MGT2003 (Slot C2 + TC2)
Fundamentals of Business Analytics Project Assignment

Phase 3

House Price Prediction

By:
Athrwa Deshmukh - 19BCE7381
Riya Deulkar - 19BEC7040

Dataset Dimensions:
Columns: 21
Rows/Samples: 21,613
Dataset Link

Predictive Analysis in R:

Splitting the data into train and test data, the train data will be used to train the machine learning model whereas the test data will be used to make predictions on and cross check with actual price value. The train data should contain a lot of rows since more the data, better will be the trained model, and for testing we require comparatively lesser data, hence splitting as 80%:20%.

```
#splitting data into train-test (80%-20%)
index <- sample(1:nrow(data), size=0.8*nrow(data))
train <- data[index, ]
test <- data[-index, ]
```

Loading the required libraries

```
library(tidyverse)
library(caret)
library(Metrics)
library(randomForest)
```

Polynomial Regression:

#Building the model with polynomial regression

model <- lm(price ~ poly(bedrooms+bathrooms+sqft_living+floors+grade, 5, raw = TRUE),data = train)

 # Making predictions

predictions <- model %>% predict(test)

modelPerfomance = data.frame(

RMSE = RMSE(predictions, test$price),

R2 = R2(predictions, test$price)

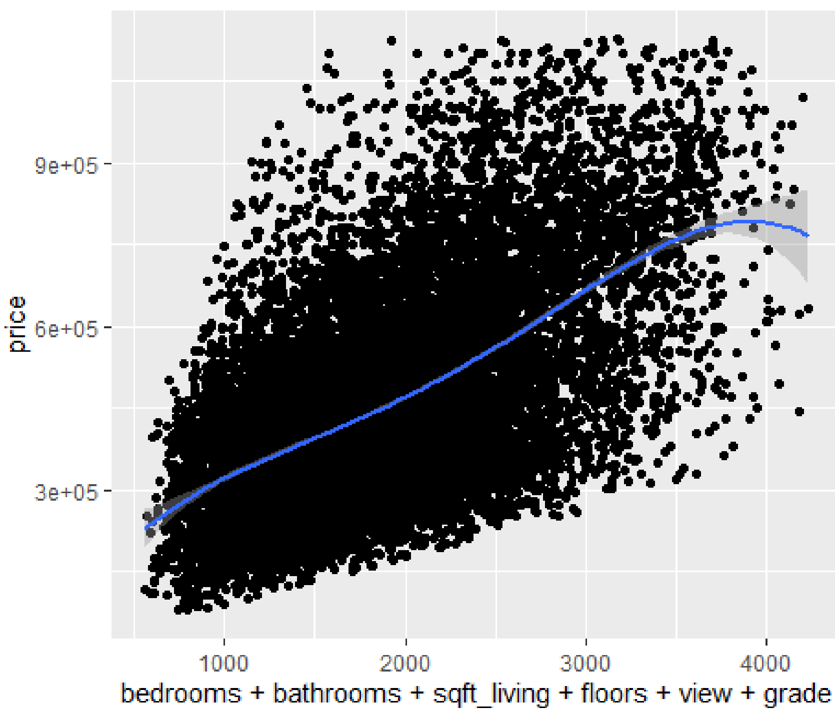
)

print(modelPerfomance)

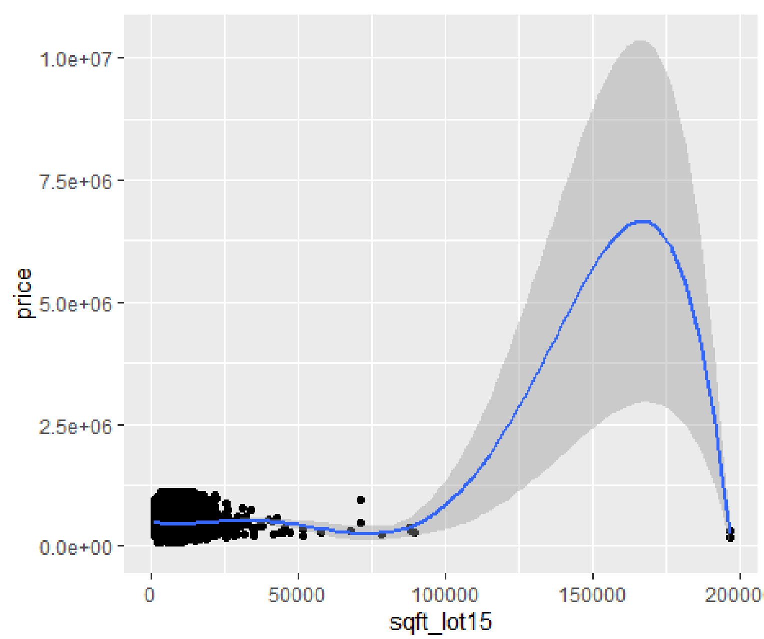
```
> print(modelPerfomance)
      RMSE        R2
1 163358.9 0.3438666
```

