HarvardX PH125.9x Data Science CYO Project

Cheng Ming

5/17/2020

1. Overview of the project

This report is the second Capstone Project of "HarvardX PH125.9x Data Science: Capstones". In this project, I'll use several advanced regression techniques to predict house prices in NYC, using the datasets from Kaggle. The datasets can be downloaded from: https://www.kaggle.com/c/house-prices-advanced-regression-techniques. Two datasets are provided on the website: train set and test set. Since the test does not contrain SalePrice variable, we'll only use the train set so that we can calculate RMSLE. This "train set" will be separated into training set and testing set for this project. I'll use the training set to train the machine learning algorithm and use the testing set to evaluate our models. As the scale of the house prices is too large, we'll use Root Mean Squared Logarithmic Error (RMSLE) to evaluate the accuracy of the predicted results, instead of using RMSE. In a nutshell, the objective of this project is to predict the value of SalePrice using other variables in the dataset while minimize the RMSLE.

2. Steps of the project

The following steps will be performed: (1) Data Cleaning: Load and mutate the dataset (2) Data EXploration and Visualization: Understand the dataset structure with an exploratory data analysis (3) Machine Learning Algorithms Evaluation: Train a machine learning algorithm to predict SalePrice (4) Results Analysis: Compare RMSLE derived from different models

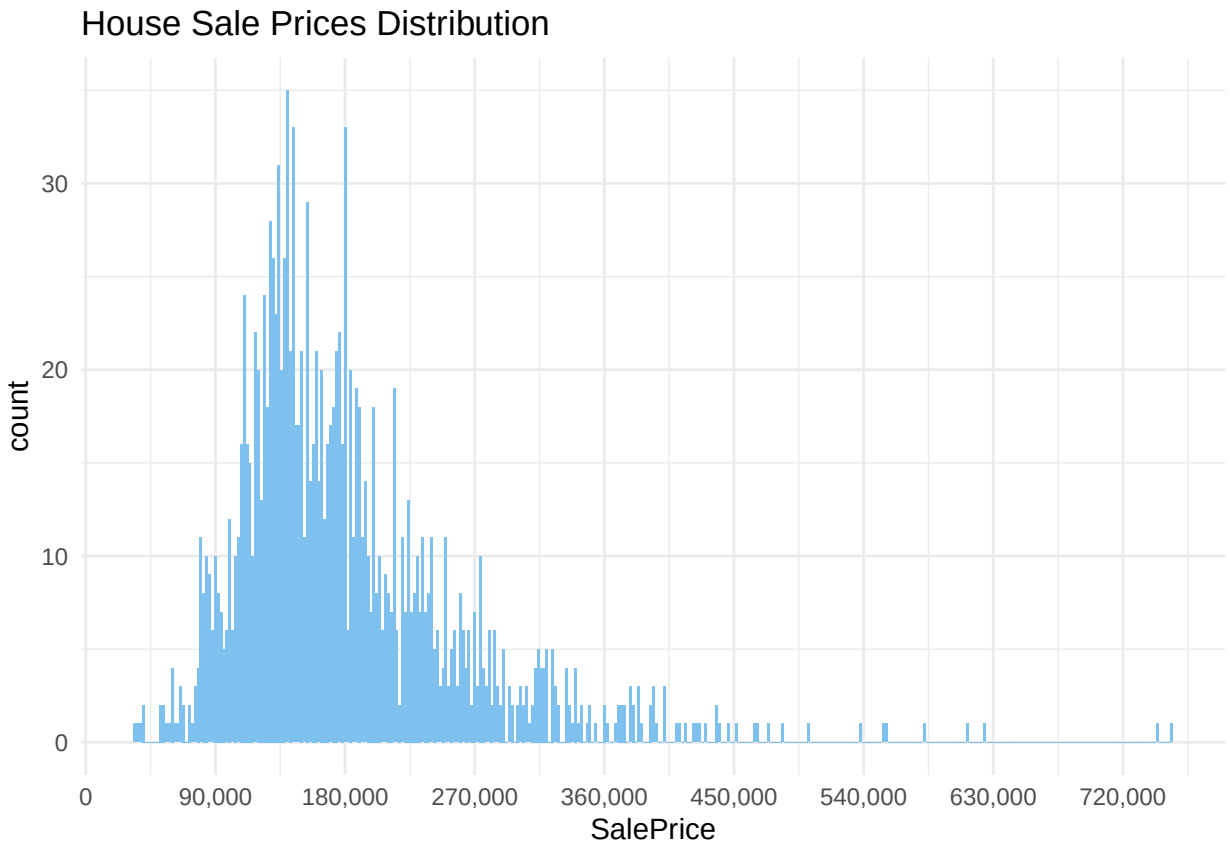
2.1 Data Cleaning: Load and mutate the dataset

