBO

NAME OF THE PROJECT

USED CAR PRICE PREDICTION

Submitted by:

DEEPAK KUMAR

Table of Contents

INTRODUCTION

The price of a new car 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 prices of new cars and the financial incapability of the customers to buy them, Used Car sales are on a global increase. Therefore, there is an urgent need for a Used Car Price Prediction system which effectively determines the worthiness of the car using a variety of features. Existing System includes a process where a seller decides a price randomly and buyer has no idea about the car and its value in the present day scenario. In fact, seller also has no idea about the car's existing value or the price he should be selling the car at. To overcome this problem, we have developed a model which will be highly effective. Regression Algorithms are used because they provide us with continuous value as an output and not a categorized value. Because of which it will be possible to predict the actual price a car rather than the price range of a car. User Interface has also been developed which acquires input from any user and displays the Price of a car according to user's inputs.

Key Words: Linear Regression, Used car Prediction, Ridge Regression, Lasso Regression, Decision Tree Repressor

Business Problem Framing

Determining whether the listed price of a used car is a challenging task, due to the many factors that drive a used vehicle's price on the market. The focus of this project is developing machine learning models that can accurately predict the price of a used car based on its features, in order to make informed purchases. We implement and evaluate various learning methods on a dataset consisting of the sale prices of different makes and models across cities in the United States. Our results show that Random Forest model and K-Means clustering with linear regression yield the best results, but are compute heavy. Conventional linear regression also yielded satisfactory results, with the advantage of a significantly lower training time in comparison to the aforementioned methods.

Conceptual Background of the Domain Problem

There are two primary phases in the system: 1. Training phase: The system is trained by using the data in the data set and fits a model (line/curve) based on the algorithm chosen accordingly. 2. Testing phase: the system is provided with the inputs and is tested for its working. The accuracy is checked. And therefore, the data that is used to train the model or test it, has to be appropriate. The system is designed to detect and predict price of used car and hence appropriate algorithms must be used to do the two different tasks. Before the algorithms are selected for further use, different algorithms were compared for its accuracy. The well-suited one for the task was chosen

Review of Literature

The first paper is Predicting the price of Used Car Using Machine Learning techniques. In this paper, they investigate the application of supervised machine learning techniques to predict the price of used cars in Mauritius. The predictions are based on historical data collected from daily newspapers. Different techniques like multiple linear regression analysis, lasso regression, decision trees and random forest regression have been used to make the predictions. The Second paper is Car Price Prediction Using Machine Learning Techniques. Considerable number of distinct attributes are examined for the reliable and accurate prediction. To build a model for predicting the price of used cars in Bosnia and Herzegovina, they have applied three machine learning techniques (Artificial Neural Network, Support Vector Machine and Random Forest). The Third paper is Price Evaluation model in second hand car system based on BP neural networks. In this paper, the price evaluation model based on big data analysis is proposed, which takes advantage of widely circulated vehicle data and a large number of vehicle transaction data to analyse the price data for each type of vehicles by using the optimized BP neural network algorithm. It aims to establish a second-hand car price evaluation model to get the price that best matches the car.

Analytical Problem Framing

Data Wrangling

In this section, it will be discussed about how data cleaning and wrangling methods are applied on the craigslist used cars data file.

Before making data cleaning, some explorations and data visualizations were applied on data set. This gave some idea and guide about how to deal with missing values and extreme values. After data cleaning, data exploration was applied again in order to understand cleaned version of the data.

Data cleaning: First step for data cleaning was to remove unnecessary features. As a next step, it was investigated number of null points and percentage of null data points .As the second step, some missing values were filled with appropriate values. For the missing 'condition' values, it was paid attention to fill depending on category.

The Exploratory Data Analysis (EDA)

While exploring the data, we will look at the different combinations of features with the help of visuals. This will help us to understand our data better and give us some clue about pattern in data. Car_Price is the feature that we are predicting in this study. Before applying any models, taking a look at price data may give us some ideas.