Checking the model w.r.t sum of various features

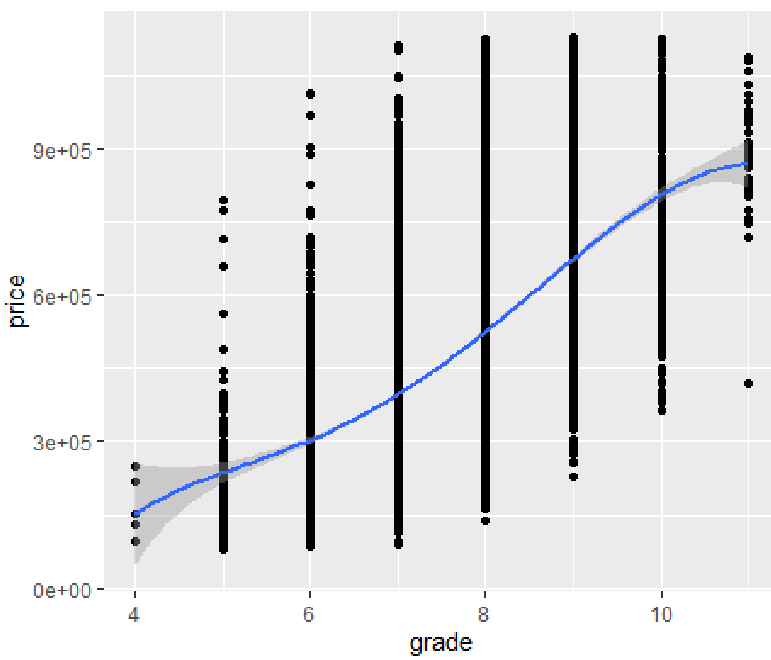
ggplot(train, aes(bedrooms+bathrooms+sqft_living+floors+view+grade, price))+geom_point()+stat_smooth(method = lm, formula = y ~ poly(x, 5, raw = TRUE))



The graph contains a lot of data points and there isn't much clarity using the sum of these features, trying to check with individual features ggplot(train, aes(sqft_lot15, price))+geom_point()+stat_smooth(method = lm, formula = y ~ poly(x, 5, raw = TRUE))



ggplot(train, aes(grade, price))+geom_point()+stat_smooth(method = lm, formula = y ~ poly(x, 5, raw = TRUE))



The RMSE for this model is 163358.9 and R2 value is 0.343 i.e. 34% which means that the model explains some of the variation in the response variable around its mean. Like we can see in the graph for sqft_lot15 is much more understandable as compared to grade and the first graph.

Linear Regression and Multiple Linear Regression:

Using 4 different equations and calculating it's RMSE(root mean square error) to see which suits best

```
e1=lm(price~grade,data=train)
p1 = predict(e1, test)
rms[1]<-rmse(test$price,p1)

e2=lm(price~sqft_living+grade+floors+bathrooms+bedrooms+view+sqft_basement+sqft_lot15+zipcode,data=train)
p2 = predict(e2, test)
rms[2]<-rmse(test$price,p2)

e3=lm(price~.,data=train)
p3 = predict(e3, test)
rms[3]<-rmse(test$price,p3)

e4=lm(price~sqft_lot15,data=train)
p4 = predict(e4, test)
rms[4]<-rmse(test$price,p4)

rms
```

```
> rms
[1] 163668.3 147496.3 145215.5 201489.0
```

We can note that model 3 / equation 3 provides the best prediction as it has the least RMSE i.e 145215.5, which also tells us that using multiple linear regression for this dataset, considering all features help in decreasing the RMSE, hence providing better results.

Therefore trying to build a multiple linear regression model with all features from original dataset after removal of outliers and missing values.

```
e5=lm(price~.,data=train)
p5 = predict(e5, test)
rms[5]<-rmse(test$price,p5)
```

rms

```
> rms
[1] 163668.3 147496.3 145215.5 201489.0 132370.9
```

We can see that this model has the least RMSE and is therefore the best amongst the 5 Linear Regression models we trained.

