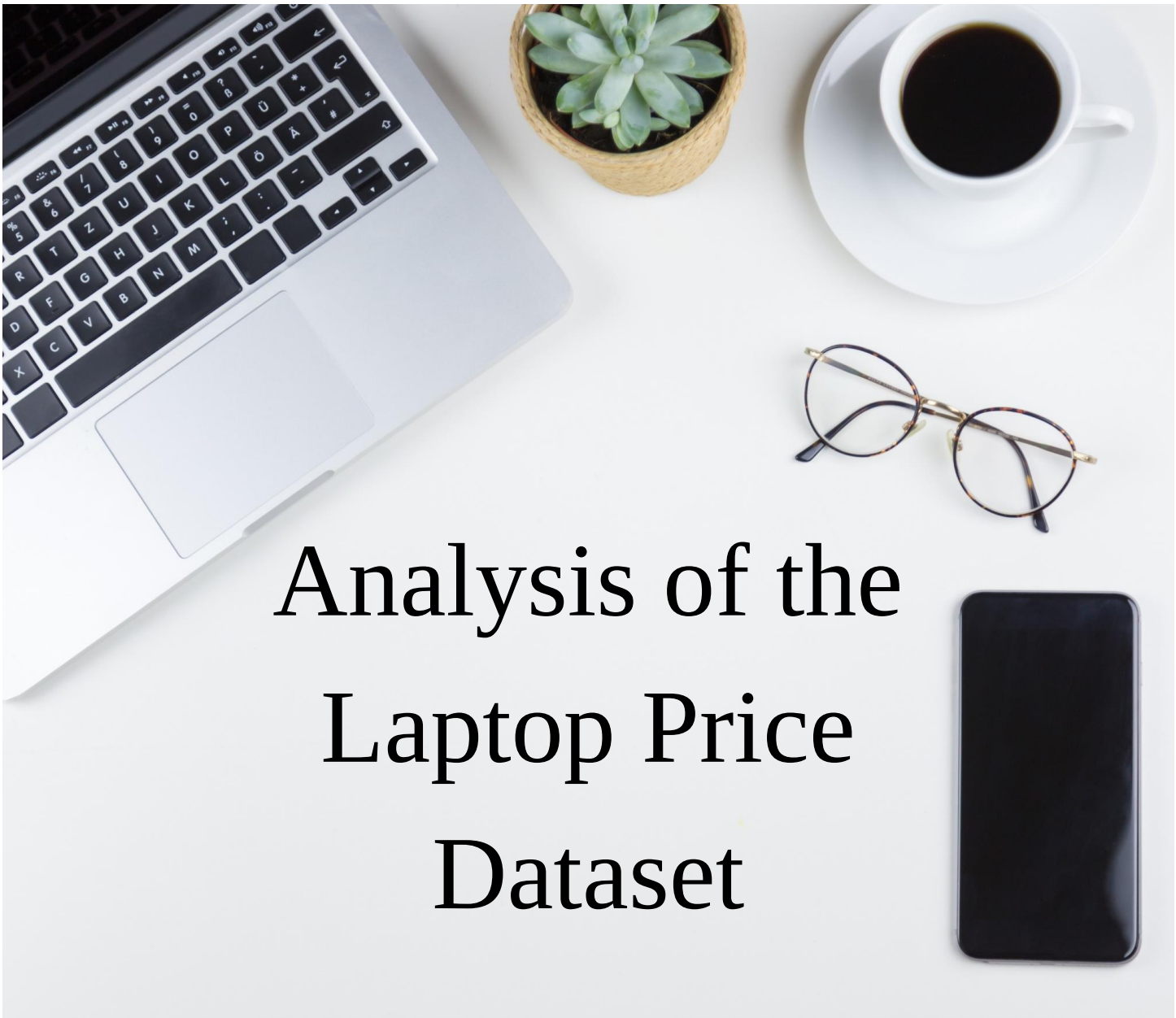


Analysis of the Laptop Price Dataset

Data Analysis Project 1

Group 4

M. D. D. De Costa	s14533
K. G. S. K. Wickramasinghe	s14594
S. P. P. M. Sudasinghe	s14510

Contents

List of Figures

Introduction

In the new normal laptops are an essential part in our day to day lives. Everyone through all the ages uses a laptop in this day and time. The laptops are more popular than the traditional desktop, mostly due to its portability.

Description of the problem

Each different laptop has its own different software and hardware that causes the price of the laptop to fluctuate. Our focus is to analyze, identify and compare the key factors that affect this change in prices and to predict how these different factors affect the price of a laptop.

Objectives

To develop a predictive model, that will predict the price of a laptop, when its features are given.

