Project Report: Car Prices

Dhanashri Phalake

Abstract

Determining whether the listed price of a car is a challenging task, due to the many factors that drive a vehicle's price on the market. The focus of this project is developing machine learning models that can accurately predict the price of a 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. Our results show that Random Forest model yield the best results, but are compute heavy.

Motivation

Deciding whether a car is worth the posted price when you see listings online can be difficult. Several factors, including mileage, make, model, etc. can influence the actual worth of a car. From the perspective of a seller, it is also a dilemma to price a car appropriately[2-3]. Based on existing data, the aim is to use machine learning algorithms to develop models for predicting car prices.

Dataset

For this project, we are using dataset of given 'car_price.csv' file.

Pre-processing

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

We converted the make, fuel-type, aspiration, num-of-doors, body-style, drive-wheels, engine-location, engine type, num-of-cylinders, fuel-system, into one hot vectors.

Methodology

We utilized several classic and state-of-the-art methods, including ensemble learning techniques, with a 90% - 10% split for the training and test data. To reduce the time required for training, we used 50 examples from our dataset. Random Forest is our baseline methods. For most of the model implementations, the open-source Scikit-Learn package was used.

Random Forest

Random Forest is an ensemble learning based regression model. It uses a model called decision tree, specifically as the name suggests, multiple decision trees to generate the ensemble model which collectively produces a prediction. The benefit of this model is that the trees are produced in parallel and are relatively uncorrelated, thus producing good results as each tree is not prone to individual errors of other trees. This uncorrelated behavior is partly ensured by the use of Bootstrap Aggregation or bagging providing the randomness required to produce robust and uncorrelated trees. This model was hence chosen to account for the large number of features in the dataset and compare a bagging technique with the following gradient boosting methods.

Results

Learning Algorithm	R-2 Score on Test Data	R-2 Score on Training Data	Training Time
Random Forest	0.5851015522094232	0.9462463430729895	52.52540850639343

Future Work

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. To correct for overfitting in Random Forest, different selections of features and number of trees will be tested to check for change in performance.