Random Forest Regressor:

Since Random Forest does not require dimensionality reduction as the ID3 algorithm checks the Information Gain for each feature and based on that builds a decision tree, hence applying the Random Forest Regressor on data after removing outliers

#applying random forest regressor before dimensionality reduction

rf.fit <- randomForest(price ~ ., data=data, ntree=1000,keep.forest=FALSE, importance=TRUE)
print(rf.fit)

```
> print(rf.fit)

Call:
 randomForest(formula = price ~ ., data = data, ntree = 1000,      keep.forest = FALSE, importance = TRU
E)
               Type of random forest: regression
                     Number of trees: 1000
No. of variables tried at each split: 5

          Mean of squared residuals: 9887572119
                    % Var explained: 75.33
```

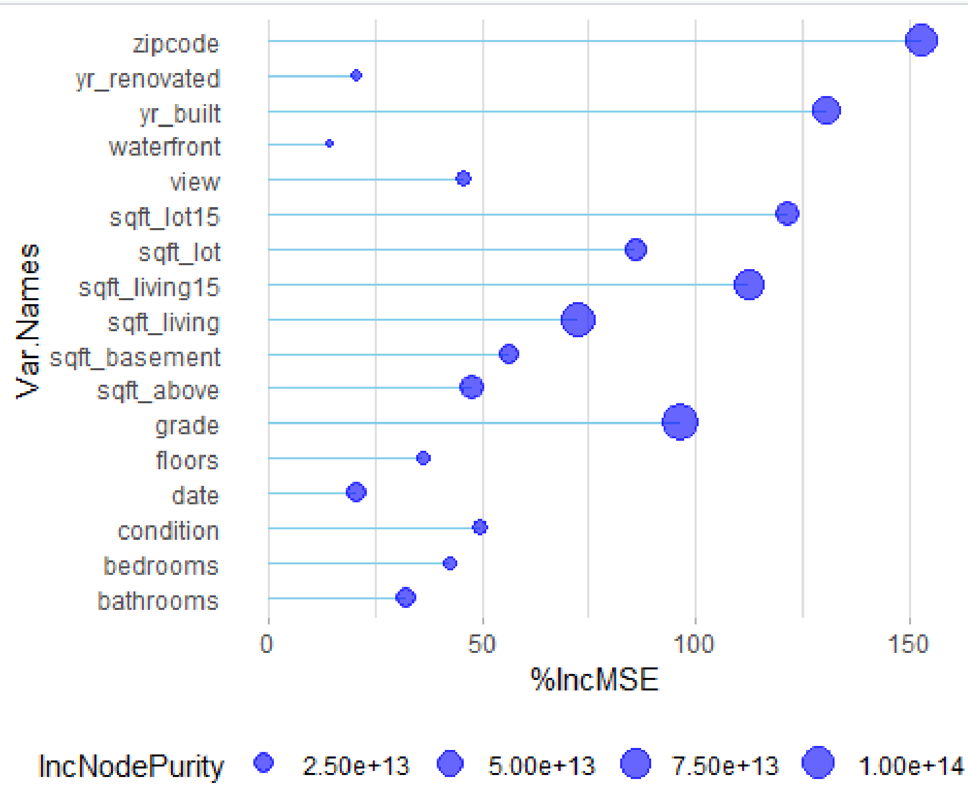
We can see that RME is mentioned, therefore we can get RMSE by taking square root i.e. 99436.27, which is the least so far from all of the algorithms we have used including polynomial regression, linear regression and multiple linear regression. Additionally, we can see that 75% of the variance is explained using this algorithm.

 # Get variable importance from the model fit

ImpData <- as.data.frame(importance(rf.fit))
ImpData$Var.Names <- row.names(ImpData)

ggplot(ImpData, aes(x=Var.Names, y=`%IncMSE`)) +
  geom_segment(aes(x=Var.Names, xend=Var.Names, y=0, yend=`%IncMSE`), color="skyblue") +
  geom_point(aes(size = IncNodePurity), color="blue", alpha=0.6) +
  theme_light() +
  coord_flip() +
  theme(
    legend.position="bottom",

```
  panel.grid.major.y = element_blank(),
  panel.border = element_blank(),
  axis.ticks.y = element_blank()
)
```



__Terminology:__

1) *__Percent Increase MSE (%IncMSE) -__ This shows how much our model accuracy decreases or Mean Decrease Accuracy if we leave out that variable.*
2) *__IncNodePurity -__ This is a measure of variable importance based on the Gini impurity index used for calculating the splits in trees.*

__Inference:__

We can see that ZipCode has the highest %IncMSE followed by yr_built and sqft_lot15.
Additionally grade and sqft_living have the highest IncNodePurity.
Similarly we can see the importance of each feature of our dataset.