Transmission: Transmission is another feature that has a dominant sub category in the used car market. Global economic recession might have an impact on used car market and affect market. Its market share is still so low compared to automatic transmission, but it is still considerable. The increase in other transmission type can be caused by a couple reason. First possibility is that increase in continuously variable transmission (CVT). CVT is more environmentally friendly and fuel efficient. There might be a promotion for this kind of technology. Another possibility is that some seller on Craigslist website did not fill transmission section of the car information. The website might directly put them in the 'other' category. This also explains the increase in the 'other' category of transmission.

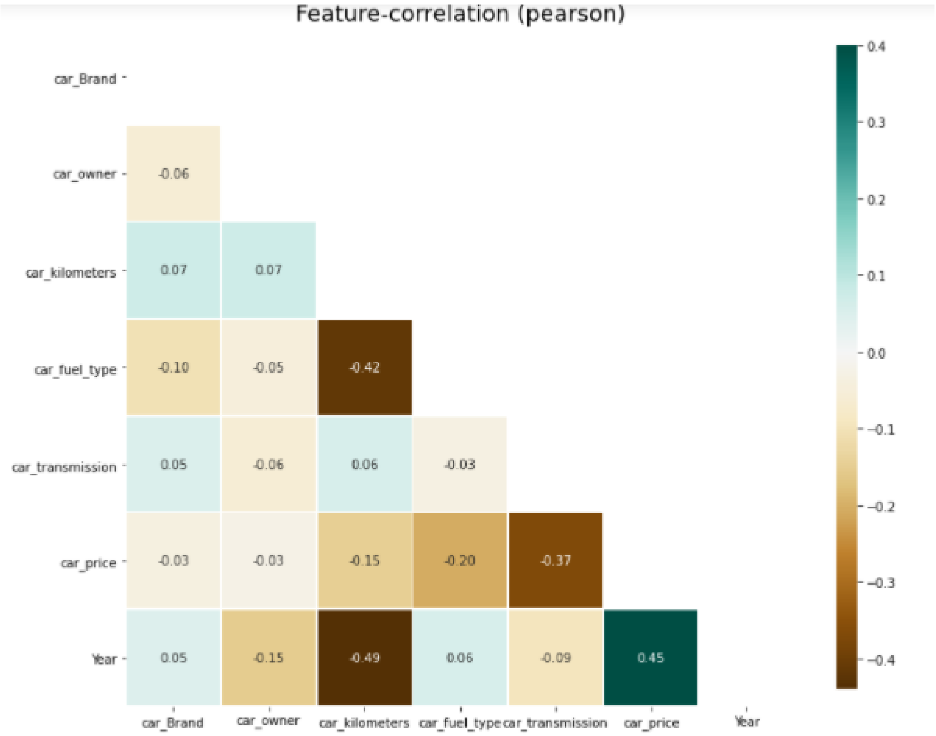
In order to get a better understanding of the data, we plotted a histogram of the data. We noticed that the dataset had many outliers, primarily due to large price sensitivity of used cars. Typically, models that are the latest year and have low mileage sell for a premium, however, there were many data points that did not conform to this. This is because accident history and condition can have a significant effect on the car's price. Since we did not have access to vehicle history and condition, we pruned our dataset to three standard deviations around the mean in order to remove outliers.

Data Inputs- Logic- Output Relationships

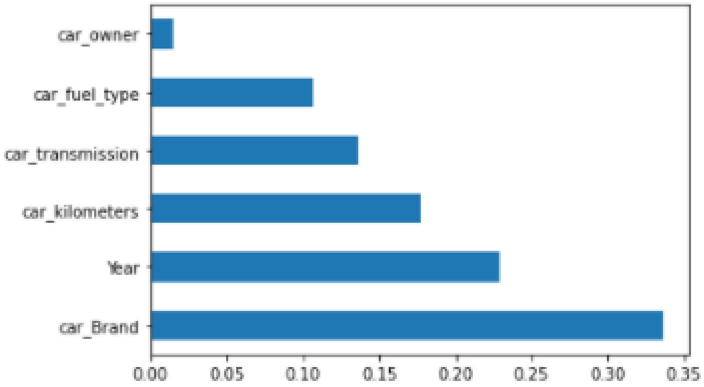
Correlations: It's often good to plot a correlation matrix to give you an idea of relationships that exist in your data. It can also guide your model building. For example, if you see a lot of your features are correlated with each other you might want to avoid linear regression. State the set of assumptions (if any) related to the problem under consideration.

