Abstract
The dataset provides more than 370000+ records of used cars information, and the table contains lots of types of information, and my question is which feature is the most important one that affect the price and which one would be the least one, I mainly used linear regression to do the analysis of my dataset.

Introduction
The contents of this report are as follow:
1 **Data Cleaning and Visualization**: This section will revolve around exploring the data and visualizing some summary statistics.
2 **Feature Engineering**: This section would select features that could do more contribution to my data analysis because some features has little relevance of our analysis.
3 **Correlation Matrix**: Get a table to directly see the relation of each column
4 **Linear regression feature**: separate data into training data and test data ,use fit and predict function to get the result.
5 **Random Forest Feature Selection**: Using the Random Forest's convenient attribute "feature_importance" to calculate and ultimately rank the feature importance.

Code with document
Feature engineering

```
from sklearn import datasets, linear_model, preprocessing, svm
from sklearn.preprocessing import StandardScaler, Normalizer
labels = ['gearbox', 'notRepairedDamage', 'model', 'brand', 'fuelType', 'vehicleType']
les = {}
#Make all the dimensions labelled
for l in labels:
   les[l] = preprocessing.LabelEncoder()
   les[l].fit(dcAutos[l])
# fit function would analyze the model argument.
   tr = les[l].transform(dcAutos[l])
   dcAutos.loc[:, l + '_feature'] = pd.Series(tr, index=dcAutos.index)
labeled = dcAutos[ ['price'
                    ,'yearOfRegistration'
                    ,'powerPS'
                    ,'kilometer'
                    ,'monthOfRegistration']
                + [x+"_feature" for x in labels]]
#
```

Preparing training data and test data

```
from sklearn.linear_model import Ridge, RidgeCV, ElasticNet, Lasso, LassoCV, LassoLarsCV
from sklearn.model_selection import cross_val_score, train_test_split
Y = labeled['price']
```

```
X = labeled.drop(['price'], axis='columns', inplace=False)
plt.rcParams['figure.figsize'] = (12.0, 6.0)
prices = pd.DataFrame({"1. Before":Y, "2. After":np.log1p(Y)})
prices.hist()
Y = np.log1p(Y)
plt.show()

def cv_rmse(model, x, y):
   r = np.sqrt(-cross_val_score(model, x, y, scoring="neg_mean_squared_error", cv = 5))
   return r
test_size = .33
#33 percent of dataset would be the test data, the rest of it would be training data
X_train, X_val, y_train, y_val = train_test_split(X, Y, test_size=test_size, random_state = 3)
print(X_train.shape, X_val.shape, y_train.shape, y_val.shape)
# this step is trying to split the dataset
r = range(2003, 2017)
```

Random Forest

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import GridSearchCV
rf = RandomForestRegressor()
#use RandomForest to training the some decision branch in the whole forest.

param_grid = { "criterion" : ["mse"]
          , "min_samples_leaf" : [3]
          , "min_samples_split" : [3]
          , "max_depth": [10]
          , "n_estimators": [500]}

gs = GridSearchCV(estimator=rf, param_grid=param_grid, cv=2, n_jobs=-1, verbose=1)
#use GridSearchCV to select optional parameters that to be used in our estimator
gs = gs.fit(X_train, y_train)
```

Feature Importance

```
importances = forest.feature_importances_
# feature_importance is an important attribute to estimate the relative importance rank in one
certain feature.
std = np.std([tree.feature_importances_ for tree in forest.estimators_],
         axis=0)
indices = np.argsort(importances)[::-1]
print("Feature importance outcome:")
for f in range(X.shape[1]):
   print("%d. feature %d (%f)" % (f + 1, indices[f], importances[indices[f]]))
```

```
print(X_train.columns.values)
plt.figure()
plt.bar(range(X.shape[1]), importances[indices],
     color="r", yerr=std[indices], align="center",tick_label = X_train.columns.values)
plt.xticks(range(X.shape[1]), indices)
plt.xlim([-1, X.shape[1]])
plt.show()
# plot the feature importance rank to get the conclusion
```

Results

```
1. feature 0 (0.666965)
2. feature 1 (0.239981)
3. feature 9 (0.026550)
4. feature 5 (0.020992)
5. feature 7 (0.018240)
6. feature 2 (0.013680)
7. feature 6 (0.010152)
8. feature 8 (0.001941)
9. feature 3 (0.000834)
10. feature 4 (0.000667)
['yearOfRegistration' 'powerPS' 'kilometer' 'monthOfRegistration'
 'gearbox_feature' 'notRepairedDamage_feature' 'model_feature'
 'brand_feature' 'fuelType_feature' 'vehicleType_feature']
```

In my conclusion, yearofRegistration has the most impotance among all the features, the second one is powerPS, and then kilometer.

Discussion

In this assignment, I tried to applied some data analysis stuff that I know, some very basic operations on this data, since my question is also a plain question, I think my solutions work for the question risen by myself. Data cleaning and munging, feature engineering and selection etc. But there still exist some issue about my analysis. First, the ML algorithms are very rough, I don't know which regression model would be most suitable one for my question, I have no idea about how precise the model that I used, Is there any other model fits my research better? Second, the research outcome is not quite satisfying, in my assumption, the kilometer and the powerPS would have much more importance than the current outcome.

Reference

- Neglected machine learning ideas
- https://www.quora.com/What-is-feature-engineering
- An Introduction to Feature Selection
- Discover Feature Engineering, How to Engineer Features and How to Get Good at It

- How valuable do you think feature selection is in machine learning? Which do you think improves accuracy more, feature selection or feature engineering?
- Data Mining Concepts and Techniques, Third Edition, Jiawei Han.
- Machine Learning in Action, Peter Harrington.