Applying Random Forest Regressor after pre-processing and dimensionality reduction.
rf.fit1 <- randomForest(price ~ ., data=train, ntree=1000,keep.forest=FALSE, importance=TRUE)
print(rf.fit1)

```
> rf.fit1 <- randomForest(price ~ ., data=train, ntree=1000,keep.forest=FALSE, importance=TRUE)
> print(rf.fit1)

Call:
 randomForest(formula = price ~ ., data = train, ntree = 1000,      keep.forest = FALSE, importance = TRU
E)
               Type of random forest: regression
                     Number of trees: 1000
No. of variables tried at each split: 4

          Mean of squared residuals: 10291645076
                    % Var explained: 74.44
>
```

It is evident that the MSE has increased w.r.t the previous Random Forest Regressor Model. Hence we can conclude that Random Forest Regressor used on the dataset without reducing dimensions works the best for predicting the housing prices.

Implications on the initial Business Questions

Q1) What are the standard KPIs which affect the house pricing?

Ans) Based on the plot in random forest regressor we can see that zipcode, year the house was built, average of 15 nearest houses lot size, living area and grade are the Key Point Indicators in determining the house price

Q2) What is the average price per sq. feet

Ans) Based on the code and analysis done so far, we can not answer this question, however to answer this question we can simply write a one line R code as follows:
sum(data$price)/sum(data$sqft_lot)

```
> sum(data$price)/sum(data$sqft_lot)
[1] 63.73511
```

Therefore the average price per square foot is 63.73511 dollars.

Q3) In which months of the year are the best offers cheaper? and more expensive?

Ans) Based on the analysis done so far, we haven't answered this question, however we can answer it with the following R code:
df1 = as.data.frame(data)

df1$Month <- months(data$date)
aggregate(price~Month, data=df1, FUN=function(df1) c(mean=mean(df1), count=length(df1)))

```
> df1 = as.data.frame(data)
> df1$Month <- months(data$date)
> aggregate(price~Month, data=df1, FUN=function(df1) c(mean=mean(df1), count=length(df1)))
       Month price.mean price.count
1      April   485547.9      1839.0
2     August   462872.0      1601.0
3   December   444720.8      1200.0
4   February   440330.6      1054.0
5    January   438173.6       799.0
6       July   467833.0      1818.0
7       June   476760.0      1750.0
8      March   466534.4      1551.0
9        May   468486.7      1944.0
10  November   449381.4      1164.0
11   October   456721.3      1543.0
12 September   461941.5      1453.0
```

Therefore, we can see that the average price of the houses is cheapest in January whereas Most expensive in April, however the difference is negligible and not a major one.

Q4) What year in the dataset had more good deal houses?

Ans) We can find this by using the code:
aggregate(price~yr_built, data=data, FUN=function(data) c(mean=mean(data), count=length(data)))

```
> aggregate(price~yr_built, data=data, FUN=function(data) c(mean=mean(data), count=length(data)))
    yr_built price.mean price.count
1       1900   537328.2        71.0
2       1901   524890.1        26.0
3       1902   566573.9        23.0
4       1903   492024.2        43.0
5       1904   527675.6        43.0
6       1905   594182.4        54.0
7       1906   568504.5        76.0
8       1907   576131.3        54.0
9       1908   495955.5        70.0
10      1909   563809.7        83.0
11      1910   563412.0       115.0
12      1911   563420.7        58.0
13      1912   570962.4        70.0
14      1913   507613.0        44.0
15      1914   529028.6        45.0
16      1915   546593.6        53.0
17      1916   511047.7        69.0
18      1917   493481.2        48.0
19      1918   428469.4        99.0
20      1919   525921.2        78.0
21      1920   497915.2        84.0
22      1921   552788.5        66.0
23      1922   531229.3        80.0
24      1923   518818.8        70.0
25      1924   535922.2       125.0
26      1925   543503.5       144.0
27      1926   570781.4       161.0
28      1927   578759.4        99.0
```

We can infer from the data that the year 1943 had more good deal houses with mean price being 331665.5 dollars.

Q5) Which ZipCode is the costliest to live?

Ans) Based on the Data Visualization on Power BI, we can say that the cheapest ZipCode to live in is 98002 with average price being 234284.04 dollars.

Q6) Which ZipCode is the cheapest to live?

Ans) Based on the Data Visualization on Power BI, we can say that the cheapest ZipCode to live in is 98039 with average price being 2160606.60 dollars.

