Data Science 3: Big Data Analytics

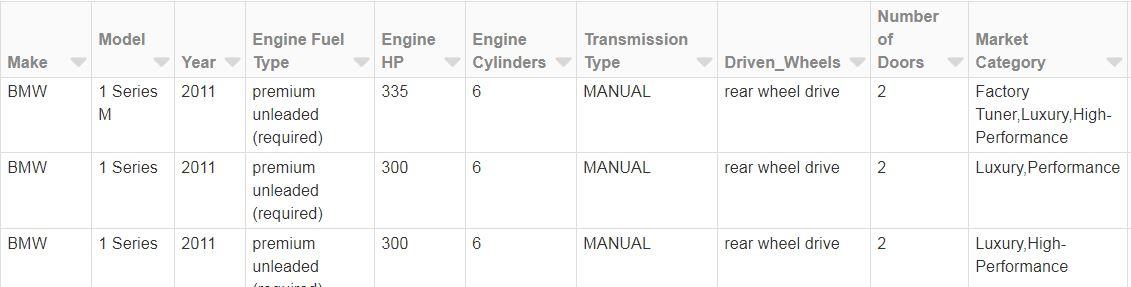
**Car Feature Predictions Using Spark and Databricks**

By Mike Coppola, Raghav Kumar, Christopher McQuarrie, Ruchika Ruchika, Jagdish Saggu

horizontal line

# Introduction

## Our group has analyzed the Car Features dataset from [Kaggle](https://www.kaggle.com/CooperUnion/cardataset). The data includes car features such as the make, model, year, engine, and other properties of the car. We have tackled the dataset using both Classification and Regression techniques. The classification models would predict whether the car belonged to the high- or low-end categories in terms of price. The regression models followed the same strategy but modeled the price of the car directly from the features. A sample data is included below.



## Support Vector Machine (SVM) Classification

A linear classifier (LinearSCV from the pyspark machine learning classification library) was chosen to predict the price class of a vehicle given the input features in the dataset. The mean of the MSRP was chosen as the classification point. Initial tuning of the classifier with all features (1000+) enabled resulted in an ROC AUC of 0.99 and an F1 score of 0.89. A fit was attempted after excluding the most influential feature Vehicle Model from the inputs. It was then observed that Vehicle Make was the most common type of feature which was influencing the classifier. Some of the other important features were: Engine Fuel Type and Market Category of Exotic, High-Performance, or Luxury. A final fit resulted in a ROC AUC of 0.98 and an F1 score of 0.88 and took approximately 2.6 minutes.

## Logistic Regression (Classification) Model

Two criteria were used: Elastic Net parameter and the maximum number of iterations to run; to get the best model possible. Each criterion had three values each, for a total of nine different models. The Elastic Net Parameter was used so that we can regularize the model depending on which of L1 or L2 loss is more predominant. This criterion was swept from 0.25 to 0.75 in increments of 0.25 (0.25, 0.5, 0.75). The maximum number of iterations was set from 10 to 30 in increments of 10 (10, 20, 30). The model that gave us the best results had an Elastic Net Parameter of 0.25 with maximum number of iterations of 30. This model was then analyzed to check the ROC curve and accuracy. The % difference between training and test data was 6.3% for AUC and 0.7% for accuracy. However, neither difference suggests the logistic regression model is overfitting the data.

## Linear Regression Model

We have created two different models one with using only the numerical values other with converting the categorical into numerical using StringIndexer, and VectorAssembler. Linear regression without usage of the categorical values gave us a bad model with only 23% accuracy. By using the stringIndexer to convert categorical data to usable points of data, we see our model is now about 82% accurate in the predictions of MSRP+/-20%. The model accuracy has decreased to about 58% to predict MSRP +/- 10% and 34% to predict MSRP +/-5%. We will try other modeling methods to improve model accuracy.

## Random Forest (RF) Model

The Random Forest Regression model was used to predict the price of a vehicle based on the given features. We had set each criterion (maximum depth, number of trees and maximum number of bins) to two different values so that we could build a parameter grid and pass off to cross validation. The model only predicted the exact price 0.028% of the time because of the large variation in price across this dataset.

## 

## Gradient Boosted Tree (GBT) Classifier

We have tried different GBT models by varying the number of iterations. The models were tested on a held-out test sample and the error in prediction was approx 4%.

## 

## Conclusion

Summarizing the performance of three classification models: SVM resulted in ROC AUC of 99%, Logistic Regression model had 92% and GBT model had 98%. Hence, SVM and GBT are the best models to choose when classifying expensive vs inexpensive cars. Logistic Regression can also be used, but it falls a few points short for predictions.

We had tried two different regression methods, Linear Regression, and Random Forest to predict the MSRP of the vehicle. Linear Regression provided an accuracy of 58% and 34% for 10% and 5% +/- MSRP respectively. Whereas, Random Forest, provided approx 46% and 24% for 10% and 5% +/- MSRP respectively. As seen above we have an 8% increase in accuracy and a 10% increase in accuracy for 10% and 5% +/- MSRP respectively. For a better prediction we can consider removing some columns which may not help in prediction of MSRP much and for Random forest increase depth and iterations.