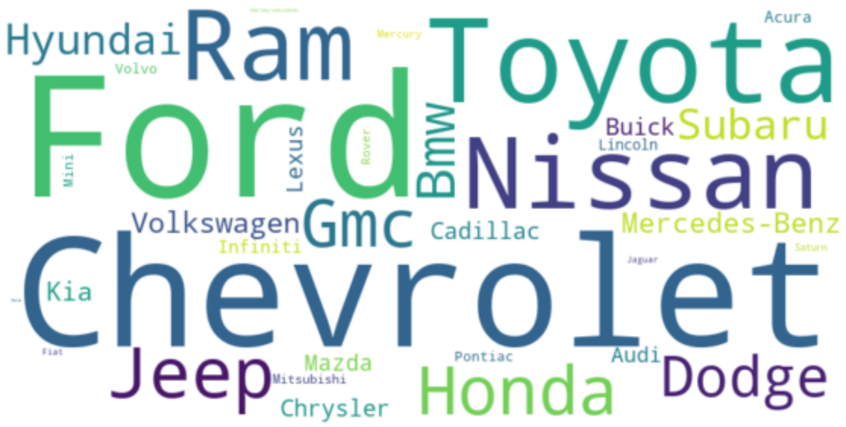
Introduction to Data Science Course Project

Cars – price prediction



WordCloud

Data Collection

My source dataset has been acquired from the website Kaggle.com. This dataset has been created by the user name "Austin Reese" who scraped Craiglist.com for several months. This dataset is composed of information about used cars that have been posted on craiglist.com by users around the US. Craiglist.com is a wonderful place to collect this kind of data since my project is dealing with price prediction and to observe the correction of the price I decided to have a dataset that it is not from an official/ business is related to this field.

Data Format Description

The dataset contains over 500K lines of items and has more than 18 categories. This set of data is being activated every few months (according to its owner Austin Reese) and recorded the data from craigslist. The entire dataset is saved as a CSV file and keeps growing every time. My topic is to observe price prediction and check the accuracy of used cars that have been advertised on craigslist. The amount of data will make it as accurate as possible by using some ML models. I have removed most of the attributes since they have nothing to do with price prediction such as 'url', 'image' and 'region'. The size of the dataset is big (more than 1.5GB) so I decided to use only a third of it, later on, I should add some more information if the results will be off than my expectation.

Cleaning the dataset and remove nan information have been done deeply with some general insights. I have noticed that there are many nan rows more particularly in attributes that have a direct impact on the price for instance 'year' and 'condition'. All nan information has been changed by using two methods:

1. The condition using odometer information to complete the status condition of the car. This method allowed me to guess the condition status of each nan fill in the 'condition' attribute by using the mean function. For example, the mean result of excellent condition was 108300 (rounded) so each car that is equal to or has more than that will be named as 'excellent' and so on.

2. Prices that were more than the average price for used cars have been removed, as well as year model that was too old and very hard to predict since those cars belong to a different category of sale (antique).

3. And finally, all nan values have been filled with the ‘ffill’ method or dropna() function.

Finally, I have found the most relevant attributes to my project that are: Finally, I have found the most relevant attributes to my project that are:

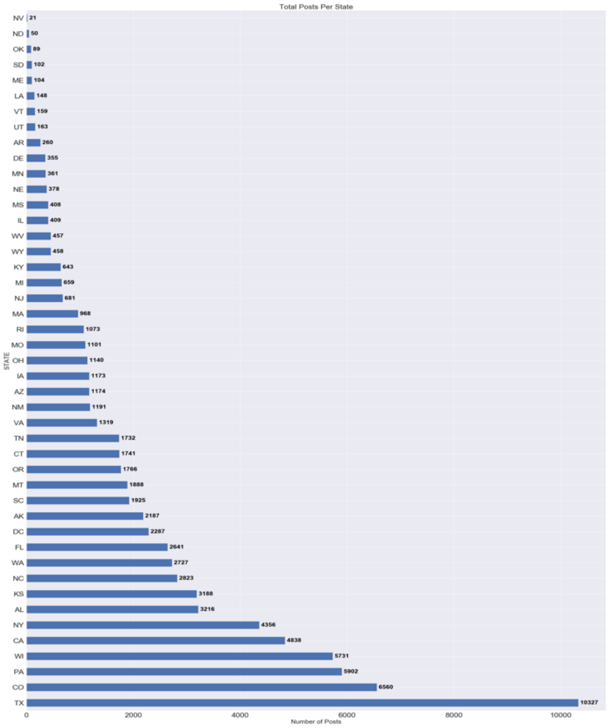
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 113341 entries, 0 to 113340
Data columns (total 13 columns):
 #   Column        Non-Null Count   Dtype  
---  ------        --------------   -----  
 0   price         113341 non-null  int64  
 1   year          113127 non-null  float64
 2   manufacturer  108349 non-null  object 
 3   model         111735 non-null  object 
 4   condition     63930 non-null   object 
 5   cylinders     68844 non-null   object 
 6   fuel          112591 non-null  object 
 7   odometer      92688 non-null   float64
 8   transmission  112497 non-null  object 
 9   drive         81049 non-null   object 
 10  type          83113 non-null   object 
 11  paint_color   77291 non-null   object 
 12  state         113341 non-null  object 
dtypes: float64(2), int64(1), object(10)
memory usage: 12.1+ MB
```

Descriptive Statistics

As I mentioned at first my project is dealing with prediction and in particular price prediction of used cars. The dataset from Craigslist has tons of information about used cars some are out of range (can be easily observed by the price – condition – year mismatch) and those were removed to work on a dataset that is accurate as much as possible. Working on this dataset and this kind of project the main factor is the price and we want to check that if the price that was given by the owners in this case in Craigslist post is accurate and matches the prediction. This can only be done by learning from many other examples.

To start to analyze my dataset I checked the range and the distribution of some of my main attributes

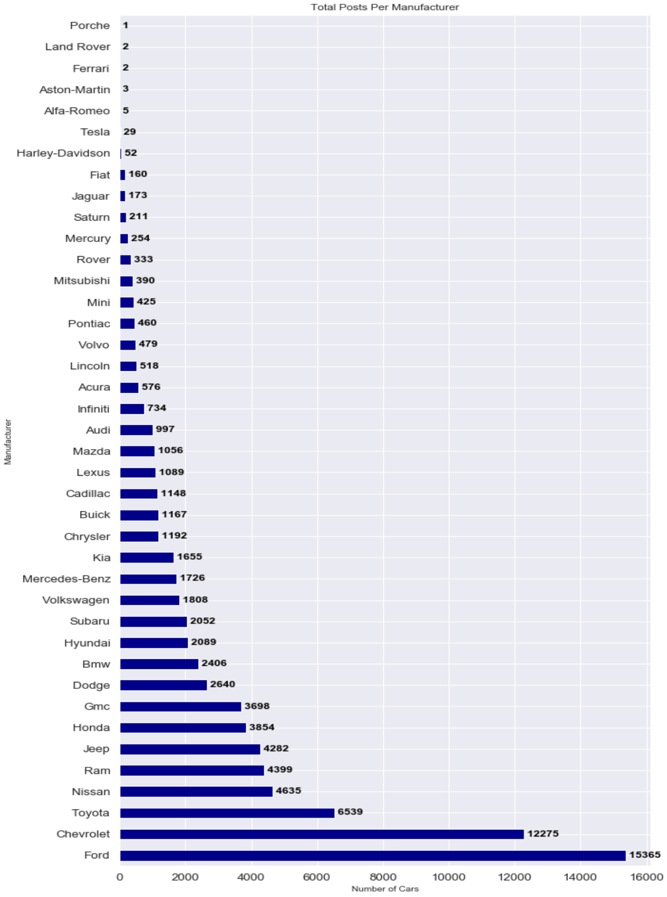
Posts per state in the US:



In this general plot we can see that the state of TX has the highest amount of posts over craigslist, next I will use this information to learn if this case has any connection between price and