Description of the dataset

The laptop prices dataset has 1303 records and 12 variables of different laptops.

```
'data.frame':   1275 obs. of  12 variables:
 $ Company         : Factor w/ 19 levels "Acer","Apple",..: 2 2 8 2 2 1 2 2 3 1 ...
 $ Product         : chr  "MacBook Pro" "Macbook Air" "250 G6" "MacBook Pro" ...
 $ TypeName        : Factor w/ 6 levels "2 in 1 Convertible",..: 5 5 4 5 5 4 5 5 5 5 ...
 $ Inches          : num  13.3 13.3 15.6 15.4 13.3 15.6 15.4 13.3 14 14 ...
 $ ScreenResolution: chr  "IPS Panel Retina Display 2560x1600" "1440x900" "Full HD 1920x1080" "IPS Panel Retina Display 2880x1800" ...
 $ Cpu             : chr  "Intel Core i5 2.3GHz" "Intel Core i5 1.8GHz" "Intel Core i5 7200U 2.5GHz" "Intel Core i7 2.7GHz" ...
 $ Ram             : chr  "8GB" "8GB" "8GB" "16GB" ...
 $ Memory          : chr  "128GB SSD" "128GB Flash Storage" "256GB SSD" "512GB SSD" ...
 $ Gpu             : chr  "Intel Iris Plus Graphics 640" "Intel HD Graphics 6000" "Intel HD Graphics 620" "AMD Radeon Pro 455" ...
 $ OpSys           : Factor w/ 9 levels "Android","Chrome OS",..: 5 5 6 5 5 7 4 5 7 7 ...
 $ Weight          : chr  "1.37kg" "1.34kg" "1.86kg" "1.83kg" ...
 $ Price_euros     : num  1340 899 575 2537 1804 ...
```

Data Cleaning

By analysis techniques in Rstudio, we found out that there are no missing values in any of the variables. So, data imputation has not been done. Also, we managed to find out that there are 28 duplicate records in our dataset, that we removed. So, in the final dataset we have 1275 records.

From the above Rstudio output we can see that the variables ‘Ram’ and ‘Weight’ are actually quantitative variables, but they are characterized as character variables due to the existence of the words in the variable. So, in data preprocessing we removed the letters and made them numeric variables.

- Removal of the letter’s ‘GB’ from variable ‘Ram’
- Removal of the letter’s ‘kg’ from variable ‘Weight’

Feature Engineering and Preprocessing

To make better sense with our data we manipulated some of the variables so that they give meaningful correlation with the price.

- Creation of 4 variables namely ‘SSD’, ‘HDD’, ‘Hybrid’ and ‘FlashStorage’ from the variable ‘Memory’.
- Creation of ‘Xres’ and ‘Yres’ from ‘ScreenResolution’.
- Creation of Pixels Per Inch (PPI) from the variables ‘Xres’, ‘Yres’ and ‘Inches’.
- Presence of ‘IPS’ screen from variable ‘ScreenResolution’
- Presence of touch screen from variable ‘ScreenResolution’
- Value of clock speed from variable ‘Cpu’
- Creation of ‘CpuBrand’ variable from ‘Cpu’
- Creation of ‘GpuBrand’ variable from ‘Gpu’
- Transforming variable ‘OpSys’ so that it gives more meaning.

After extracting all the data from the initial variables into more meaningful variables, we decided to remove some of the initial variables. So, finally we have 16 variables.

```
'data.frame':   1275 obs. of  16 variables:
 $ Company     : Factor w/ 19 levels "Acer","Apple",..: 2 2 8 2 2 1 2 2 3 1 ...
 $ TypeName    : Factor w/ 6 levels "2 in 1 Convertible",..: 5 5 4 5 5 4 5 5 5 5 ...
 $ Ram         : int  8 8 8 16 8 4 16 8 16 8 ...
 $ OpSys       : Factor w/ 5 levels "Linux","Mac",..: 2 2 3 2 2 5 2 2 5 5 ...
 $ Weight      : num  1.37 1.34 1.86 1.83 1.37 2.1 2.04 1.34 1.3 1.6 ...
 $ Price_euros : num  1340 899 575 2537 1804 ...
 $ IPS         : Factor w/ 2 levels "No","Yes": 2 1 1 2 2 1 2 1 1 2 ...
 $ TouchScreen : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
 $ PPI         : num  227 128 141 221 227 ...
 $ cpuBrand    : Factor w/ 5 levels "AMD Processor",..: 3 3 3 4 3 1 4 3 4 3 ...
 $ ClockSpeed  : num  2.3 1.8 2.5 2.7 3.1 3 2.2 1.8 1.8 1.6 ...
 $ SSD         : num  128 0 256 512 256 0 0 0 512 256 ...
 $ HDD         : num  0 0 0 0 0 500 0 0 0 0 ...
 $ FlashStorage: num  0 128 0 0 0 0 256 256 0 0 ...
 $ Hybrid      : num  0 0 0 0 0 0 0 0 0 0 ...
 $ GpuBrand    : Factor w/ 4 levels "AMD","ARM","Intel",..: 3 3 3 1 3 1 3 3 4 3 ...
```

