ASHUTOSH AGARWAL

Predicting House Prices in Bengaluru

Dataset: Predicting-House-Prices-In-Bengaluru

The train and test data will consist of various features that describe that property in Bengaluru. This is an actual data set that is curated over months of primary & secondary research by our team. Each row contains fixed size object of features. There are 9 features and each feature can be accessed by its name.

Features

1. Area_type - describes the area
2. Availability - when it can be possessed or when it is ready(categorical and time-series)
3. Location - where it is located in Bengaluru
4. Price - Value of the property in lakhs(INR)
5. Size - in BHK or Bedroom (1-10 or more)
6. Society - to which society it belongs
7. Total_sqft - size of the property in sq.ft
8. Bath - No. of bathrooms
9. Balcony - No. of the balcony

Code to build a Model:

#Loading training data to build model

```
> prices_train = read.csv(file.choose())
```

#checking the dimension of train data

```
> dim(prices_train)
[1] 13320     9
```

#looking top 5 rows

```
> head(prices_train,5)
            area_type  availability                location       size society total_sqft bath balcony  price
1 Super built-up  Area       19-Dec Electronic City Phase II      2 BHK Coomee        1056    2       1  39.07
2           Plot  Area Ready To Move        Chikka Tirupathi 4 Bedroom Theanmp       2600    5       3 120.00
3       Built-up  Area Ready To Move             Uttarahalli      3 BHK              1440    2       3  62.00
4 Super built-up  Area Ready To Move      Lingadheeranahalli      3 BHK Soiewre       1521    3       1  95.00
5 Super built-up  Area Ready To Move                Kothanur      2 BHK              1200    2       1  51.00
```

#Data Cleaning

```
> summary(prices_train)
                area_type          availability                   location             size              society     
 Built-up  Area      :2418   Ready To Move:10581   Whitefield        :  540   2 BHK    :5199              :5502  
 Carpet  Area        :  87   18-Dec       :  307   Sarjapur  Road    :  399   3 BHK    :4310   GrrvaGr:   80  
 Plot  Area          :2025   18-May       :  295   Electronic City:  302   4 Bedroom: 826   PrarePa:   76  
 Super built-up  Area:8790   18-Apr       :  271   Kanakpura Road    :  273   4 BHK    : 591   Prtates:   59  
                             18-Aug       :  200   Thanisandra       :  234   3 Bedroom: 547   Sryalan:   59  
                             19-Dec       :  185   Yelahanka         :  213   1 BHK    : 538   GMown E:   56  
                             (Other)      : 1481   (Other)           :11359   (Other)  :1309   (Other):7488  
   total_sqft         bath           balcony          price       
 1200   :  843   Min.   : 1.000   Min.   :0.000   Min.   :   8.0  
 1100   :  221   1st Qu.: 2.000   1st Qu.:1.000   1st Qu.:  50.0  
 1500   :  205   Median : 2.000   Median :2.000   Median :  72.0  
 2400   :  196   Mean   : 2.693   Mean   :1.584   Mean   : 112.6  
 600    :  180   3rd Qu.: 3.000   3rd Qu.:2.000   3rd Qu.: 120.0  
 1000   :  172   Max.   :40.000   Max.   :3.000   Max.   :3600.0  
 (Other):11503   NA's   :73       NA's   :609                     
```

#Predictors/features matrix

```
> x = prices_train[,c(1:8)]
```

#Response Vector

```
> y = prices_train[,9]

> unique(x$area_type)
[1] Super built-up Area  Plot Area  Built-up Area  Carpet Area
Levels: Built-up  Area Carpet  Area Plot  Area Super built-up  Area
```

#There are four different area types and can be converted to numerical representation for calculation purpose

#Built-up Area = 1

#Carpet Area = 2

#Plot Area = 3

#Super built-up Area = 4

```
> x$area_type = as.numeric(x$area_type)

> unique(x$size)
```