The first step is to load relevant packages needed. To make this report more succinct, I've hiden this r chunk.

2.2 Data Exploration and Visualization: Understand the dataset structure with an exploratory data analysis

Let's take a look at how is the SalePrice distributed

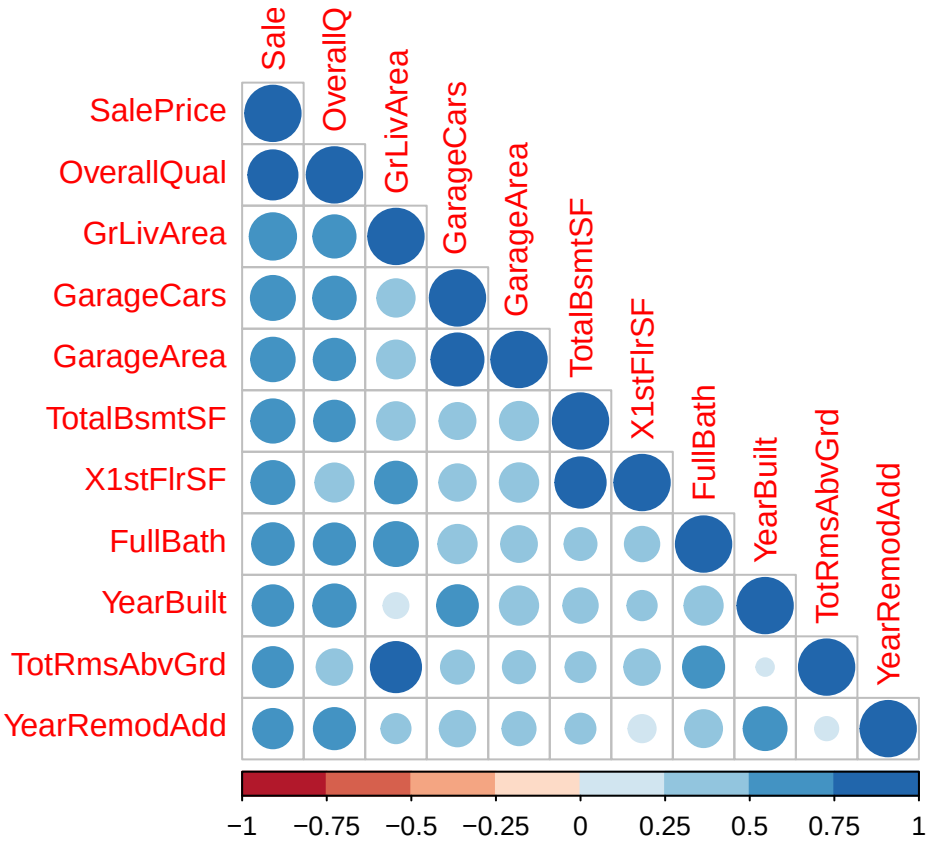
```
ggplot(data=train, aes(x=SalePrice)) +
  geom_histogram(fill="skyblue2", binwidth = 2000) +
  scale_x_continuous(breaks= seq(0, 1000000, by=90000),labels = comma) +
  labs(title="House Sale Prices Distribution")+
  theme_minimal()
```



We can see that SalePrice in the dataset is right skewed.

