USE CASE STUDY REPORT

Group No.: Group 14

Student Names: Shruthi Subbaiah Machimada, Hemanth Lakshman Raju

Executive Summary:

The goal of this study is to build a predictive model that predicts the price of Airbnbs in Boston. The data was obtained from Airbnb open data and the dataset was selected which had the relevant variables for the predictions.

For data processing, we firstly identified and eliminated the unwanted variables. Then we converted the categorical variables to numerical variables by factorizing them. This was done for all the different amenities. For further dimension reduction, we used principal component analysis. We created different visualizations to observe our findings.

The data mining techniques chosen were ‘linear regression without PCA’, ‘linear regression with PCA’, ‘decision trees’ and ‘random forests’.

We found that PCA was very effective for dimension reduction as we had a lot of predictors. Random forest was the best model as the RMSE was the least among all the models.
This model can be used by an Airbnb host to gauge the pricing of his house and how it stands in the market alongside the competitors. Based on the generated predicted price, they can price their house accordingly.

I. Background and Introduction

Airbnb is a popular online marketplace, enabling people to list or rent private spaces. It links hosts and visitors, by sharing details of the space, price, and allowing them to communicate. However, many hosts and visitors do not have an idea of what a fair price would be, for a listing. To avoid scams, we will be predicting the price for a listing based on its specifications, for reference.

- The problem – To develop a predictive model to predict the rental prices for Airbnbs in Boston.
- The goal of the study - Perform an exploratory analysis of our data and select variables from a set like availability, date, neighborhood, space, descriptions provide, location, reviews, etc. and build a predictive model.
- The possible solution - Determine the price of the rentals and also the variables that affect the price of a rental, and by how much.

II. Data Exploration and Visualization

On observing the listings.csv file, we saw many irrelevant variables that described the web scraping details, urls, pictures, etc. which are not relevant to our outcome and cannot be used to conduct any exploratory analysis or predictions because of the nature of their values.

We first considered only the numerical variables which are relevant to our analysis. These variables are: host_listings_count, host_total_listings_count, accommodates, bathrooms,bedrooms,beds, price, guests_included, review_scores_rating, number_of_reviews. We then considered the categorical variables, such as host_is_superhost, host_has_profile_pic, property_type, host_identity_verified, bed_type, amenities, requires_license, instant_bookable, require_guest_profile_picture, require_guest_phone_verification, cancellation_policy. We created dummy variables for each of them.

The review_scores_rating column had a lot of NULL values, however since this variable was only describing the review scores, it had nothing to do with our outcome variable so we deleted the variable from our dataframe. We eliminated the rows with Null values for number of bedrooms, bathrooms because there was no way of substituting these values with any value that would not affect the outcome, and also there were very few rows with such values.

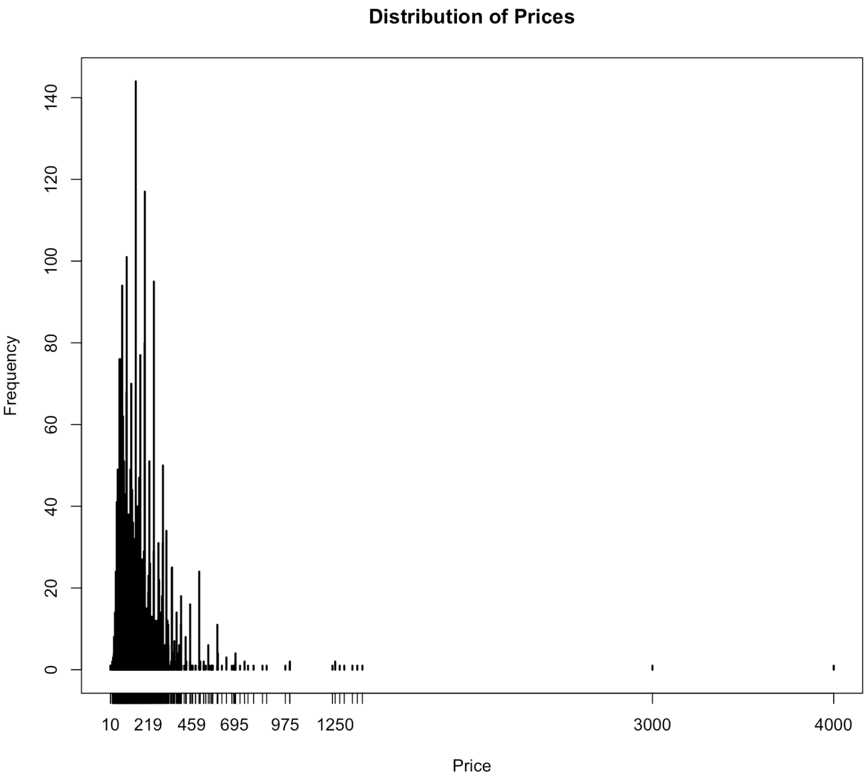
The Price column had values in the format $100, we removed the '$' from every row and converted the value to a numeric value.