#As size shows the number of bedrooms – as BHK and Bedroom both shows the number of rooms, so created a column as # of bedroom

```
> x$bedroom = 0
> for (i in c(1:13320)){
+ x$bedroom[i] = as.numeric(strsplit(as.character(x$size)," ")[[i]][1])
+ }
```

#removing size column as it is no longer needed as it is replaced by bedroom column

```
> x = x[,-4]

#As number of NAs are less so it can be replaced by mean value of bedroom
> mu_bedroom = mean(x$bedroom, na.rm = TRUE)
> x$bedroom[is.na(x$bedroom)] = mu_bedroom

> unique(x$bath)
 [1]  2  5  3  4  6  1  9 NA  8  7 11 10 14 27 12 16 40 15 13 18

#As number of NAs are less so it can be replaced by mean value of bath
> mu_bath = mean(x$bath, na.rm = TRUE)
> x$bath[is.na(x$bath)] = mu_bath

> unique(x$bath)
 [1]  2.00000  5.00000  3.00000  4.00000  6.00000  1.00000  9.00000  2.69261
8.00000  7.00000 11.00000 10.00000
[13] 14.00000 27.00000 12.00000 16.00000 40.00000 15.00000 13.00000 18.00000


> unique(x$balcony)
[1]  1  3 NA  2  0

#As number of NAs are less so it can be replaced by mean value of balcony
> mu_balcony = mean(x$balcony, na.rm = TRUE)
> x$balcony[is.na(x$balcony)] = mu_balcony

> unique(x$balcony)
[1] 1.000000 3.000000 1.584376 2.000000 0.000000
```

```
> unique(x$total_sqft)
```

#As some of the rows in the total_sqft shows the range, so taking the mean of them for calculation purpose.

```
> x$total_sqfts = 0
> for (i in c(1:13320)){
+   x$total_sqfts[i] = mean(as.numeric(strsplit(as.character(x$total_sqft),"-
")[[i]]))
+ }
```

#removing total_sqft column as it is no longer needed as it is replaced by total_sqfts column

```
> x = x[,-5]
```

#As number of NAs are less so it can be replaced by mean value of total_sqfts

```
> mu_total_sqfts = mean(x$total_sqfts, na.rm = TRUE)
> x$total_sqfts[is.na(x$total_sqfts)] = mu_total_sqfts
```

#Feature Scaling

#Calculating mean and range for each feature to normalize them

```
> mu_area_type = mean(x$area_type)
> ran_area_type = max(x$area_type)-min(x$area_type)
> mu_bath = mean(x$bath)
> ran_bath = max(x$bath)-min(x$bath)
> mu_balcony = mean(x$balcony)
> ran_balcony = max(x$balcony)-min(x$balcony)
> mu_bedroom = mean(x$bedroom)
> ran_bedroom = max(x$bedroom)-min(x$bedroom)
> mu_total_sqfts = mean(x$total_sqfts)
> ran_total_sqfts = max(x$total_sqfts)-min(x$total_sqfts)
```

#Normalizing the features

```
> area_type_nor = (x$area_type - mu_area_type)/ran_area_type
> bath_nor = (x$bath - mu_bath)/ran_bath
> balcony_nor = (x$balcony - mu_balcony)/ran_balcony
> bedroom_nor = (x$bedroom - mu_bedroom)/ran_bedroom
> total_sqfts_nor = (x$total_sqfts - mu_total_sqfts)/ran_total_sqfts
> x_zero = 1
> x_norm = data.frame(x_zero,area_type_nor,bath_nor,balcony_nor,bedroom_nor,t
otal_sqfts_nor)
```

#finding feature coefficients for linear regression i.e. theta

#Initializing theta with zero values

```
> theta = matrix(0,9,1)
```

#Converting y into matrix

```
> y = as.matrix(y)
> x_norm = as.matrix(x_norm)
```

#USing gradient Descent to minize the cost function J.

