Junior Data Analyst Test Submission

1. Conceptual

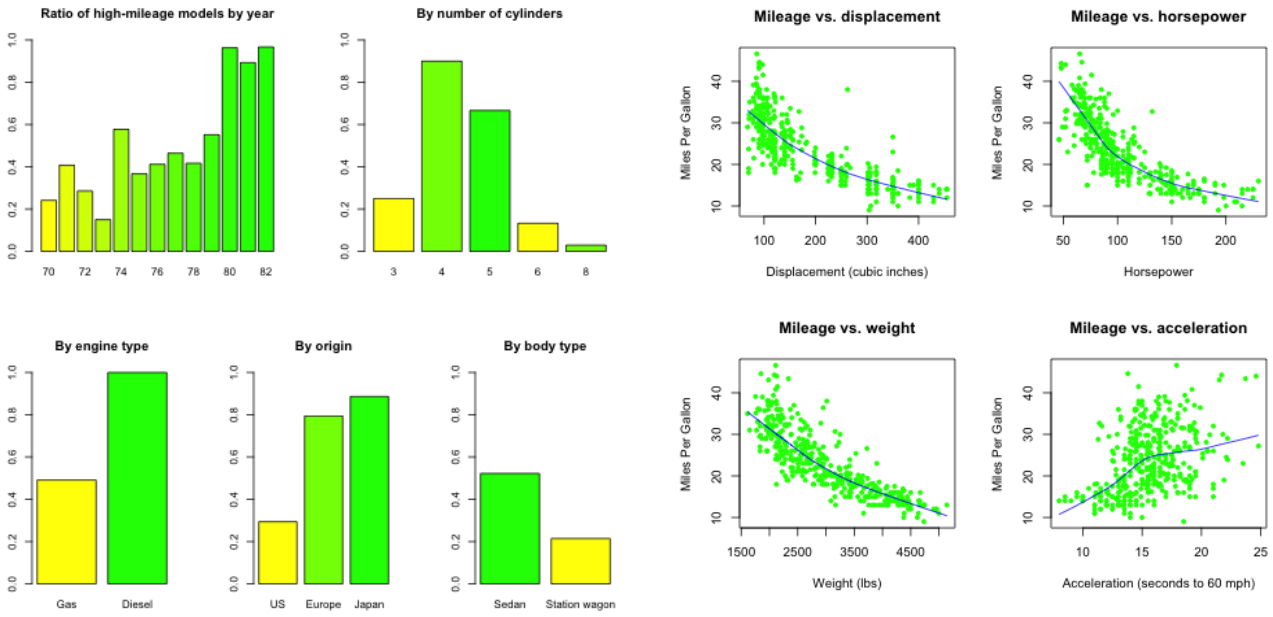
a) The objective here would be inference for which a regression approach would be most suitable. With n = 350, the predictors would be the industry, number of employees and total profit. The target variable would be the salary of the CEO. We are not trying to predict what a CEO's salary would be, but we could use the other data to build a model that predicts the known salaries as a continuous variable. After fitting the model, we could look at the weights given to the predictors to learn what their relative influence is on salary.

b) This is a binary classification problem aiming to predict whether the product would succeed or fail. Here n = 31, and the predictors are the price of each past product, competition price, marketing budget and the 10 other variables. The targets that the classifier would be trained and tested on would be a binary variable — success or failure. The trained model would then predict based on the corresponding predictor variables for the new product whether it would end up in the success or failure category.

c) Regression is the suitable method, the goal being to predict the change in the dollar based on changes in world markets. The predictors for the n = 52 samples, one for each week of the year, would be the % changes in the markets in the United States, China and France. This market data would be regressed against the recorded % change in the dollar, and the fitted model could then predict the % change in the dollar based on new market data.

2. Applied

Analysis of the data showed a few missing values and some outliers in the horsepower data. The five records with missing values were removed, but the outliers were not since only one feature was involved and all would easily be classified as low mileage. The data was a mix of numerical and categorical, and scikit-learn algorithms cannot handle categorical data unless it is one-hot encoded, which does not work so well with random forests. I thought a random forest would do well with the dataset, which contained good features it could split on, so I quickly learned enough R to take advantage of its capacity to use factors. After some data preprocessing, I generated some graphics that showed the relationships of the various features to mileage, classified as either high or low.



From the plots, it appears that the single most useful predictor is whether or not the car has a diesel engine, since all of the diesel cars in the data set get above-median mileage, although there are not so many of them. Origin and number of cylinders also look to be good predictors. Horsepower and displacement also look good, but they and weight look very much alike and seem highly correlated. Model year looks not so good overall, but models from the years '80-'82 are almost all in the high mileage category.

For the random forest, the test error rate was approximately 10%. As expected, displacement, weight and horsepower did not go well together, and the best results were obtained by keeping only horsepower, which had the highest importance ranking. Interestingly, dropping acceleration — which seemed the worst predictor — decreased performance slightly.

The gradient boosting algorithm produced better results with the same train/test split, achieving a test error of only about 8%, and also was significantly quicker. It also assigned importance to fewer features. Its eighth-ranked feature was almost zero, while the random forest algorithm had 15 features that had higher rankings. Interestingly, removing weight and horsepower *decreased* the gradient boosting performance. They ranked second and third after displacement. Below is a comparison of the feature rankings.