

Predicting The Price of Used Cars

Andrew Wright, Data Scientist

The problem

A company, called Acme Used Cars Co. has purchased a spot in a new auto mall being developed in Utah. They are expanding into a new state and need to know which cars are popular in the state to add to the inventory and how much to price them.

Data Information

I am using data from Craigslist via a dataset found on Kaggle: <https://www.kaggle.com/austinreese/craigslist-carstrucks-data>

Data Wrangling

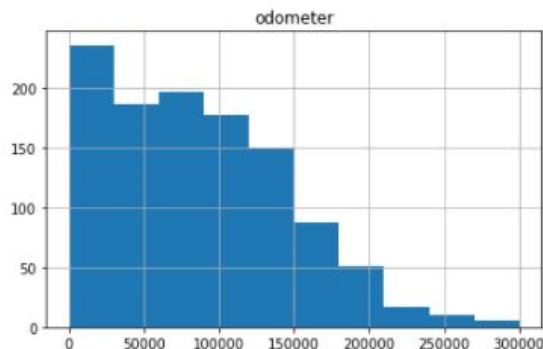
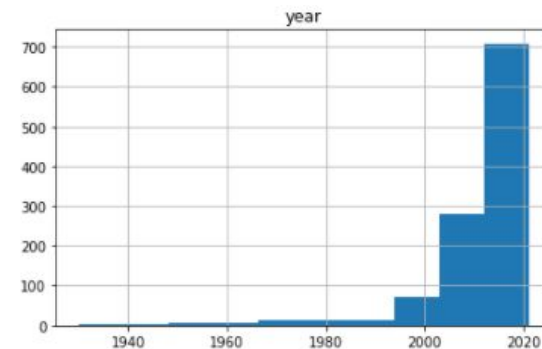
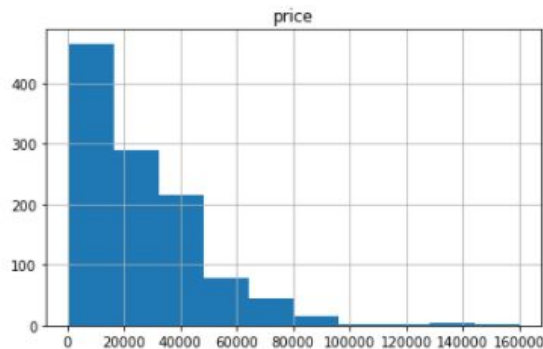
The raw dataset from contained 42,6880 rows with 22 columns, which meant that to run some of the analysis, I might need to take a smaller sample. I started with removing duplicates and looking at amount of missing values. I found that the description column had some of the missing info. Using regex, we filled in missing data with what was in the description. I then merged the dataset with another one so that there were state and region columns for analysis. I checked the outliers and had to make sure that the price, year and odometer made logical sense.

Next I pared down the data to just include the state of Utah, since that is where the business is located.