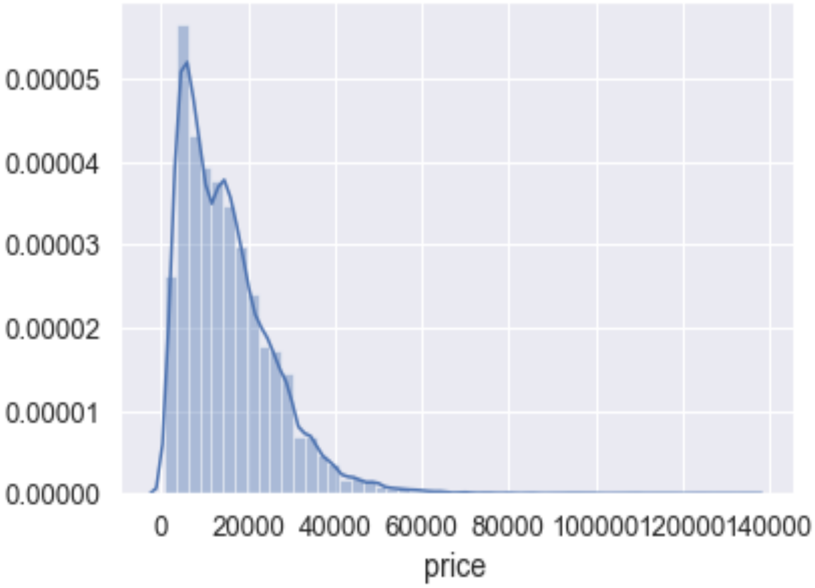
Popular car over the dataset:

With the highest volume of posts "Ford" is the most popular. This fact should give us the factor the 'Ford' value is important in this observation and should be take into consideration while working on prediction.



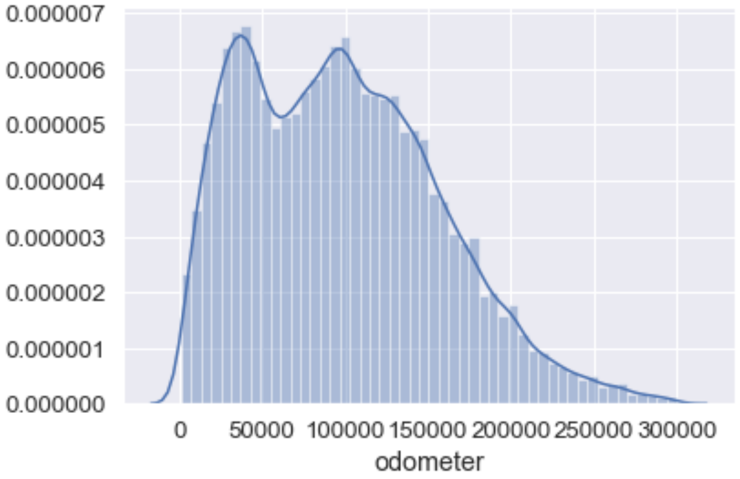
Price mean:

From this observation we can learn that the range of used cars prices is between 10K to 30K. from this plot we can see that 'price' attribute has a wide distribution. There is a peak in the price range between 10k -18K which means that most of the cars being sold on Craig list are in good



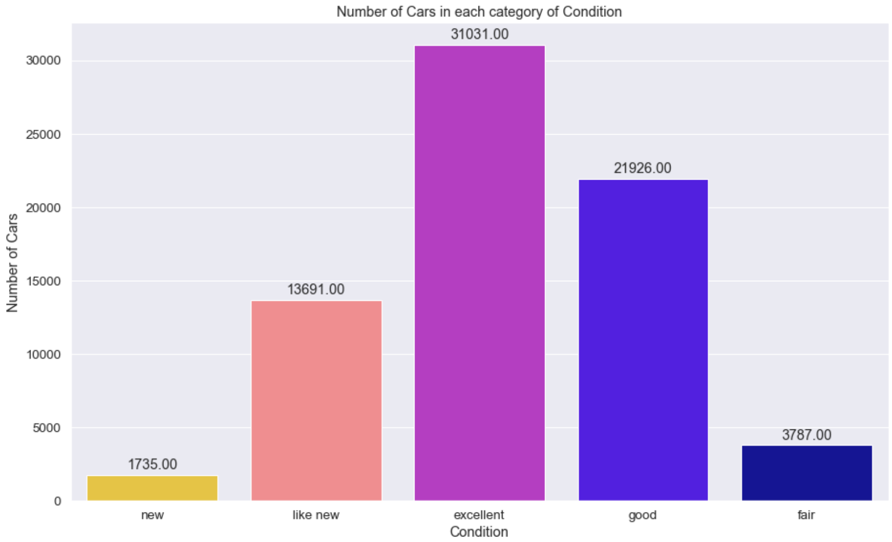
Odometer mean evaluation:

Another factor/ category that is very important when we work with used cars is the fact of how good is the car in other words how much use has been done with a particular car. From this distribution we can learn that used cars tend to be driven somewhere between 50K to 150K depends on the type, and year.



Measure car condition:

When users advertise their cars on Craigslist they want to say something about the condition of the car, this status can contribute in a different on the sale such as increase more buyers to show some interest. Also, as mentioned above this column was lack of information in many rows so the best way to complete it was according to the mileage that a cars has been driven. In this graph we can see that excellent condition is the most popular status (makes sense when we talk about used cars).



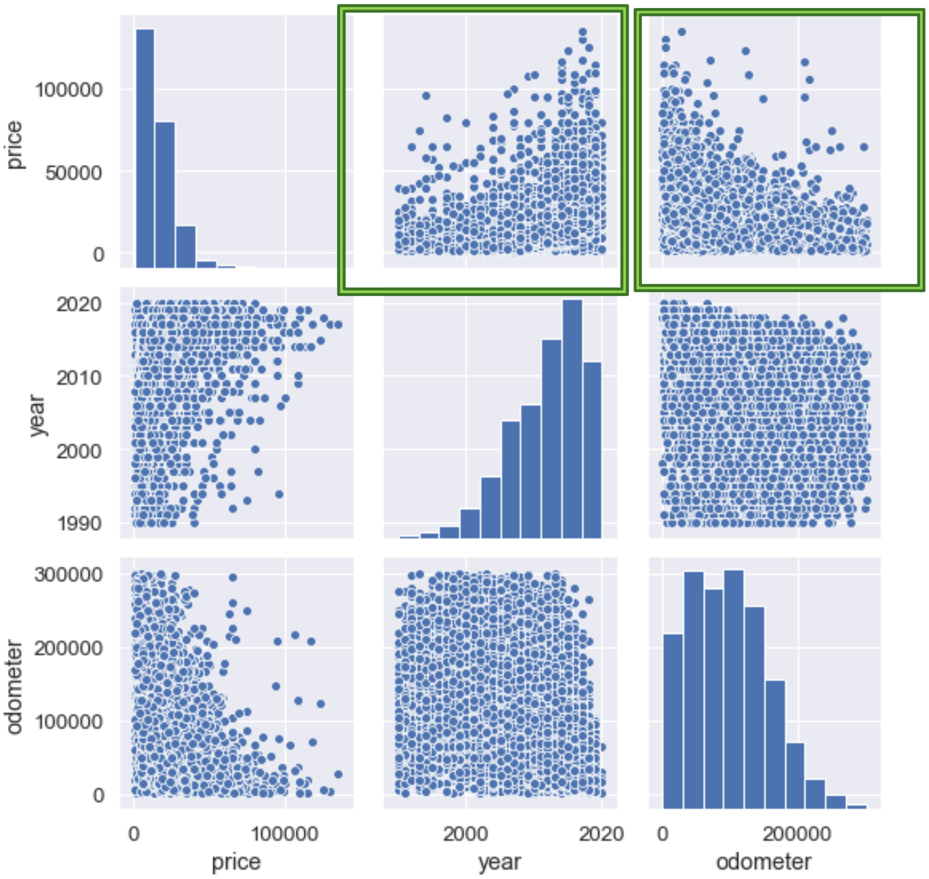
Data Analysis, Visualization, and Insights