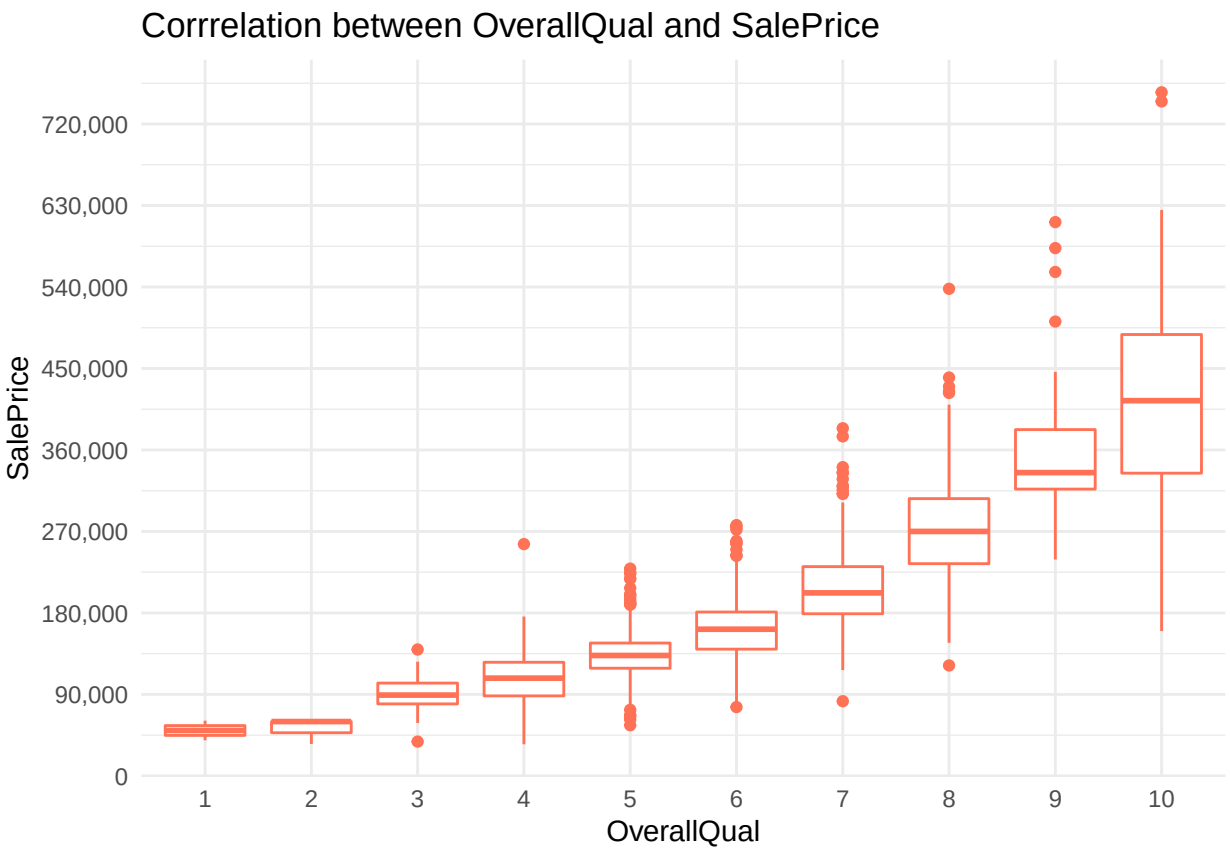
There're 80 more variables in the dataset other than SalePrice, how should we know which variables are useful for predicting SalePrice? A correlation matrix is helpful for solving such query.

```
# Identify index vector of numeric variables
numeric_vars<- which(sapply(training_set, is.numeric))

# Identify the column names of numeric variables
numeric_vars_col<-data.table(names(numeric_vars))
temp<-training_set[,numeric_vars]

# Get the correlations of all numeric variables
corr_all<-cor(temp,use="pairwise.complete.obs")

# Sort the correlations with SalePrice on a descending order
corr_sorted<-as.matrix(sort(corr_all[,'SalePrice'], decreasing = TRUE))

# Select high corelations only
high_corr<-names(which(apply(corr_sorted, 1, function(x) abs(x)>0.5)))
high_corr_col<-data.table(high_corr)
temp2<- corr_all[high_corr, high_corr]

# Plot the correlations
corrplot(temp2, addrect = 2, type = "lower", col = brewer.pal(n = 8, name = "RdBu"))
```



