Data Analytics Source Code

COM681(22279) Data Analysis on House Prices in England for 2021

Student: Verejan Vasile

Student ID: B00787947

```
library(tidyverse)
library(dplyr)
library(ggplot2)
library(cluster)
library(hrbrthemes)
library(NbClust)
library(factoextra)
library(caTools)
library(rpart);

#Importing data
house.data <- read.csv('pp-2021.csv',header = TRUE)

#Checking Data
summary(house.data)
str(house.data)

# 1. Cleaning Data
#Checking for missing values
any(is.na(house.data))
sum(is.na(house.data))

# 2. Checking for duplicate rows and removing them
house.data <- unique(house.data)

#3. Checking house prices for outliers
```

```
house.data %>%
ggplot(aes(x = Price))+
geom_boxplot()+
xlab('house price')
summary(house.data)
```

#4. There are some prices like 1, and couple millions which are outliers will remove them by setting price condition.

```
house.data <- subset(house.data,Price>=170000 & Price<=415000)
```

#Ploting again to look for outliers

```
house.data %>%
ggplot(aes(x = Price))+
geom_boxplot(fill = 'blue')+
xlab('house price')
summary(house.data)
```

#5. Deleting the columns that are missing data as well some columns such as postcode will,be remove because its outside the scope of the project.

```
house.data[,c('Transaction_ID','PAON','SAON','Locality','PPD',"Record_Status",'Postcode',"Street", 'District' )] <- list(NULL)
glimpse(house.data)
summary(house.data)
```

#6. Since all houses are sold in 2021, removed the full date and left only the month.

```
house.data$Transfer_Date<- format(as.Date(house.data$Transfer_Date, format="%d/%m/%Y"),"%m")
str(house.data)
```