The correlation measure used here is Pearson's correlation. In our case the lighter the square the stronger the correlation between two variables.

Features related to space such as lot frontage, garage area, ground living area were all positively correlated with sale price as one might expect. The logic being that larger properties should be more expensive. No correlations look suspicious here.



Categorical Relations: Sales price appears to be approximately normally distributed within each level of each category. No observations appear, untoward. Some categories contain little to no data, whilst other show little to no distinguishing ability between sales class. See full project on GitHub for data visualization.



Get Importance of variables From the correlations, we can get an overview of some important numeric variables .I want to get more details about the importance of variables which including the factor variables.

Hardware and Software Requirements and Tools Used

Tools: I used Python and Jupyter notebooks for the competition. Jupyter notebooks are popular among data scientist because they are easy to follow and show your working steps. Please be aware this code is not for production purposes, it doesn’t follow software engineering best practices.

Libraries: These are frameworks in python to handle commonly required tasks. I Implore any budding data scientists to familiarize themselves with these libraries:

Pandas — For handling structured data

Scikit Learn — For machine learning

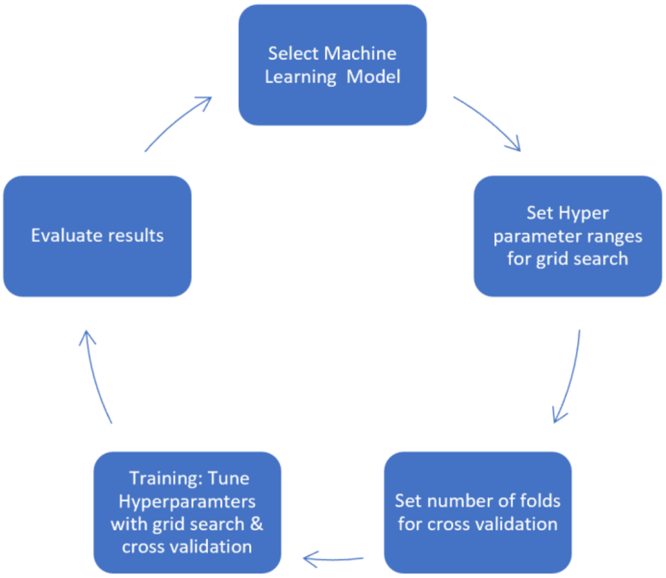
NumPy — For linear algebra and mathematics

Seaborn — For data visualization

Model/s Development and Evaluation

Testing of Identified Approaches (Algorithms)

I follow a standard development cycle for machine learning. As a beginner or even a pro, you’ll likely have to go through many iterations of the cycle before you are able to get your models working to a high standard. As you gain more experience the number of iterations will reduce (I promise!).


Select Machine Learning Model
Set Hyper parameter ranges for grid search
Set number of folds for cross validation
Training: Tune Hyperparamters with grid search & cross validation
Evaluate results

Run and Evaluate selected models

Model Selection

As mentioned at the start of the article the task is supervised machine learning. We know it's a regression task because we are being asked to predict a numerical outcome (sale price).

Therefore, I approached this problem with three machine learning models. Decision tree, random forest and gradient boosting machines. I used the decision tree as my baseline model then built on this experience to tune my candidate models. This approach saves a lot of time as decision trees are quick to train and can give you an idea of how to tune the hyperparameters for my candidate models.

Model mechanics: I will not go into too much detail about how each model works here. Instead I'll drop a one-liner and link you to articles that describe what they do "under the hood".