Analysis

Univariate Analysis

Distribution of target column (Laptop Price)

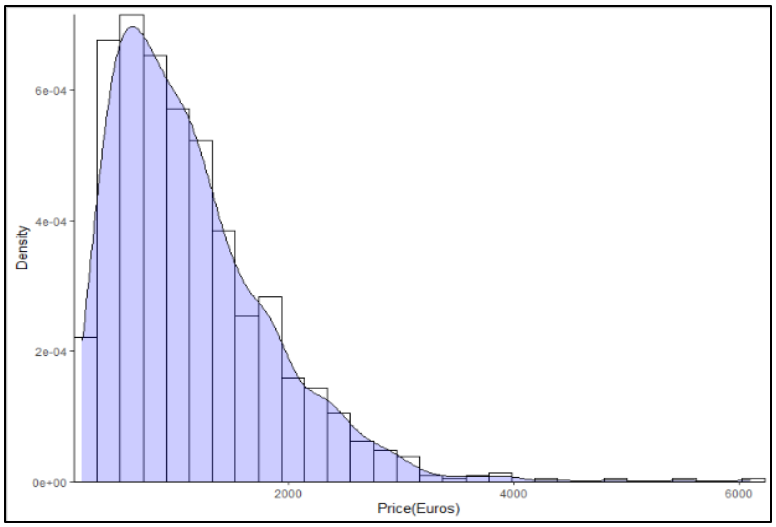


Figure 1

Since the Laptop Price is the response variable of our question it is the most important outcome of the analysis. So, it is important to understand the distribution of prices. According to this density plot, the distribution of the target variable is skewed to the right and it is obvious that laptops with low prices are sold and purchased more than the branded ones.

Distribution of Companies.

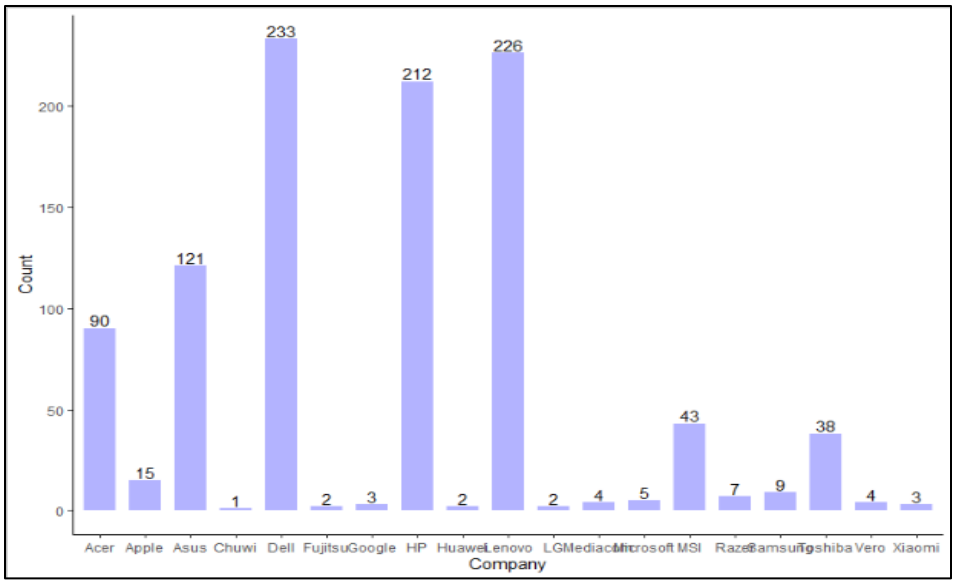


Figure 2

We want to understand the impact of laptop brand names. According to this bar plot the distribution of Lenovo, Dell, HP laptops are the highest selling laptops currently.

So, we can assume that the market price of laptops is mostly controlled by the above mentioned companies.

Bivariate Analysis

As the Laptop Prices in euros is our main response variable which needs to be analyzed to see any connection with the rest of the variables in the dataset, we have done bivariate analysis in relating the laptop prices with the rest of the variables.