Many interesting insights can be derived from this data, such as how many cars people hold in each state? What is the most popular car and models that is been sold over Craigslist? Is Craigslist a better platform than other digital media? How to predict if the determent price is accurate. What type of engine people like? Is it different among other states? In my project I will try to answer most of the questions above but, the main idea behind this is to predict current prices. The question is “does the price match the condition of the car?” or “What impact on the price?” When a seller tries to sell his car, he makes a guess and determines his price according to the same car that is already advertised over the media. I will try to predict how accurate is that determination about what people have on craigslist.

To understand how the prices are determined I need to see what affects the price. For this, I will explore some attributes concerning the price. Also, from my own experience I can tell that condition, odometer and year have a great impact on the price and I should start my data exploration there.

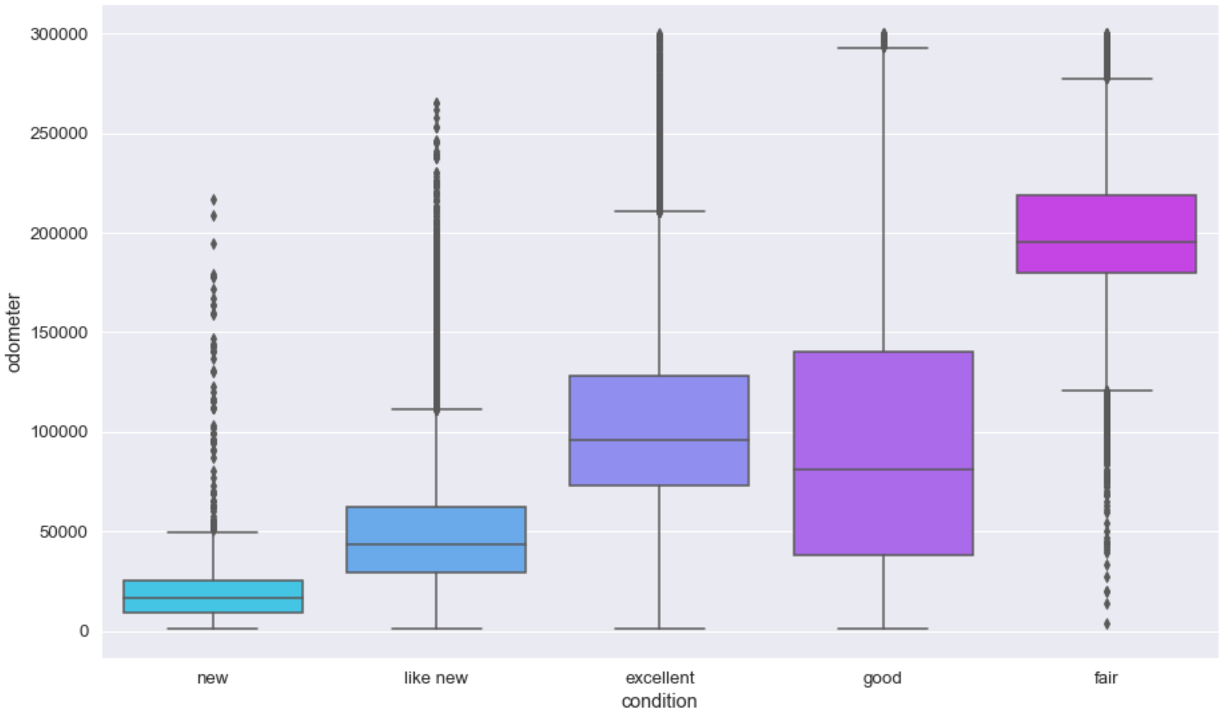
Explore features that could impact the price:

Using seaborn plot in order to explore the relation between price, odometer and year. From this graph we can easily say that the higher level of odometer the less price as well as the older car the lower price. So in fact price is related to how a specific car has been used and what is the year.



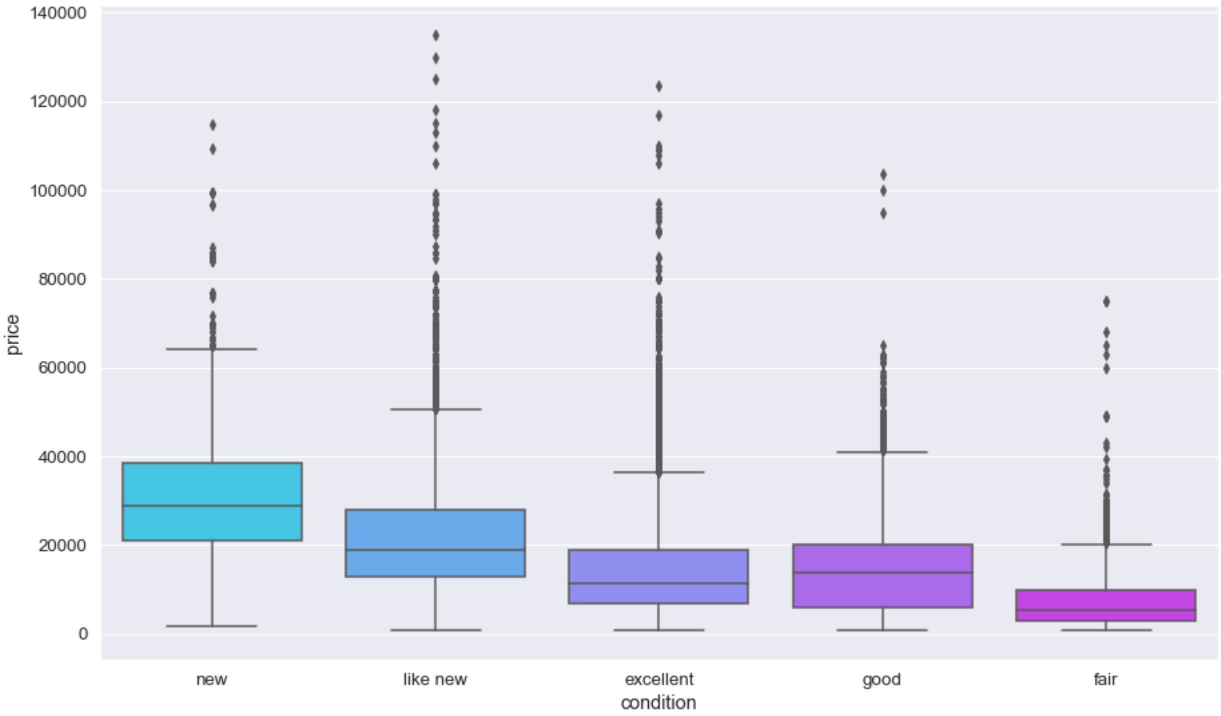
Odometer Vs Condition:

Based on the boxplot graph we can see that more than 50% of the ‘good’ condition cars have done more then 100K miles, however, people that tend to sell new cars drive much less where new cars consider as no older than 2 years. Most of the ‘fair’ condition which is ranked lowest in the condition ranking have driven more than 200K these cars are somewhere between 1990 to 2000.

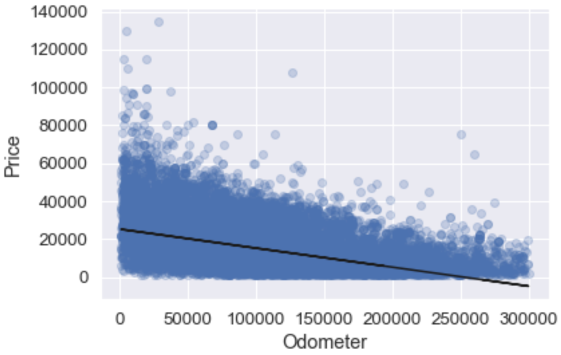


Price Vs Condition:

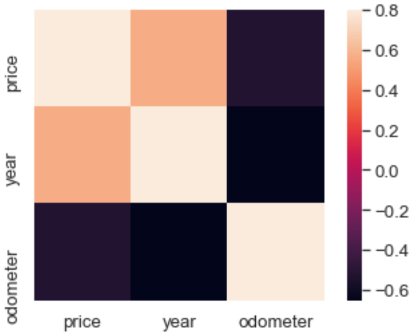