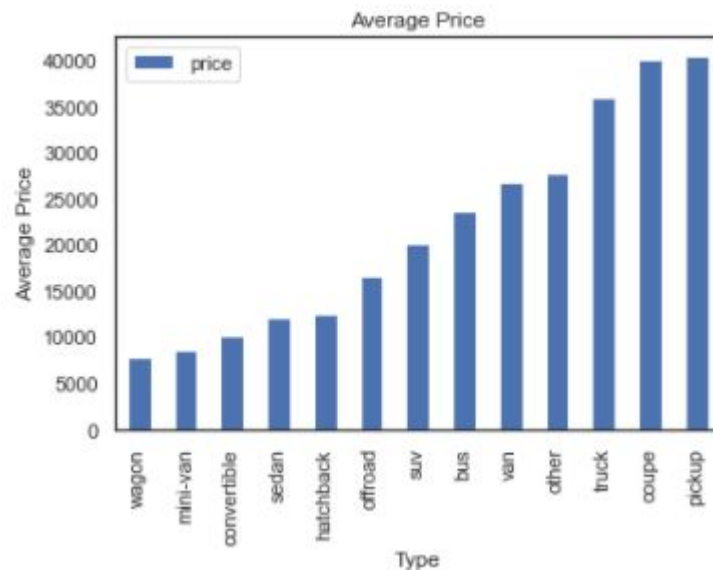
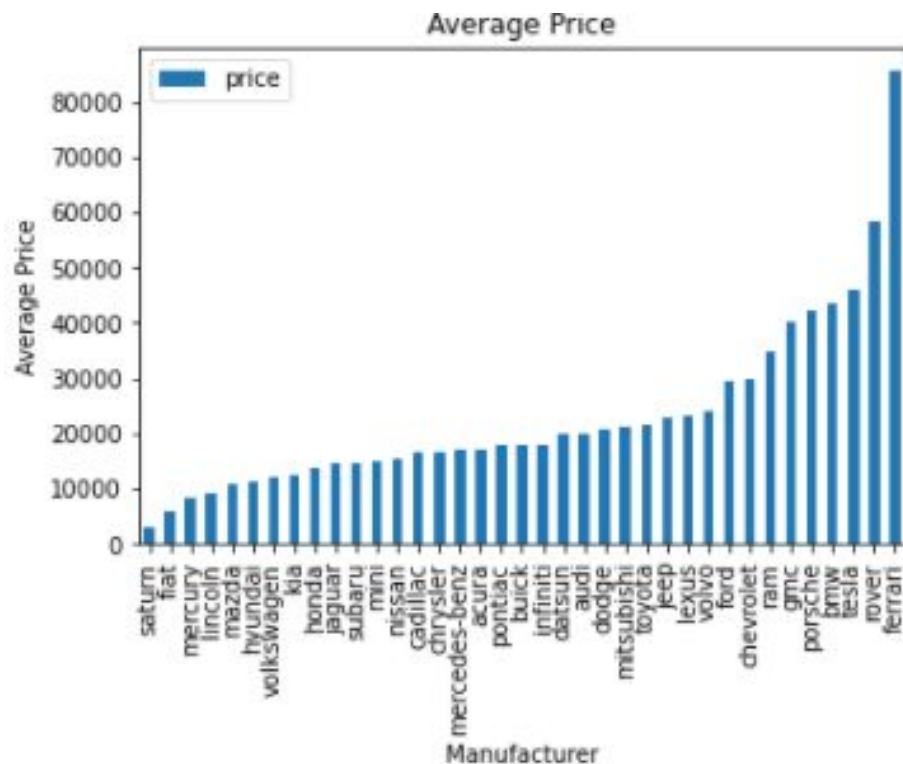
The dataset now ended up with 1,113 columns and 21 rows columns

Data Exploration

	price	year	odometer
count	1113.000000	1113.000000	1113.000000
mean	25782.264151	2010.697215	87243.246181
std	21469.142187	11.120698	59644.244936
min	500.000000	1930.000000	10.000000
25%	8550.000000	2007.000000	35967.000000
50%	20000.000000	2014.000000	77688.000000
75%	35999.000000	2017.000000	127529.000000
max	160000.000000	2021.000000	300000.000000




Data Exploration



Machine Learning Modeling

```
candidate_max_leaf_nodes = [5, 25, 50, 100, 250, 500]
scores = {leaf_size: get_mae(leaf_size, X_train, X_test, y_train, y_test) for leaf_size in candidate_max_leaf_nodes}
best_tree_size = min(scores, key=scores.get)
```

Classifier	Mean Absolute Error	Accuracy
Decision Tree	7,308	46.92%
Decision Tree (Grid Search)	460	99.86%
Decision Tree CV		54.95%
Random Forest	6,407	67.12%
Random Forest CV		63.98% 

Conclusion

The best model would be Random Forest. Even though the decision tree regressor after it was optimized its had 99% accuracy, validation only gave it 54%. Where Random forest still had 63% after cross validation.