We checked the statistics such as mean, median, min and max values to determine whether we had values that were outliers or too far from the rest of our values, and eliminated them.

We created visualizations to see how the Price (outcome variable) was affected by the predictors. We observed that some predictors, such as Number of bedrooms, Total occupancy, the presence of bathrooms, etc increased the price of the rentals.

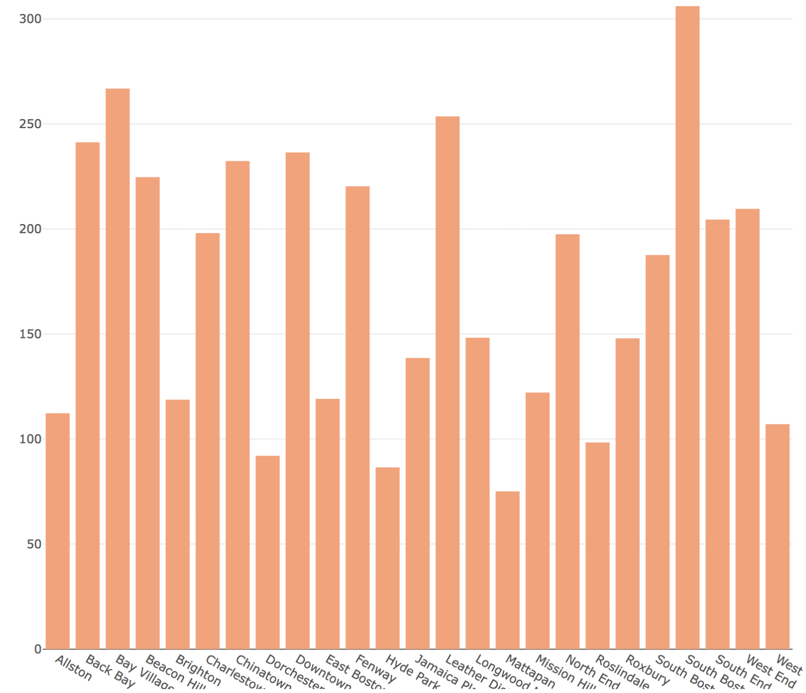
We first saw how the prices were distributed in our dataset. The below figure shows the price distribution.

Distribution of Price:



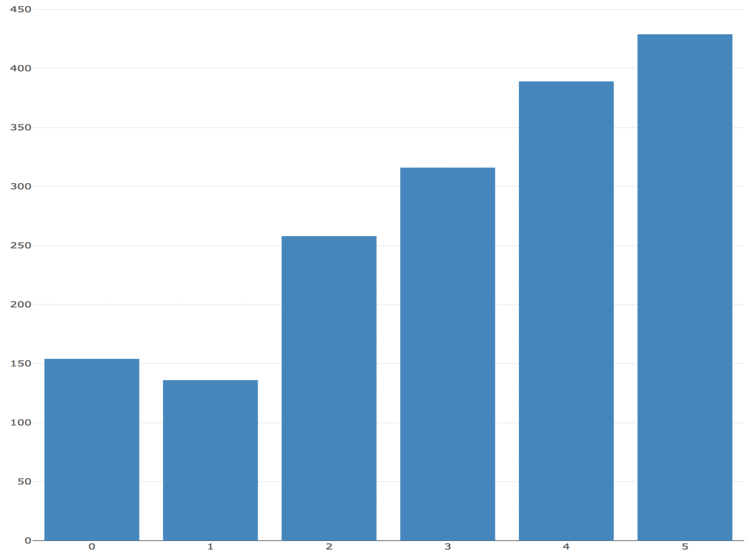
We made some analysis on how the prices are distributed over various neighborhoods. We calculated the average price of rentals for each neighborhood and plotted it in the below diagram. As we can see, few places are more expensive than others and this is an important variable that affects the price.
However, some variables such as city, country, etc are not needed and so they were eliminated.

<u>Average prices for each Neighborhood:</u>

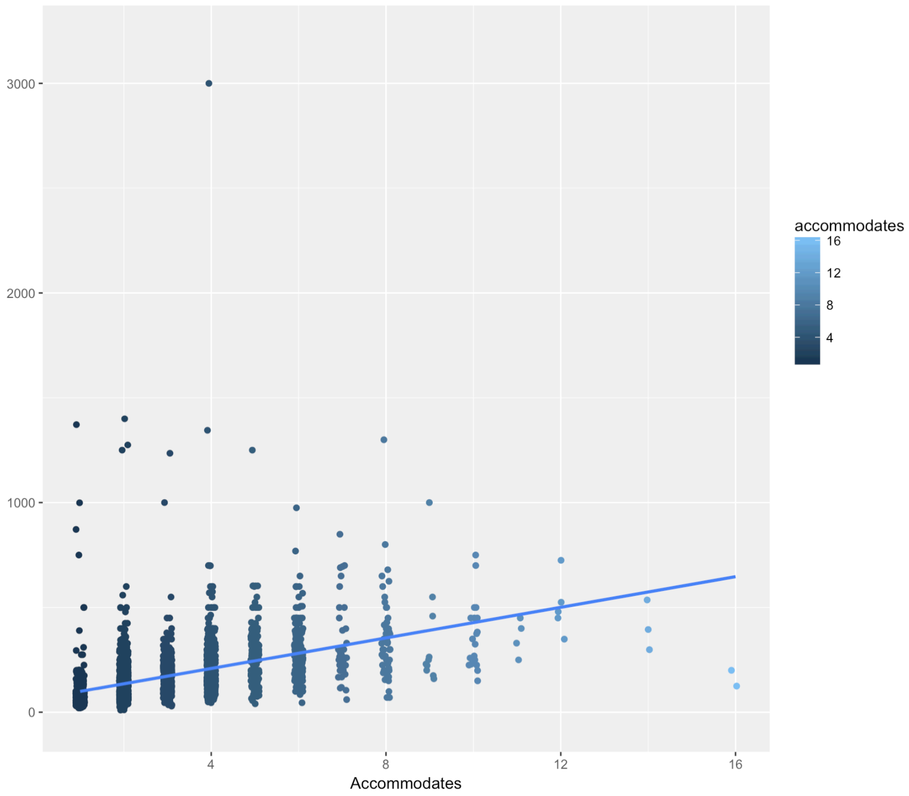


The number of bedrooms, bathrooms, number of people that can accommodate the rental, the bed type, room type, property type and the presence of amenities like kitchen, TV, washer, etc are all factors which contribute to the change in the price of the rentals. We have plotted these variables to observe how the price is affected by them. A few figures are given below.