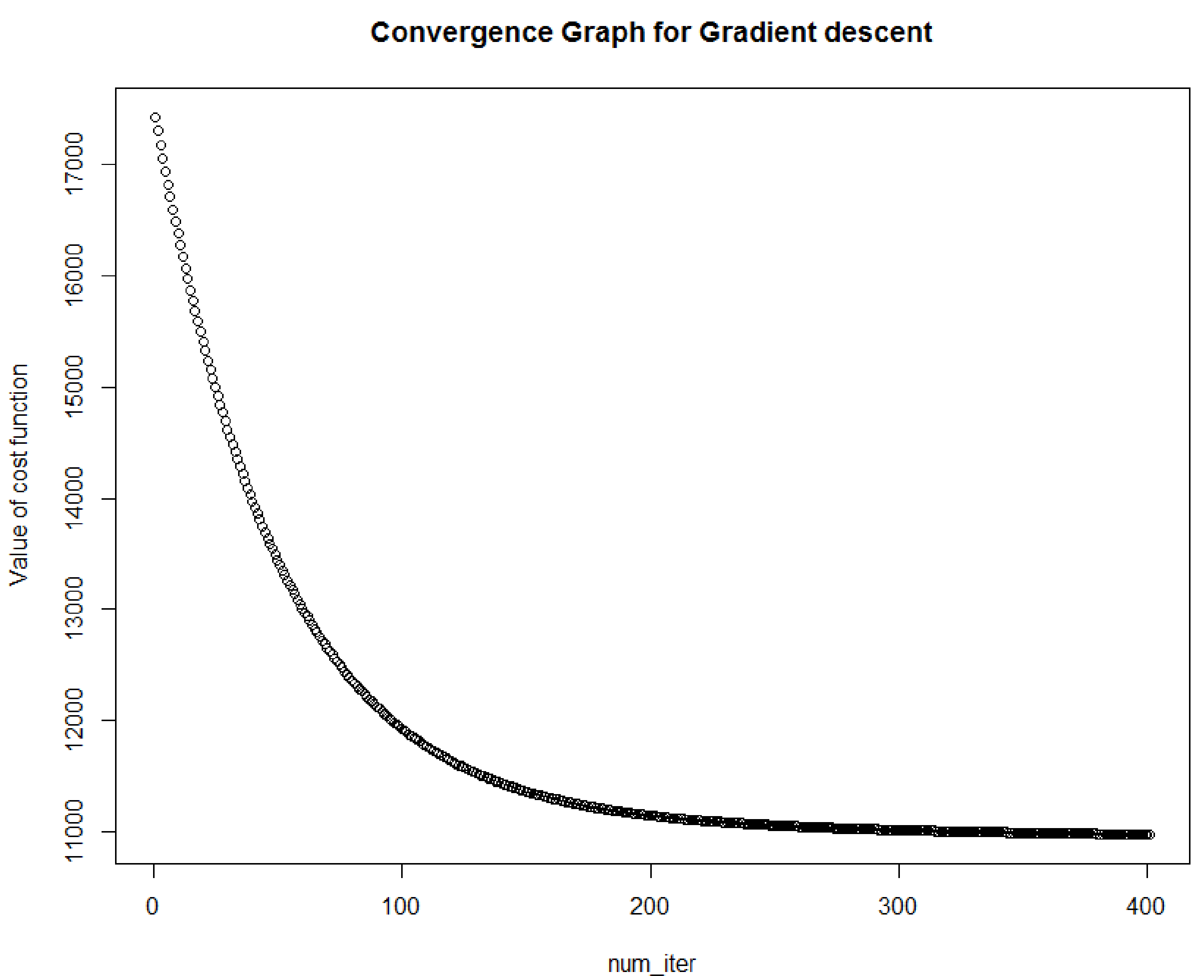
```
> alpha = 0.01
> num_iter = 400
> j_vals = matrix(0,1,401)
> m = length(y)
```

```
> j_vals[1] = (sum((x_norm%*%theta - y)^2))/(2*m)
```

```
> for (i in c(2:401)){
+   delta = (colSums((as.vector(x_norm%*%theta - y))*x_norm))/m
+   delta_transpose = as.matrix(delta)
+   theta = theta - (alpha*delta_transpose)
+
+   j_vals[i] = (sum((x_norm%*%theta - y)^2))/(2*m)
+ }
```

```
> theta
                      [,1]
x_zero          110.545011
area_type_nor   -10.084483
bath_nor          9.187752
balcony_nor      14.496869
bedroom_nor       7.200320
total_sqfts_nor   8.048063
```

```
> j_vals = as.vector(j_vals)
> plot(j_vals,xlab = 'num_iter',ylab = 'Value of cost function',
+ main = 'Convergence Graph for Gradient descent')
```