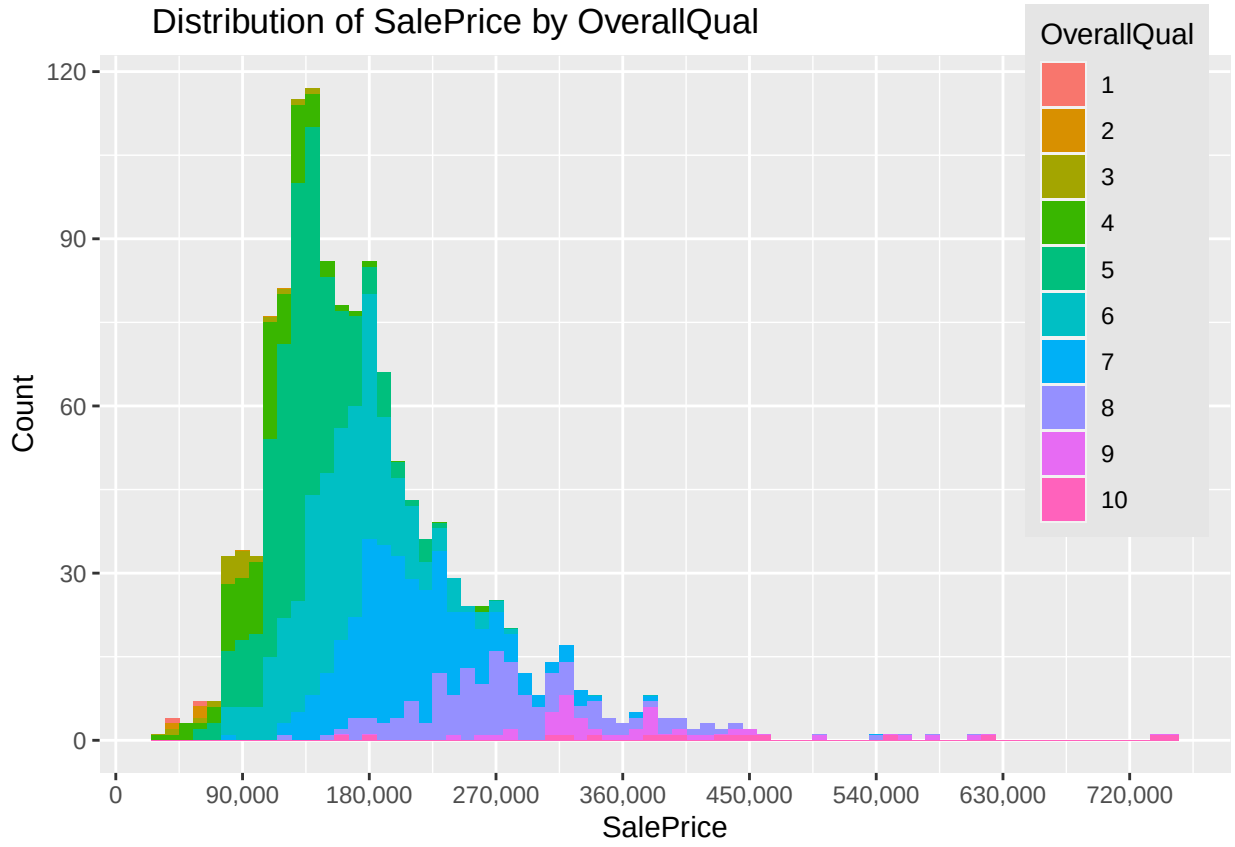
As shown in the correlation matrix table above, there're 10 numeric variables in the training dataset that have correlations above 0.5. Among these 10 numeric variables, OverallQual has the highest correlation of 0.8. We can see from the box plot below that there's indeed an obvious correlation between OverallQual and SalePrice.

```
training_set%>%ggplot(aes(x=factor(OverallQual),y=SalePrice))+
  geom_boxplot(col='coral1')+
  labs(x='OverallQual')+
  scale_y_continuous(breaks= seq(0, 1000000, by=90000),labels = comma)+
  labs(title="Corrrelation between OverallQual and SalePrice")+
  theme_minimal()
```