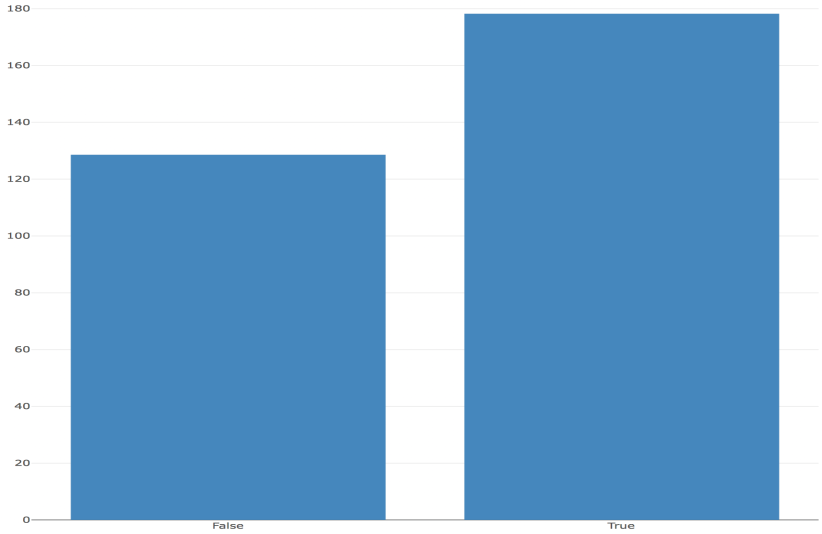
Prices over different Number of bedrooms:



Scatterplot of Accomodates and Price:

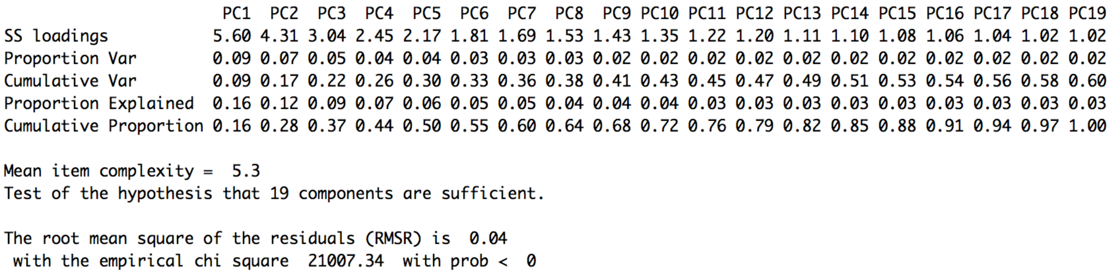


Prices over the Presence or absence of a Kitchen:



III. Data Preparation and Preprocessing

Originally, our dataset had over 100 predictors, of which most were irrelevant to our study as they described the web scraping details, urls, pictures, etc. We eliminated these variables. We then converted our categorical variables which had non-numeric values to factors and created separate variables for different amenities, which was in the form of a list in our dataset. This brought our total predictors to around 70.
Using the correlation matrix, we further removed a few unnecessary variables such as guest_verification, requires_license, other.pets, etc. We used then Principal component analysis for further dimension reduction. We retrieved 19 PCs, accounting for a total variance of 60%.



```
                       PC1  PC2  PC3  PC4  PC5  PC6  PC7  PC8  PC9 PC10 PC11 PC12 PC13 PC14 PC15 PC16 PC17 PC18 PC19
SS loadings           5.60 4.31 3.04 2.45 2.17 1.81 1.69 1.53 1.43 1.35 1.22 1.20 1.11 1.10 1.08 1.06 1.04 1.02 1.02
Proportion Var        0.09 0.07 0.05 0.04 0.04 0.03 0.03 0.03 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02
Cumulative Var        0.09 0.17 0.22 0.26 0.30 0.33 0.36 0.38 0.41 0.43 0.45 0.47 0.49 0.51 0.53 0.54 0.56 0.58 0.60
Proportion Explained  0.16 0.12 0.09 0.07 0.06 0.05 0.05 0.04 0.04 0.04 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.03
Cumulative Proportion 0.16 0.28 0.37 0.44 0.50 0.55 0.60 0.64 0.68 0.72 0.76 0.79 0.82 0.85 0.88 0.91 0.94 0.97 1.00

Mean item complexity =  5.3
Test of the hypothesis that 19 components are sufficient.

The root mean square of the residuals (RMSR) is  0.04
 with the empirical chi square  21007.34  with prob <  0
```