#7.Converting the column names in Capital letters and more meaningful names.

```
house.data <- rename(house.data,
PRICE = Price,
TRANSFER_DATE=Transfer_Date,
TYPE = Property_Type,
OLD_NEW = Old.New,
DURATION = Duration,
TOWN = Town.City,
COUNTY = County
)
glimpse(house.data)

#8. Some columns such as type, old_new, duration is repeating the same value, so we have to transform them into factors for better analysis
house.data1 <- house.data %>%
mutate(
TYPE = as.factor(TYPE),
OLD_NEW = as.factor(OLD_NEW),
DURATION = as.factor(DURATION)

)%>%
mutate_if(is.character,as.factor)%>%
dplyr::select_all()

summary(house.data1)

#The months are coming with 0 in front which is not ok, so it will be converted into numeric.
house.data1$TRANSFER_DATE <- as.numeric(house.data1$TRANSFER_DATE)
summary(house.data1)

#9. Before selecting the particular area lets see how our plot for price/duration looks like, are the leasehold property cheaper?
house.data1 %>%
ggplot(aes(x = DURATION, y = PRICE, color = 'region'))+

geom_boxplot(fill = 'purple')+

xlab('Property Type')+

ylab('House Price')


#The prices for new properties are higher than average of old properties.

house.data1 %>%

ggplot(aes(x = OLD_NEW, y = PRICE, color = 'region'))+

geom_boxplot(fill = 'pink')+

xlab('Property Type')+

ylab('House Price')




#There are too many records in this dataset so before continuing we have to limit
our Analysis to Swindon Area which is the main scope.

swindon_area <- house.data1 %>%

filter(TOWN == 'SWINDON')

plot(swindon_area$PRICE)

glimpse(swindon_area)


#10, Scope related Q/A. .


#Average Price per house type.

swindon_area %>%

ggplot(aes(x = TYPE, y = PRICE, color = 'region'))+

geom_boxplot(fill = 'yellow')+

xlab('Property Type')+

ylab('House Price')


#Check if the price for leasehold is lower than average price for freehold.
```

```
swindon_area %>%
ggplot(aes(x = DURATION, y = PRICE, color = 'region'))+
geom_boxplot(fill = 'yellow')+
xlab('Duration Free/Lease')+
ylab('House Price')

#IS there houses for same price as old houses
swindon_area %>%
ggplot(aes(x = OLD_NEW, y = PRICE, color = 'region'))+
geom_boxplot(fill = 'yellow')+
xlab('Old/New')+
ylab('House Price')


#Which month you could pay a lower price for buying a property
swindon_area %>%
ggplot(aes( x = as.factor(TRANSFER_DATE), y = PRICE, color = 'region'))+
geom_boxplot(fill = 'yellow')+
xlab('Transfer Month')+
ylab('House Price')

#Analize price by months and types.
swindon_area %>%
ggplot(aes( x = as.factor(TRANSFER_DATE), y = PRICE, color = 'region'))+
geom_boxplot(fill = 'green')+
xlab('Transfer Month')+
ylab('House Price')+
facet_grid(~TYPE)

#Is it true that new flats are the same price as old ?
swindon_area %>%
ggplot(aes( x = TYPE, y = PRICE, color = 'region'))+
geom_boxplot(fill = 'yellow')+
xlab('Property Type')+
ylab('House Price')+
facet_grid(~OLD_NEW)

#Are properties tending to be more on leasehold base or freehold.
swindon_area %>%
ggplot(aes( x = OLD_NEW, y = PRICE, color = 'region'))+
geom_boxplot(fill = 'yellow')+
xlab('Old/New')+
ylab('House Price')+
facet_grid(~DURATION)

#Are people interested in buying the detached and semi-dethached more?
swindon_area %>%
ggplot(aes( x = OLD_NEW, y = PRICE, color = 'region'))+
geom_boxplot(fill = 'lightblue')+
xlab('Old/New')+
ylab('House PRICE')+
facet_grid(~TYPE)




#11. Clustering
#Using daisy method for dissimilarity matrix
dissimilar_matrix <- daisy(swindon_area, metric="gower")

#Visualize matrix
gradient.color <- list(low = "white",  high = "red")
```

```
fviz_dist(dissimilar_matrix,
          gradient = gradient.color,order=T)


#Applying silhouette method to determine the number of clusters.
clusters_no <- NbClust(diss=dissimilar_matrix,distance=NULL,
                       min.nc = 3, max.nc = 10,
                       method = "median",
                       index="silhouette")
#checking number of clusters, index is ranging from 1 : -1 very well clustered/bad clustered.
clusters_no$Best.nc


#Using three clusters
dom_pam <- pam(dissimilar_matrix,3)


#2D visualization of the clusters.
dom_mds <- as.data.frame(cmdscale(dissimilar_matrix,2))
dom_mds$domains_cluster <- as.factor(dom_pam$clustering)
ggplot(dom_mds,aes(x=V1,y=V2,color=domains_cluster)) +
  geom_point() + theme_ipsum() +
  labs(title="PLOT for Domain",
       subtitle="Each color represents a cluster") +
  scale_color_brewer(palette="Set1")




#Prediction analysis


#Split data in 70% and 30%
splited_data <- sample.split(swindon_area,SplitRatio = 0.3)
```

```
#sub-setting into Train data
swindon_train =subset(swindon_area,splited_data==TRUE)


#sub-setting into Test data
swindon_test =subset(swindon_area,splited_data==FALSE)


#Model 1: Creating linear regression model, using training dataset to train the model
linear_model <- lm(PRICE ~  TRANSFER_DATE + OLD_NEW + DURATION, data =
swindon_train);


#Model 2: Creating decision tree model.


tree_model  <- rpart(PRICE ~  TRANSFER_DATE + OLD_NEW + DURATION, data =
swindon_train);



#Both models will be used to make predictions.
linear_prediction <- predict(linear_model, swindon_test)
linear_prediction <- data.frame(Price_Pred = linear_prediction, PRICE =
swindon_test$PRICE,
                TRANSFER_DATE = swindon_test$TRANSFER_DATE, OLD_NEW =
swindon_test$OLD_NEW,
                DURATION =swindon_test$DURATION)



tree_prediction  <- predict(tree_model,  swindon_test)
tree_prediction <- data.frame(Price_Pred = tree_prediction, PRICE =
swindon_test$PRICE,
                TRANSFER_DATE = swindon_test$TRANSFER_DATE, OLD_NEW =
swindon_test$OLD_NEW,
                DURATION =swindon_test$DURATION)
```

```
#To check if everything worked out we have to look on the first observations of both models

head(linear_prediction);

head(tree_prediction);


par(mfrow = c(1, 2));

plot(linear_prediction$Price_Pred - linear_prediction$PRICE, main = "Predicted Price - Actual Price (Linear)")

plot(tree_prediction$Price_Pred  - tree_prediction$PRICE,  main = "Predicted Price - Actual Price (Tree Model)")


par(mfrow = c(2, 2));

hist(linear_prediction$Price_Pred - linear_prediction$PRICE, main = "Predicted Price - Actual Price (Linear)")

hist(tree_prediction$Price_Pred  - tree_prediction$PRICE,  main = "Predicted Price - Actual Price (Tree Model)")

hist(linear_prediction$Price_Pred, main = "Linear Regression Model")

hist(tree_prediction$Price_Pred, main = "Decision Tree Model")
```