Predicting the prices for test data:

#Loading test data to predict the prices

```
> prices_test = read.csv(file.choose())

#checking the dimension of test data
> dim(prices_test)
[1] 1480    9
```

#looking top 5 rows

```
> head(prices_test,5)
      area_type  availability        location      size society total_sqft bath balcony price
1 Super built-up  Area Ready To Move      Brookefield     2 BHK Roeekbl       1225    2       2    NA
2           Plot  Area Ready To Move    Akshaya Nagar 9 Bedroom              2400    9       2    NA
3           Plot  Area       18-Apr      Hennur Road 4 Bedroom Saandtt       1650    5       2    NA
4 Super built-up  Area Ready To Move Kodichikkanahalli     3 BHK Winerri       1322    3       1    NA
5 Super built-up  Area Ready To Move       Konanakunte     2 BHK AmageSa       1161    2       1    NA
```

#Data Cleaning

```
> summary(prices_test)
```

#Predictors/features matrix

```
> x_test = prices_test[,c(1:8)]

> x_test$area_type = as.numeric(x_test$area_type)

> x_test$bedroom = 0
> for (i in c(1:1480)){
+ x_test$bedroom[i] = as.numeric(strsplit(as.character(x_test$size)," ")[[i]]
[1])
+ }
```

#removing size column as it is no longer needed as it is replaced by bedroom column

```
> x_test = x_test[,-4]

> mu_bedroom_test = mean(x_test$bedroom, na.rm = TRUE)
> x_test$bedroom[is.na(x_test$bedroom)] = mu_bedroom_test

> mu_bath_test = mean(x_test$bath, na.rm = TRUE)
> x_test$bath[is.na(x_test$bath)] = mu_bath_test
> mu_balcony_test = mean(x_test$balcony, na.rm = TRUE)
> x_test$balcony[is.na(x_test$balcony)] = mu_balcony_test
> x_test$total_sqfts = 0
> for (i in c(1:1480)){
+ x_test$total_sqfts[i] = mean(as.numeric(strsplit(as.character(x_test$total_
sqft),"-")[[i]]))
+ }

> x_test = x_test[,-5]

> mu_total_sqfts_test = mean(x_test$total_sqfts, na.rm = TRUE)
> x_test$total_sqfts[is.na(x_test$total_sqfts)] = mu_total_sqfts_test
```

#Feature Scaling for test data

#Normalizing the features

```
> area_type_nor_test = (x_test$area_type - mu_area_type)/ran_area_type
> bath_nor_test = (x_test$bath - mu_bath)/ran_bath
> balcony_nor_test = (x_test$balcony - mu_balcony)/ran_balcony
> bedroom_nor_test = (x_test$bedroom - mu_bedroom)/ran_bedroom
> total_sqfts_nor_test = (x_test$total_sqfts - mu_total_sqfts)/ran_total_sqft
s
> x_zero_test = 1
> x_norm_test = data.frame(x_zero_test,area_type_nor_test,bath_nor_test,balco
ny_nor_test,bedroom_nor_test,total_sqfts_nor_test)

> predicted_price = as.matrix(x_norm_test)%*%theta
> predicted_price = as.vector(predicted_price)

> write.csv(predicted_price,"./house_price.csv",row.names = FALSE)
```