IV. Data Mining Techniques and Implementation

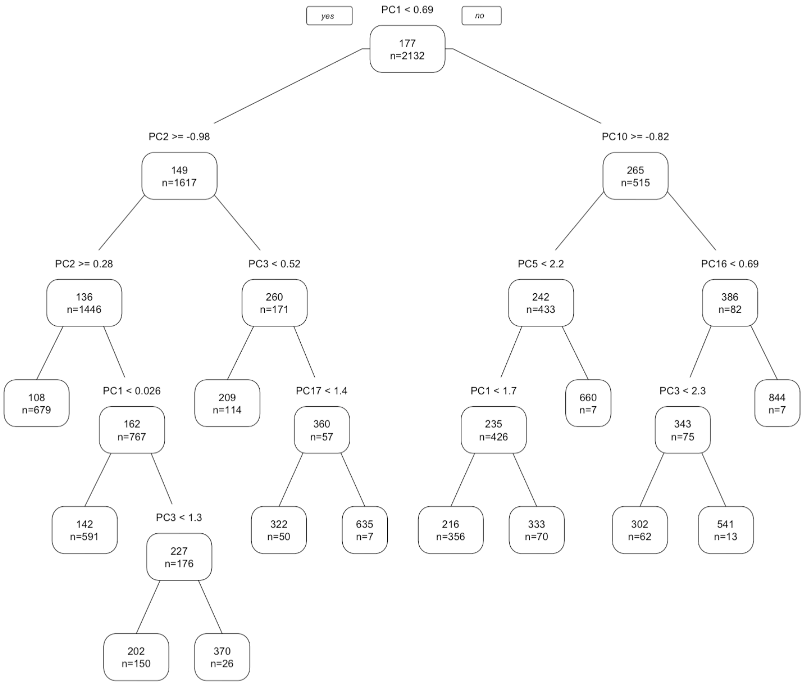
For the Prediction of Airbnb prices, we explored several models which we could use such as :

Linear regression without PCA:
We used the dataset, without the dimension reduction, divided into 60% training and 40 % validation sets. We fit a linear regression model and predicted the prices of the validation set.

Linear regression with PCA:
We then used our 19 principal components as in variables and price as outcome as the new dataset. We divided this dataset into 60% training and 40 % validation sets. We fit a linear regression model and predicted the prices of the validation set.

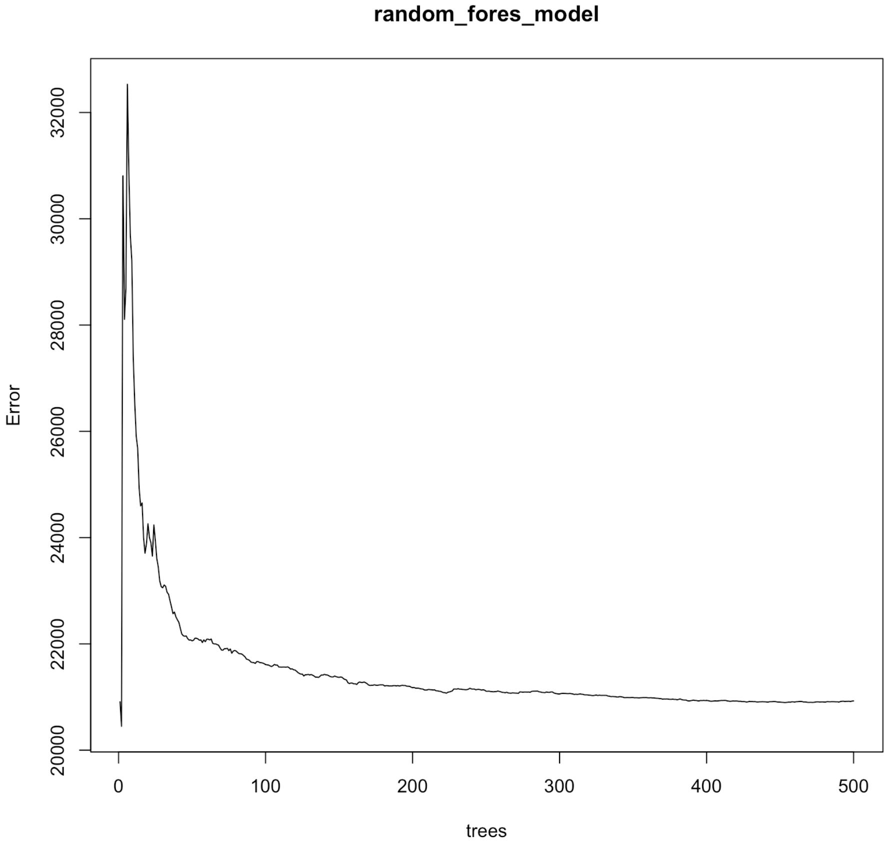
Decision Trees:

We used decision trees as one of the techniques to get close to accurate predictions. Based on the predictor variables, we arrive at a decision at each step of the way to finally reach a decision on the price predictions.



<u>Random Forest:</u>

We used all the 19 principal components that we extracted using PCA as the predictors and fitted a model by building 500 decision trees using random forest.



There was no significant reduction in error rate after 100 trees. We also found the important input variables, which increased the node purity. We used this model to predict the prices in our validation set.

V. Performance Evaluation

Present performance evaluation for all data mining techniques explored in your study, and select the best approach and explain why it is the best. Please ALWAYS divide your data into training set and validation set, and use validation set to evaluate the performance. If using pruned decision trees, please separate your data set into training, validation, and test.

The data mining techniques that we explored for our study were:

- Linear Regression without PCA.
- Linear Regression with PCA.
- Decision Trees.
- Random Forests.

We divided 60% of the data as training set and 40% of the data as validation set to evaluate the performance across all the data mining techniques. We used RMSE to evaluate the performance of each of the models.

From the study, we found that the RMSE of the each of the models are found to be:

- Linear Regression without PCA : 161.27
- Linear Regression with PCA : 92.575
- Decision Trees : 111.374
- Random Forests : 90.8879

From this we can clearly see that the 'Linear Regression with PCA' and 'Random Forests' have considerably lower RMSE than the other models.
The RMSE of the Random Forests was found to be the least and hence this was the best approach for the study.

VI. Discussion and Recommendation

The overall approach to our study was fairly straightforward, although we did spend a lot of time in data pre-processing. We used the most relevant data mining techniques that would give a desirable prediction.
The main advantage of using linear regression techniques was that it is a statistical model which shows a clear correlation between the dependent and the independent variables to the price and principal component analysis was also implemented for dimension reduction.
Decision trees were the best for variable screening as they implicitly perform feature selection. Since we had a lot of variables, decision trees was a necessary data mining technique that was implemented.
As compared to decision trees, there is a significantly lower risk of overfitting if we use random forests. Since multiple trees are used, the error rate will be lesser, Hence, the performance of random forest was more accurate than the decision trees. We would recommend random forest as the best data mining technique.

VII. Summary

The aim of the study was to develop a predictive model to predict the rental prices of Airbnbs in Boston. We cleaned and explored our data to identify the variables that significantly affect the price of the rental.

We generated various visualizations and observed the findings. Based on these observations, we decided to implement various data mining techniques namely, Linear Regression without PCA, Linear regression with PCA, Decision Trees and Random Forests.
After implementing all these models, we found the performance of the Random Forest model was the best.