Scatter plot of PPI vs Price

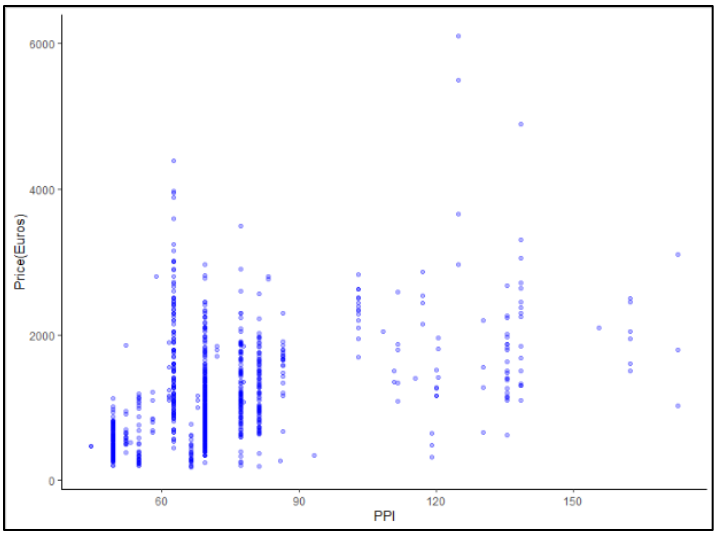


Figure 3

After analyzing the variables X_resolution(Xres) and Y_resolution(Yres) with the variable Price_euros, we saw a strong correlation among them. Since there was not a good correlation between the Inches and Price_euros, we combined the 3 variables to make a new, more meaningful variable PPI(Pixels Per Inch). As you can see the new variable has a moderately strong relationship with the response variable.

Bar plot of Memory type vs Average Price

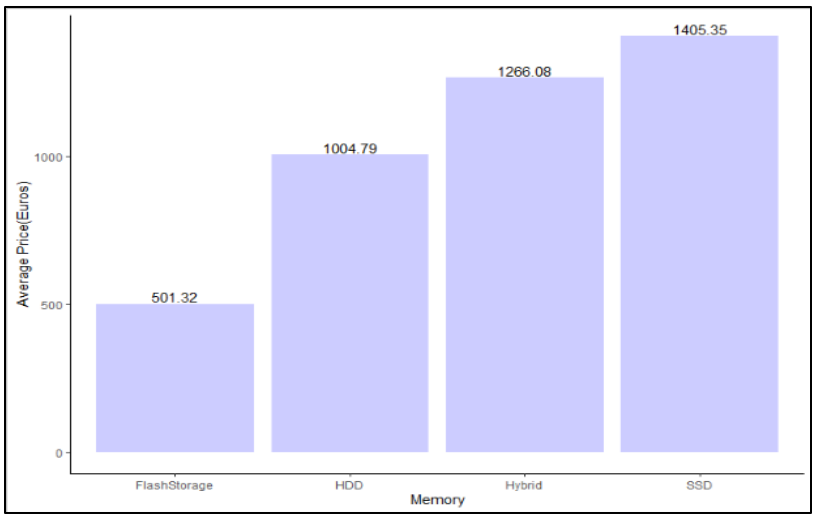


Figure 4

According to this bar plot we can see that laptops with a SDD are expensive than others. A hybrid storage is a combination of SSD and HDD storages. So, laptops with a hybrid storage is expensive than laptops with a HDD and cheaper than laptops with a SSD. As the cheapest option we have laptops with a FlashStorage.

Boxplot of Company vs Price

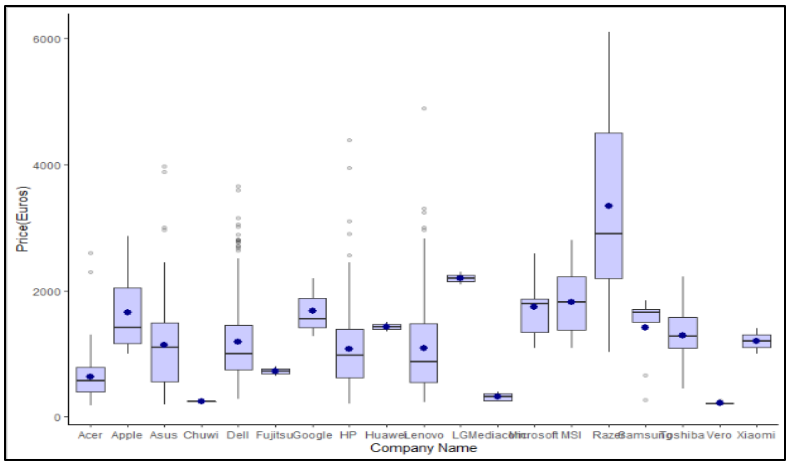


Figure 5

We plot the company relationship with price and then observed that how prices varies with different brands. According to this boxplot Razer, Apple, LG, Microsoft, Google, MSI laptops are expensive, and others are in the budget range. Since Razer is a gaming pc brand we can see that it is the most expensive one.