And the distribution of SalePrice by OverallQual is shown below:

```
training_set%>%ggplot(aes(x = SalePrice,fill = as.factor(OverallQual)))+
  geom_histogram(position = "stack", binwidth = 10000)+
  ggtitle("Distribution of SalePrice by OverallQual")+
  ylab("Count")+
  xlab("SalePrice")+
  scale_x_continuous(breaks= seq(0, 1000000, by=90000),labels = comma)+
  scale_fill_discrete(name="OverallQual")+
  theme(plot.title = element_text(hjust = 0.1),
        legend.position=c(0.9,0.7),
        legend.background = element_rect(fill="grey90",size=0.5,linetype="solid"))
```



2.3 Machine Learning Algorithms Evaluation: Train a machine learning algorithm to predict SalePrice

Model 1: Multiple Linear Regression Model

First, we'll use the 10 most relavant numeric variables identified previously to run a Multiple Linear Regression model. Using this model, we'll then predict SalePrice in the testing set.

```
# Run the Multiple Linear Regression Model
fit<-lm(SalePrice~OverallQual+GrLivArea+GarageCars+GarageArea+TotalBsmtSF+X1stFlrSF+FullBath+TotRmsAbvG

summary(fit)
```

```
##
## Call:
## lm(formula = SalePrice ~ OverallQual + GrLivArea + GarageCars +
##     GarageArea + TotalBsmtSF + X1stFlrSF + FullBath + TotRmsAbvGrd +
##     YearBuilt + YearRemodAdd, data = training_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -481549  -19463   -2258   15912  292073
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  -1.237e+06  1.391e+05  -8.897  < 2e-16 ***
## OverallQual   1.950e+04  1.271e+03  15.339  < 2e-16 ***
## GrLivArea     5.156e+01  4.460e+00  11.560  < 2e-16 ***
## GarageCars    1.118e+04  3.298e+03   3.388 0.000725 ***
## GarageArea    1.249e+01  1.116e+01   1.119 0.263174
## TotalBsmtSF   1.784e+01  4.602e+00   3.877 0.000111 ***
## X1stFlrSF     1.496e+01  5.327e+00   2.809 0.005049 **
## FullBath     -7.238e+03  2.906e+03  -2.491 0.012877 *
## TotRmsAbvGrd -9.744e+01  1.185e+03  -0.082 0.934488
## YearBuilt     3.098e+02  5.487e+01   5.647 2.01e-08 ***
## YearRemodAdd  2.825e+02  6.847e+01   4.126 3.93e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38530 on 1301 degrees of freedom
## Multiple R-squared:  0.7663, Adjusted R-squared:  0.7645
## F-statistic: 426.7 on 10 and 1301 DF,  p-value: < 2.2e-16
```

```
# Predict SalePrice in testing set using the model
SalePrice_prediction<-predict(fit,testing_set)

# Scale the data
log_SalePrice_Prediction<-log(SalePrice_prediction)
log_SalePrice_Real<-log(testing_set$SalePrice)

#Calculate RMSLE
RMSLE<-RMSE(log_SalePrice_Prediction,log_SalePrice_Real)
RMSLE
```

```
## [1] 0.2053125
```