Appendix: R Code for use case study

R Code:

```
features<-read.csv(“features.csv")
head(features,n=5)
names(features)
features[!complete.cases(features), ]
features<-features[,-14]
features[!complete.cases(features), ]
features<-features[complete.cases(features), ]
features[!complete.cases(features), ]
features$price=as.numeric(gsub("[\\$,]","",features$price))
features[,'price']
head(features,5)
names(features)
features<-features[,-71]
names(features)
features<-features[,-70]
names(features)
str(features)

#convert cat into num var - 0 ot 1
features$Wireless.Internet <- as.integer(features$Wireless.Internet == "True")
features
features$Wheelchair.Accessible <- as.integer(features$Wheelchair.Accessible == "True")
features
features$Washer...Dryer <- as.integer(features$Washer...Dryer == "True")
features
features$host_is_superhost <- as.integer(features$host_is_superhost == "t")
features$host_has_profile_pic <- as.integer(features$host_has_profile_pic == "t")
features$host_total_listings_count <- as.integer(features$host_total_listings_count ==
"True")
features$host_identity_verified <- as.integer(features$host_identity_verified == "True")
features$requires_license <- as.integer(features$requires_license == "True")
features$instant_bookable <- as.integer(features$instant_bookable == "True")
features$require_guest_profile_picture <-
as.integer(features$require_guest_profile_picture == "True")
features$require_guest_phone_verification <-
as.integer(features$require_guest_phone_verification == "True")
features

write.csv(features, file = Featuresnew.csv")

features_2<-read.csv("Featuresnew.csv")
```

```
features_2
features_2<-features_2[,-1]
str(features_2)
names(features_2)
features_2[!complete.cases(features_2), ]
features_2<-features_2[complete.cases(features_2), ]

#Dimension reduction:
  #Create cov matrix:

cov(features_2)
cor(features_2)

#Remove requires_license
features_2<-features_2[,-15]
features_2<-features_2[,-5]
features_2<-features_2[,-14]
features_2<-features_2[,-15]
features_2<-features_2[,-48]
features_2<-features_2[,-15]
#PCA on everything
library(psych)
fa.parallel(features_2[,-18],fa="pc", main="Scree Plot with Parallel Analysis")
# 12 components

pca_features = principal(features_2[,-18], nfactor=19, rotate="none")
pca_features
pca_features$loadings
pca_features$scores

library(car)

#Linear Regression:
train1.index <- sample(row.names(features_2), 0.6*dim(features_2)[1])
valid1.index <- setdiff(row.names(features_2), train1.index)
train1.df <- features_2[train1.index,]
valid1.df <- features_2[valid1.index,]
valid1.df

a_1<-lm(price~.,data=train1.df)
a_1

#a<-lm(training)
pred_1<-predict(a_1,valid1.df[,-18])
pred_1
rmse(pred_1,valid1.df$price)
```

```
head(pred,n=15)

head(valid.df[,18],n=15)

#Using PCs
new_df<-as.data.frame(pca_features$scores)
new_df$price<-features_2[,18]
new_df

train.index <- sample(row.names(new_df), 0.6*dim(new_df)[1])
valid.index <- setdiff(row.names(new_df), train.index)
train.df <- new_df[train.index,]
valid.df <- new_df[valid.index,]
valid.df

a<-lm(price~.,data=train.df)
a
pred1<-predict(a,valid.df[,-20])
pred1
library(Metrics)
rmse(pred1,valid.df$price)

#Regression Tree
library(rpart)
library(rpart.plot)
fit = rpart(price ~ .,data=train.df, method = 'anova')
printcp(fit)
plotcp(fit)
summary(fit)
prp(fit, type = 1, extra = 1, split.font = 1, varlen = -10)
pred<-predict(fit,valid.df)
pred

rmse(pred,valid.df$price)

#Random forest
library(randomForest)
random_fores_model <- randomForest(price~.,
                                   train.df,
                               ntree=500,
                               importance=T)
plot(random_fores_model)

#Variable importance plot
varImpPlot(random_fores_model,
        sort = T,
        main="Variable Importance",
        n.var=5)
var.imp <- data.frame(importance(random_fores_model,
                        type=2))
var.imp$IncNodePurity
var.imp$Variables <- row.names(var.imp)
var.imp[order(var.imp$IncNodePurity ,decreasing = T),]



#Predicting:
predict_rf<-predict(random_fores_model,valid.df)
predict_rf
rmse(predict_rf,valid.df$price)
```