Boxplot of Type Name of laptop vs Price

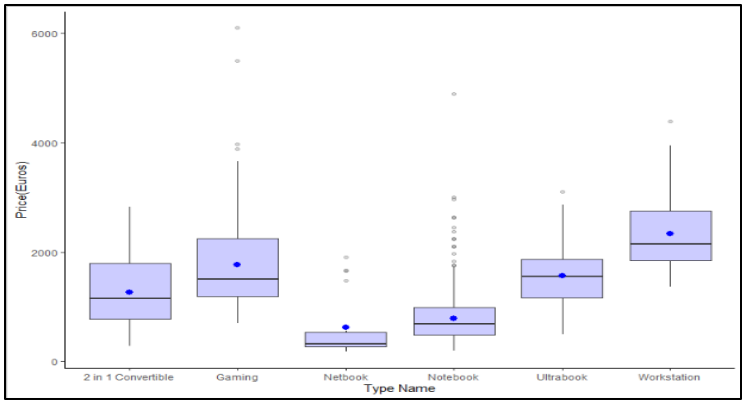


Figure 6

From the boxplot above, we can conclude that, on average, workstation and gaming laptops have a higher price than other types of laptops. This is to be expected as these types of laptops often have better specification configurations (better CPU, more memory, etc.) to meet the demands of clients in the professional workspace. Notebooks and netbooks have lower prices due to their low-powered configurations.

Bar plot of Touch screen vs Average Price

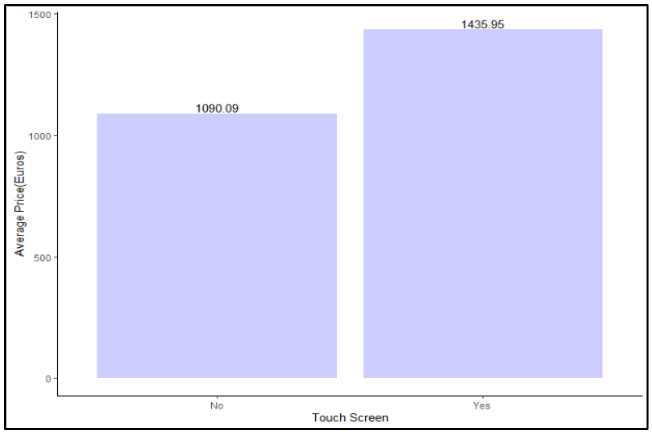
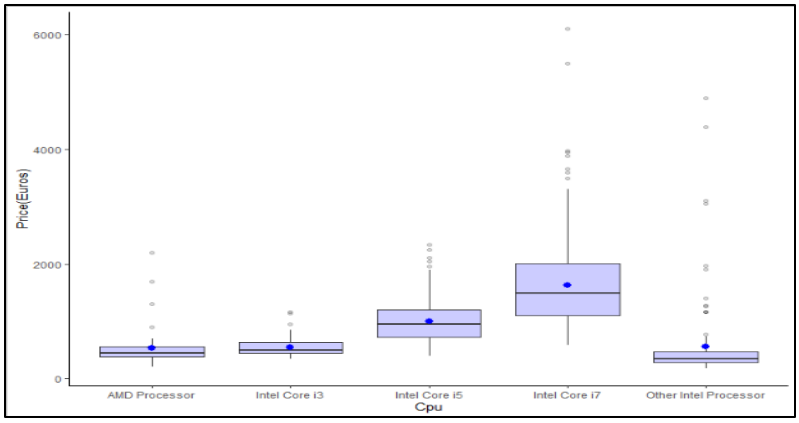


Figure 7

According to this bar plot we can say that laptops with touch screens are expensive which is true in real life.

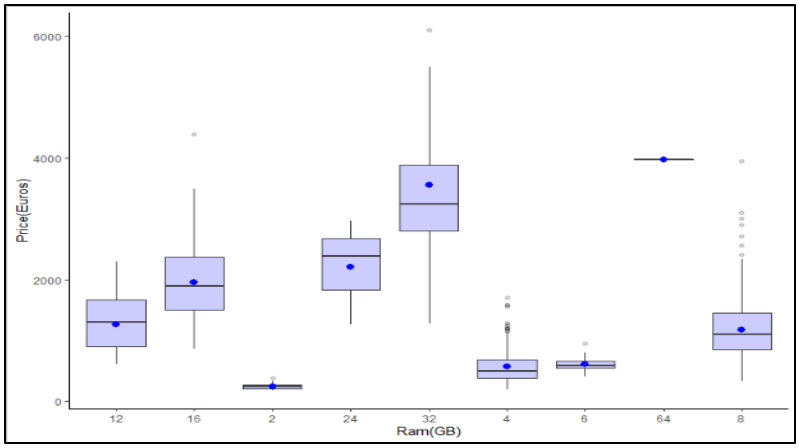
Boxplot of CPU type vs Price



We are having 5 categories in our dataset as i3, i5, i7, other Intel processors and AMD processors. We can see how the price varies with processors using this boxplot. And as obvious the price of i7 processors is high, then of i5 processor, i3 and AMD processor lies at the almost the same range. Hence price will depend on the processor.