Q7) Does the year the house was built have any effect on the price?

Ans) Yes, the year built has a major impact on the price. We can see so from the plot made for random forest regressor where yr_built has approx 130 %IncMSE and 7.5*10^13 IncNodePurity. However when considering correlations, the yr_built has a mere correlation of 0.002 on the price.

Q8) Find the average house price for renovated houses compared to not renovated ones.

Ans) Based on the analysis done so far we cannot solve this question, however we can use the following R code to get the answer:

sum(subset(data, yr_renovated == "0")$price)/nrow(subset(data, yr_renovated == "0"))

```
> sum(subset(data, yr_renovated == "0")$price)/nrow(subset(data, yr_renovated == "0"))
[1] 458939.6
```

sum(subset(data, yr_renovated != "0")$price)/nrow(subset(data, yr_renovated != "0"))

```
> sum(subset(data, yr_renovated != "0")$price)/nrow(subset(data, yr_renovated != "0"))
[1] 570622.4
```

Therefore, the average house price for non renovated houses is 458939.6 dollars, whereas for renovated houses is 570622.4 dollars.

Q9) How costly are houses with a waterfront as compared to ones without one?

Ans) We can get the above answer by using the following R code:
sum(subset(data, waterfront== "0")$price)/nrow(subset(data, waterfront == "0"))

```
> sum(subset(data, waterfront== "0")$price)/nrow(subset(data, waterfront == "0"))
[1] 462399.9
```

sum(subset(data, waterfront== "1")$price)/nrow(subset(data, waterfront == "1"))

```
> sum(subset(data, waterfront== "1")$price)/nrow(subset(data, waterfront == "1"))
[1] 725727.2
```

We can see that the average house price for houses without waterfront is 462399.9 dollars whereas for houses with waterfront is 725727.2 dolars.
To check the price per square feet we can use the following code:
sum(subset(data, waterfront== "0")$price)/sum(subset(data, waterfront== "0")$sqft_lot)
sum(subset(data, waterfront== "1")$price)/sum(subset(data, waterfront== "1")$sqft_lot)

```
> sum(subset(data, waterfront== "0")$price)/sum(subset(data, waterfront== "0")$sqft_lot)
[1] 63.93674
> sum(subset(data, waterfront== "1")$price)/sum(subset(data, waterfront== "1")$sqft_lot)
[1] 63.63525
```

Therefore we can see that the price per square feet is almost similar for houses with and without waterfront i.e approximately 64 dollars.

Q10) What are the average pricings of houses based on the number of bedrooms?

Ans) We can get the above by using the following R code:
aggregate(price~bedrooms, data=data, FUN=function(data) c(mean=mean(data), count=length(data)))

```
> aggregate(price~bedrooms, data=data, FUN=function(data) c(mean=mean(data), count=length(data)))
  bedrooms price.mean price.count
1        2   390848.5      2553.0
2        3   431023.6      8643.0
3        4   527393.8      5418.0
4        5   562112.0      1102.0
```

Q11) Estimate the cost on the basis of the condition of the house.

Ans) We can get the above by using the following R code:
aggregate(price~condition, data=data, FUN=function(data) c(mean=mean(data), count=length(data)))

```
> aggregate(price~condition, data=data, FUN=function(data) c(mean=mean(data), count=length(data)))
  condition price.mean price.count
1         1   290673.5        17.0
2         2   305588.7       120.0
3         3   461569.2     11524.0
4         4   454863.6      4645.0
5         5   515221.8      1410.0
```

Similarly we can check for grade too using:
aggregate(price~grade, data=data, FUN=function(data) c(mean=mean(data), count=length(data)))

```
> aggregate(price~grade, data=data, FUN=function(data) c(mean=mean(data), count=length(data)))
  grade price.mean price.count
1     4   206300.0        10.0
2     5   241490.7       167.0
3     6   299574.3      1806.0
4     7   398455.1      8228.0
5     8   520264.3      5213.0
6     9   686214.6      1808.0
7    10   787084.3       441.0
8    11   888607.2        43.0
```

Q12) What factors can be ignored/do not affect the pricing of houses?

Ans) Factors like waterfront, year renovated, view, bathrooms, date and condition can be ignored for this dataset. Waterfront has a correlation of 0.05 with the price additionally, it has the least %IncMSE and IncNodePurity. Year renovated has comparable %IncMSE and IncNodePurity and does not provide much information gain. Similarly View, Bathrooms, Date and condition have low %IncMSE and IncNodePurity and hence can be ignored. All these factors have minimal effect on the pricing of the house.