According to this box plot, we can tell that (as expected) the better condition the higher the price, however, there is slightly different in the 'new' condition where some of the new cars are low than the total advertised cars this could be because of the model or any other variable that will be explored next. In general, this graph determines condition and price where the average is somewhere between 40 – 50% which is the number as expected. From here we cannot make a strict determination so we need to explore some more factors how prices have determined by its sellers.



With the scatter plot below, we can see that most of the cars in the range of $ 0 – 20, 0000 have in average 50,000 miles. This graph allows me to put more emphasize on Odometer Vs Price when prediction is in act.



As well as the graph bellow- using heatmap to observe the correlation between the three factors (price, year and odometer) these are my main factors that will be taking into calculation.



ML Section:

Machine learning will help to analysis the data set as well as to predict the price for used cars. Machine learning has a lot to offer but in this particular analysis not all the methods could help to predict prices and its accuracy. In this section two method have analyzed:

1. Linear Regression – for continues variable (Multiple Regression model).

   In this method all the features have used in order to analyze what the accuracy of these features can contribute to the price prediction.

2. Random Forest method – to analyze what features can make the prediction more accurate that Linear Regression.

3. XGBoost trees – this method will help to improve the linear regression results and reduce some of the features that contribute less.

LinearRegression:

Starting with all features in the dataset, convert some of the columns to fit the process. In this dataset some features contain categorical details so we have to convert them in order to fit the linear regression method.