Figure 8

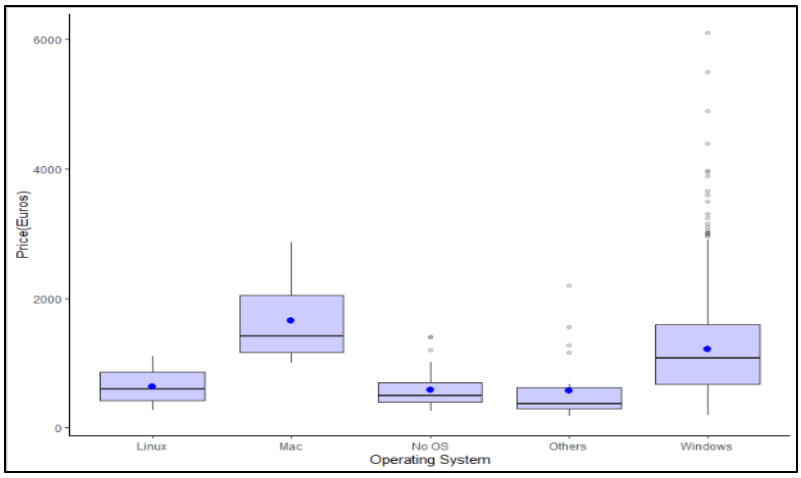
RAM vs Laptop Price



According to this boxplot we can see higher Ram capacities reflect higher prices in laptops. Price is having a very strong positive correlation with RAM.

Figure 9

Operating system vs Laptop Price



According to this boxplot as usual Mac is most expensive. Also we can clearly see that the laptops having Windows Operating system has many outlier points.

Figure 10

Suggestions for advanced analysis

Correlation matrix of variables

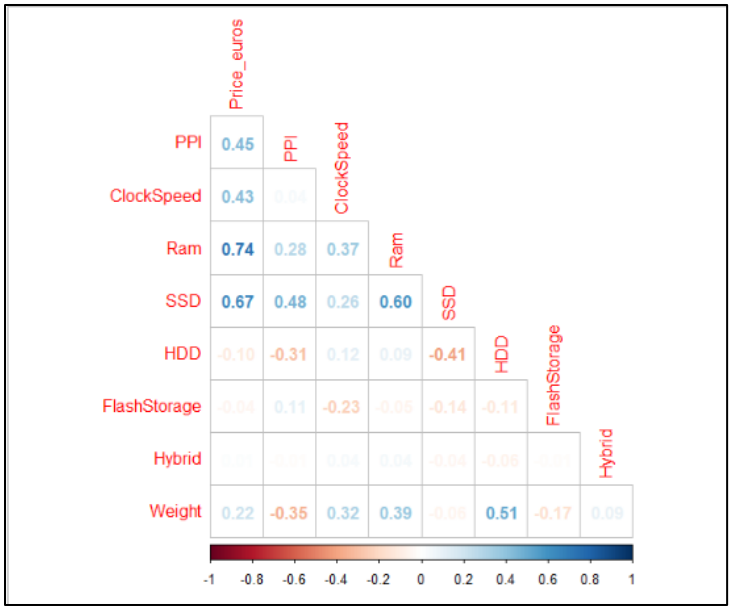


Figure 11

First, we drew a correlation matrix from our numeric variables. As you can see here clearly there exist multicollinearity in between the variables. Therefore, we can use Ridge regression, Lasso regression or Elastic Net regression when fitting the model, to mitigate that problem.

Detecting outliers

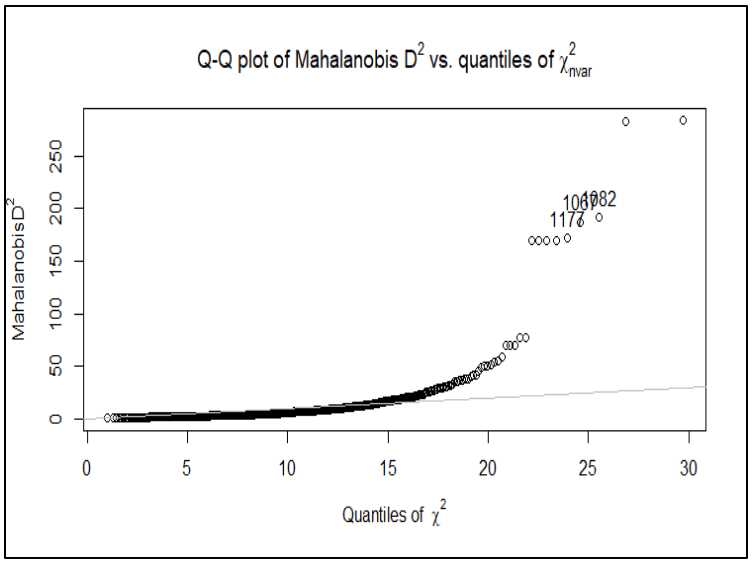


Figure 12

In the boxplots that we drew in the bivariate analysis, we clearly noticed the presence of outliers in the dataset. So, by using the Mahalanobis distance (MD) and then drawing a QQ plot we can clearly see the presence of outliers at alpha level 0.001.