Model 2: Random Forests Model

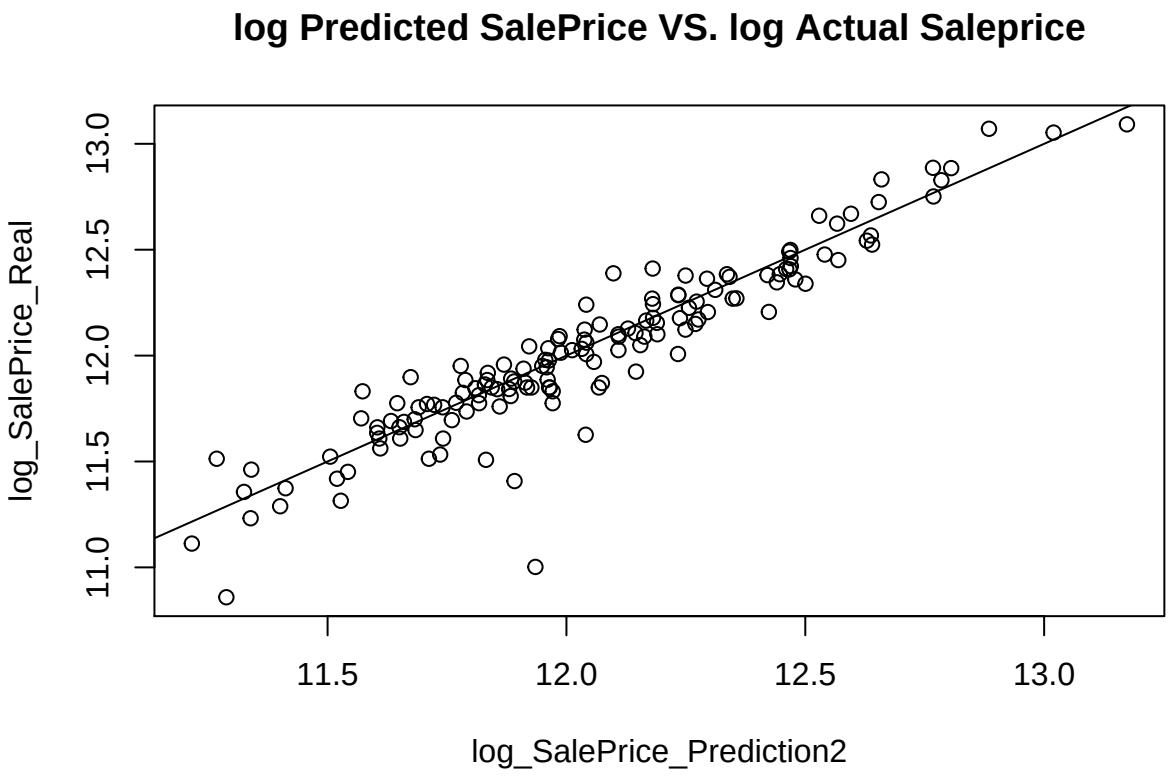
Next, we'll use Random Forests Model to predict the SalePrice. Random forests Model is an ensemble learning method that can be used for classification and regression. Random forests solve the overfitting habit of decision trees.

```
rf<-randomForest(SalePrice~OverallQual+GrLivArea+GarageCars+GarageArea+TotalBsmtSF+X1stFlrSF+FullBath+T
SalePrice_prediction2<-predict(rf,testing_set)

log_SalePrice_Prediction2<-log(SalePrice_prediction2)
log_SalePrice_Real<-log(testing_set$SalePrice)
RMSLE2<- RMSE(log_SalePrice_Real,log_SalePrice_Prediction2)
RMSLE2
```

```
## [1] 0.1438977
```

The graph below plots the relationship between log predicted SalePrice and log actual Saleprice

```
plot(log_SalePrice_Prediction2,log_SalePrice_Real,main = "log Predicted SalePrice VS. log Actual Salepr
```




```
## integer(0)
```

2.4 Results Analysis: Compare RMSLE results derived from different models

```
bind_rows(tibble(Method="Multiple Linear Regression Model",RMSLE=RMSE(log_SalePrice_Prediction,log_Sale
          tibble(Method="Random Forests Model",RMSLE=RMSE(log_SalePrice_Real,log_SalePrice_Prediction2))
```

```
## # A tibble: 2 x 2
##   Method                          RMSLE
##   <chr>                           <dbl>
## 1 Multiple Linear Regression Model 0.205
## 2 Random Forests Model             0.144
```

3. Conclusions

Based on the two RMSLE derived above, we know that Random Forests Model is better than Multiple Linear Regression Model at predcting SalePrice, using the top 10 most relevant variables from the dataset.

4. Limitations

In our model, only the top 10 most relevant numeric variables are used to predict SalePrice. However, there're many other factors that may affect SalePrice, such as YrSold(Year Sold) and SaleCondition(Condition of sale). More variables can be included in a futher analysis.

My Harvard_Data_Science_House Prices Github repository can be found at https://github.com/somsomcheng/HarvardX-PH125.9x-Data-Science-CYO-Project

Thank you so much for spending time reading my project! Really appreciate it :)