```
linreg = linear_model.LinearRegression()
linreg.fit(X_train, y_train)
# print the coeffcient
print('Coefficients: ' ,linreg.coef_)
X_pred = linreg.predict(X_test)
print(X_pred)

#Residu al and variance score:
print("Residual sum of squares: %.2f" % np.mean((X_pred - y_test) **
2))

 print('Variance score: %.2f' % linreg.score(X_test, y_test))
 Residual sum of squares: 65170937.38
 Variance score: 0.4
```

Model evaluation for LineaRegression:

As for evaluation of the linear regression we can see that including all the features from the data set is not suitable for this prediction. However, from this result can learned that there are some features that cannot be included and must be taken off. Accuracy for this model can be improved.

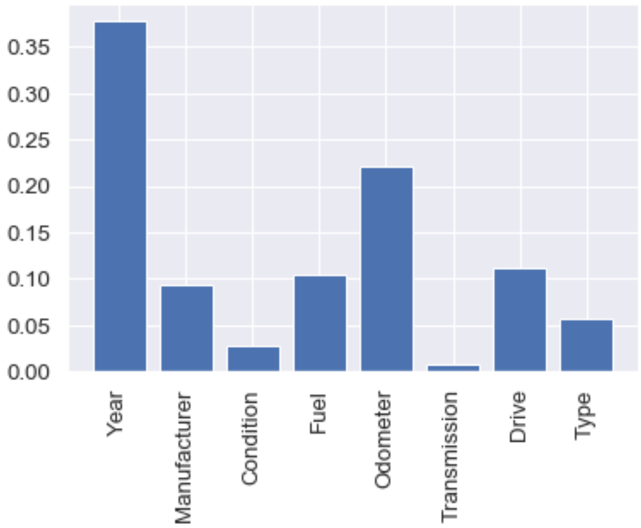
```
from sklearn import metrics
from sklearn.metrics import mean_squared_error as mse
from sklearn.metrics import accuracy_score