By using Mahalonobis function, we can find out that there are 44 outliers in this dataset.

Model building

The histogram of the response variable(price of a laptop) is highly skewed, so we'll be using a log transformation to make the variable normal.

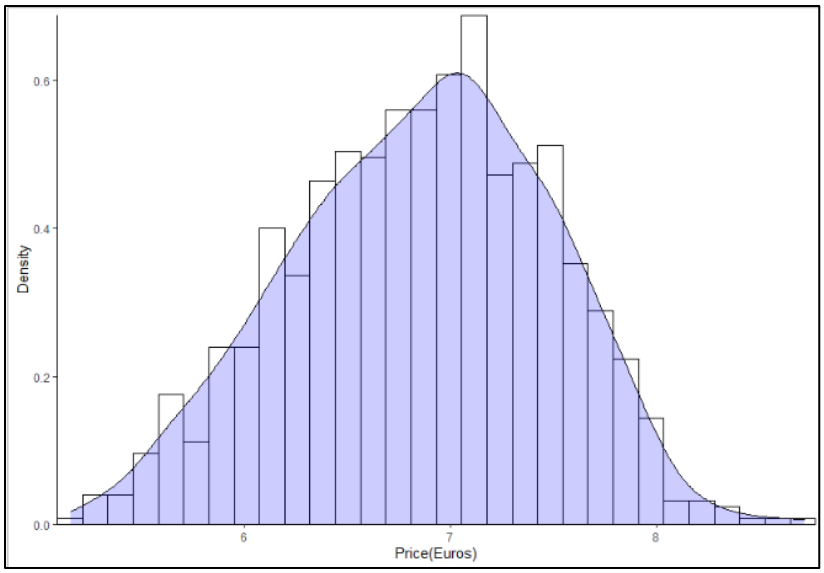


Figure 13

So, in total we'll be fitting 4 models and will be using the best model out of the 4. The 4 models are,

- Y = log(Price) with outlier data points
- Y = log(Price) without outlier data points
- Y = Price with outlier data points
- Y = Price without outlier data points