> **Decision Tree** — A tree algorithm used in machine learning to find patterns in data by learning decision rules.

> **Random Forest** — A type of bagging method that plays on 'the wisdom of crowds' effect. It uses multiple independent decision trees in parallel to learn from data and aggregates their predictions for an outcome.

Random forests and gradient boosting can turn individually weak decision trees into strong predictive models. They're great algorithms to use if you have small training data sets like the one we have.

Training: In machine learning training refers to the process of teaching your model using examples from your training data set. In the training stage, you'll tune your model hyperparameters.

Before we get into further detail, I wish to briefly introduce the bias-variance trade-off.

> **Model Bias** — Models that underfit the training data leading to poor predictive capacity on unseen data. Generally, the simpler the model the higher the bias.

> **Model Variance** — Models that overfit the training data leading to poor predictive capacity on unseen data. Generally, the more complexity in the model the higher the variance.

Complexity can be thought of as the number of features in the model. Model variance and model bias have an inverse relationship leading to a trade-off. There is an optimal point for model complexity that minimizes the error. We seek to establish that by tuning our hyper parameters.

Here’s a good article to help you explore this stuff in more detail.

Hyperparameters: Hyperparameters help us adjust the complexity of our model. There are some best practices on what hyperparameters one should tune for each of the models. I’ll first detail the hyperparameters, then I’ll tell you which I’ve chosen to tune for each model.

max_depth — The maximum number of nodes for a given decision tree.

max_features — The size of the subset of features to consider for splitting at a node.

n_estimators — The number of trees used for boosting or aggregation. This hyperparameter only applies to the random forest and gradient boosting machines.

learning_rate — The learning rate acts to reduce the contribution of each tree. This only applies for gradient boosting machines.

> Decision Tree — Hyperparameters tuned are the max_depth and the max_features
>
> Random Forest — The most important hyperparameters to tune are n_estimators and max_features [1].

Grid search: Choosing the range of your hyperparameters is an iterative process. With more experience you’ll begin to get a feel for what ranges to set. The good news is once you’ve chosen your possible hyperparameter ranges, grid search allows you to test the model at every combination of those ranges. I’ll talk more about this in the next section.

Cross validation: Models are trained with a 5-fold cross validation. A technique that takes the entirety of your training data, randomly splits it into train and validation data sets over 5 iterations.

You end up with 5 different training and validation data sets to build and test your models. It’s a good way to counter overfitting.

More generally, cross validation of this kind is known as k-fold cross validation. More on k-fold cross validation here.

Implementation: SciKit Learn helps us bring together hyperparameter tuning and cross validation with ease in using GridSearchCV. It gives you options to view the results of each of your training runs.

Evaluation: This is the last step in the process. Here’s where we either jump with joy or pull our hair with frustration (just kidding, we don’t do that…ever). We can use data visualisation to see the results of each of our candidate models. If we are not happy with our results, we might have to revisit our process at any of the stages from data cleaning to machine learning.

CONCLUSION

By performing different models, it was aimed to get different perspectives and eventually compared their performance. With this study, it purpose was to predict prices of used cars. With the help of the data visualizations and exploratory data analysis, the dataset was uncovered and features were explored deeply. The relation between features were examined. At the last stage, predictive models were applied to predict price of cars in an order: random forest, linear regression, ridge regression.

The increased prices of new cars and the financial incapability of the customers to buy them, Used Car sales are on a global increase. Therefore, there is an urgent need for a Used Car Price Prediction system which effectively determines the worthiness of the car using a variety of features. The proposed system will help to determine the accurate price of used car price prediction. This paper compares 3 different algorithms for machine learning: Linear Regression, Ridge Regression and Ridge Regression.

FUTURE SCOPE:

In future this machine learning model may bind with various website which can provide real time data for price prediction. Also we may add large historical data of car price which can help to improve accuracy of the machine learning model. We can build an android app as user interface for interacting with user. For better performance, we plan to judiciously design deep learning network structures, use adaptive learning rates and train on clusters of data rather than the whole dataset.