mae_errors = metrics.mean_absolute_error(y_test, X_pred)
print('MAE: ', mae_errors)
mse_errors = metrics.mean_squared_error(y_test, X_pred)
print('MSE: ', mse_errors)
rmse_errors = np.sqrt(mse(y_test, X_pred))
print('RMSE: ', rmse_errors)
accuracy = linreg.score(X_test,y_test)
print('Accuracy = ', accuracy*100,'%')
MAE:  5905.062563149076
MSE:  65170937.380065784
RMSE:  8072.851874032236
Accuracy =  47.03 %
```

Next, using Forest Random method to help to predict what features can be invoked to help with XGBoost prediction:

```
from sklearn.ensemble import RandomForestRegressor
from matplotlib import pyplot
model = RandomForestRegressor(n_estimators=8)
# fit the model
model.fit(X_train, y_train)
# get importance
importances_list = dict(zip(feature_names, model.feature_importances_))
# plot feature importance
plt.bar(importances_list.keys(), importances_list.values(),orientation = 'vertical')
plt.xticks(rotation='vertical')
plt.show()
# summarize feature importance
for i,v in enumerate(importances_list.values()):
print('Feature: {}, Score: {}'.format(list(importances_list.keys())[i], round(v, 3)))
```

from the random forest some of the following features have seen as the most important for this data set.

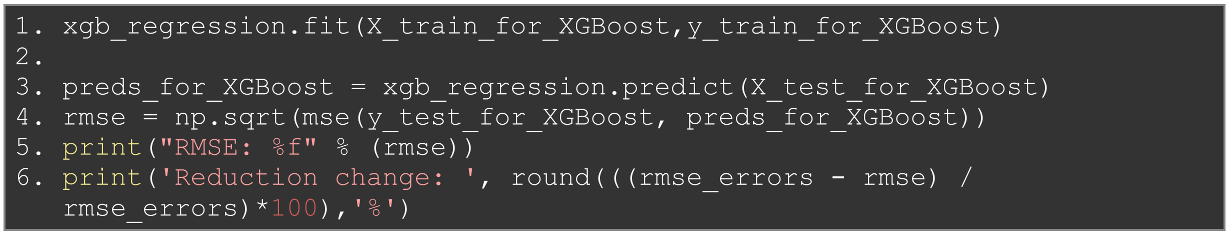


```
Feature: Year, Score: 0.377
Feature: Manufacturer, Score: 0.093
Feature: Condition, Score: 0.027
Feature: Fuel, Score: 0.105
Feature: Odometer, Score: 0.221
Feature: Transmission, Score: 0.009
Feature: Drive, Score: 0.111
Feature: Type, Score: 0.057
```

Features to be used in XGBoost to improve the prediction process: Year, Odometer, Drive, Fuel, Manufacturer.

XGBoost:

XGBoost is the best known for tree method to help solving such complicate dataset. By using the XGBoost machine learning method I will try to reduce the RMSE result and have better accuracy rate. In order to achieve the simple result and show the performance of machine learning algorithm is to use Train set and Data set. “This algorithm evaluation technique is fast. It is ideal for large datasets (millions of records) where there is strong evidence that both splits of the data are representative of the underlying problem. Because of the speed, it is useful to use this approach when the algorithm you are investigating is slow to train.” (Jason Brownlee)



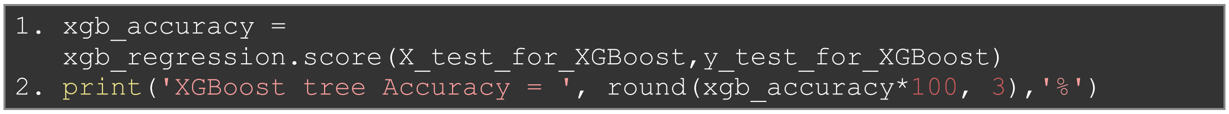
```python
xgb_regression.fit(X_train_for_XGBoost,y_train_for_XGBoost)

preds_for_XGBoost = xgb_regression.predict(X_test_for_XGBoost)
rmse = np.sqrt(mse(y_test_for_XGBoost, preds_for_XGBoost))
print("RMSE: %f" % (rmse))
print('Reduction change: ', round(((rmse_errors - rmse) / rmse_errors)*100),'%')
```

The result of this model:

```
RMSE: 6057.313155
reduction change (from previous model):  25.0 %
```

Using XGBoost have allowed to reduce the RMSE in 25%.



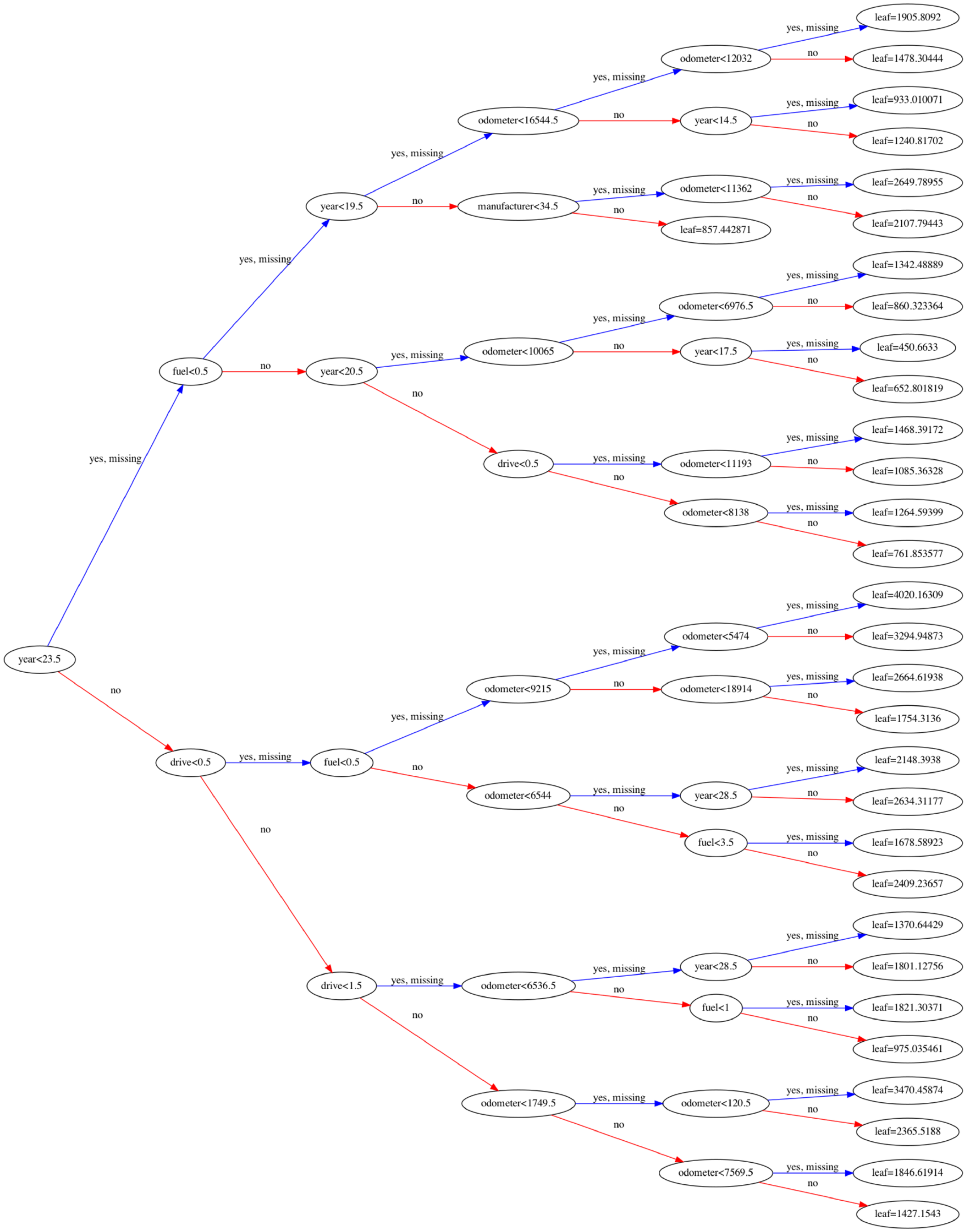
```python
xgb_accuracy = xgb_regression.score(X_test_for_XGBoost,y_test_for_XGBoost)
print('XGBoost tree Accuracy = ', round(xgb_accuracy*100, 3),'%')
```

```
 XGBoost tree Accuracy =  70.467 %
```

Indeed, there is significant change from the Linear Regression model to the XGBoost tree by reducing the number of feature (use the most 5 important) and improve the rate of accuracy.

Perhaps this is not an ideal result but this is the highest rate that this dataset has showed using different methods. This can be improved by organize the data even more and work on different feature.

XGBoost tree:



Conclusion:

This project is dealing with large dataset that acquired over time from Craigslist (credit is given). The data set was not organized because Craigslist is not specialized in selling used cars, it is best known for selling products between people but it is not supervised and organized by car dealers for example. Using Craigslist can be biased because some of the data was not correlate to real. Some work needed to be done on the data set such as provide average price for missing rows (using average price for year and condition). From the analyzing of this data we can learn that Ford is the most popular car among many states in the U.S in particular the state of TX where Ford is ranked number one among many different cars. To predict car prices I had to use some of the most interesting tool in machine learning which is XGBoost. At first, LinearRegression was the best option to learn about te data and predict the prices since this data involves continues variables. Linear Regression has produced average to bad result prediction prices. In order to improve it I decided to try XGBoost since it is well known for dealing with large data sets. The results were immediately, showing change of 25% change from Linear Regression to XGBoost. XGBoost has showed 70% accuracy which is not the best results but the improving process was better.

References:

1. Austin Reese (2020, March). Used Cars dataset: Vehicles listing from Craigslist. version 5. 14, April 2020. https://www.kaggle.com/austinreese/craigslist-carstrucks-data

2. XGBoost:

   https://xgboost.readthedocs.io/en/latest/python/python_api.html

   https://www.datacamp.com/community/tutorials/xgboost-in-python

   How to Evaluate Gradient Boosting Models with XGBoost in Python: https://machinelearningmastery.com/evaluate-gradient-boosting-models-xgboost-python/

3. Linear Regression:

   https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html