Appendix

```
#Load libraries
library(dplyr)
library(stringr)
library(mgsub)
library(ggplot2)
library(corrplot)
library(psych)

#Data cleaning
#Missing values
sapply(laptop_price,function(x) sum(is.na(x)))

#Remove laptop_ID column
laptop_price=subset(laptop_price, select=-c(laptop_ID))

#Number of duplicate rows
sum(duplicated(laptop_price))

#Remove duplicates
lap=distinct(laptop_price)
View(lap)

#Feature engineering & preprocessing
#1.ScreenResolution column
#1.1.Extract IPS column
for(i in 1:length(lap$ScreenResolution)){
  if(str_detect(lap$ScreenResolution[i], "IPS")){
    lap$IPS[i]='Yes'
  }else{
    lap$IPS[i]='No'
  }
}

#1.2.Extract Touch screen information
for(i in 1:length(lap$ScreenResolution)){
  if(str_detect(lap$ScreenResolution[i], "Touchscreen")){
    lap$TouchScreen[i]='Yes'
  }else{
    lap$TouchScreen[i]='No'
  }
}

#1.3.Extract X-axis and Y-axis screen resolution dimensions
lap$Xres=as.integer(substr(word(lap$ScreenResolution,-1),1,4))
lap$Yres=as.integer(substr(word(lap$ScreenResolution,-1),6,9))


#1.4.Calculate PPI- pixels per inch
lap$PPI=sqrt((lap$Xres)+(lap$Yres^2))/lap$Inches

#2.CPU column
#2.1.Extract CPU brand name
for(i in 1:length(lap$Cpu)){
  if(str_detect(lap$Cpu[i], "AMD")){
    lap$CpuBrand[i]='AMD Processor'
  }else if (str_detect(lap$Cpu[i], "i3")){
    lap$CpuBrand[i]='Intel Core i3'
  }else if (str_detect(lap$Cpu[i], "i5")){
    lap$CpuBrand[i]='Intel Core i5'
  }else if (str_detect(lap$Cpu[i], "i7")){
    lap$CpuBrand[i]='Intel Core i7'
  }else{
    lap$CpuBrand[i]='Other Intel Processor'
  }
}

#2.2.Extract clock speed
lap$ClockSpeed=word(lap$Cpu,-1)
lap$ClockSpeed=as.numeric(gsub('GHz','',as.character(lap$ClockSpeed)))
```

```
#3.Ram column
#3.1.Remove units in Ram & convert to integer
lap$Ram=as.integer(gsub('GB','',as.character(lap$Ram)))

#4.Memory column
lap$M= strsplit(lap$Memory, split = " ")

#4.1.Create SSD column
for(i in 1:length(lap$Memory)){
  if((str_count(lap$Memory[i], "SSD")==1) &
     (str_count(lap$Memory[i], "HDD")==1)){
    lap$SSD[i]= as.integer(mgsub::mgsub(as.character(lap$M[[i]][1]), c("GB", "TB"),
                                        c(" ", "000")))
  }else if(str_count(lap$Memory[i], "SSD")==1){
    lap$SSD[i]= as.integer(mgsub::mgsub(as.character(lap$M[[i]][1]), c("GB", "TB"),
                                        c(" ", "000")))
  }else if(str_count(lap$Memory[i], "SSD")==2){
    lap$SSD[i]= as.integer(mgsub::mgsub(as.character(lap$M[[i]][1]), c("GB", "TB"),
                                        c(" ", "000")))*2
  }else{
    lap$SSD[i]=0
  }
}

#4.2.Create HDD column
for(i in 1:length(lap$Memory)){
  if((str_detect(lap$Memory[i], "SSD")) & (str_detect(lap$Memory[i], "HDD"))){
    lap$HDD[i]=  as.integer(mgsub::mgsub(as.character(lap$M[[i]][5]),
                                         c("GB", "TB"), c(" ", "000")))
  }else if(str_detect(lap$Memory[i], "HDD")){
    lap$HDD[i]=  as.integer(mgsub::mgsub(as.character(lap$M[[i]][1]),
                                         c("GB", "TB"), c(" ", "000")))
  }else{
    lap$HDD[i]=0
  }
}


#4.3.Create FlashStorage column
for(i in 1:length(lap$Memory)){
  if(str_detect(lap$Memory[i], "Flash")){
    lap$FlashStorage[i]=  as.integer(mgsub::mgsub(as.character(lap$M[[i]][1]),
                                                  c("GB", "TB"), c(" ", "000")))
  }else{
    lap$FlashStorage[i]=0
  }
}

#4.4.Create Hybrid column
for(i in 1:length(lap$Memory)){
  if((str_detect(lap$Memory[i], "SSD")) & (str_detect(lap$Memory[i], "Hybrid"))){
    lap$Hybrid[i]=  as.integer(mgsub::mgsub(as.character(lap$M[[i]][5]),
                                            c("GB", ".0TB"), c(" ", "000")))
  }else if(str_detect(lap$Memory[i], "Hybrid")){
    lap$Hybrid[i]= as.integer(mgsub::mgsub(as.character(lap$M[[i]][1]),
                                           c("GB", ".0TB"), c(" ", "000")))
  }else{
    lap$Hybrid[i]=0
  }
}

#5.GPU column
#5.1.Exract Gpu Brand
lap$GpuBrand=word(lap$Gpu,1)

#6.OpSys column
#6.1.Transform opSys
for(i in 1:length(lap$OpSys)){
  if(str_detect(lap$OpSys[i], "Windows")){
    lap$OpSys[i]='Windows'
  }else if (str_detect(lap$OpSys[i], "macOS")){
    lap$OpSys[i]='Mac'
  }else if (str_detect(lap$OpSys[i], "Mac OS X")){
    lap$OpSys[i]='Mac'
  }else if(str_detect(lap$OpSys[i], "Linux")){
    lap$OpSys[i]='Linux'
  }else if(str_detect(lap$OpSys[i], "No OS")){
    lap$OpSys[i]='No OS'
  }else{
    lap$OpSys[i]='Others'
  }
}

#7.Weight Column
#7.1.Remove units in Weight & convert to numeric
lap$Weight=as.numeric(gsub('kg','',as.character(lap$Weight)))

#Remove unnecessary variables
lap=subset(lap, select=-c(ScreenResolution,Cpu,Memory,
                          Gpu,M,Xres,Yres,Inches,Product))
```

```r
#Convert into Factors
lap$OpSys = as.factor(lap$OpSys)
lap$IPS = as.factor(lap$IPS)
lap$TouchScreen = as.factor(lap$TouchScreen)
lap$CpuBrand = as.factor(lap$CpuBrand)
lap$Company = as.factor(lap$Company)
lap$TypeName = as.factor(lap$TypeName)
lap$GpuBrand = as.factor(lap$GpuBrand)

str(lap)

#Divide data set into training and testing
set.seed(100)
indexes = sample(1:nrow(lap),size=0.2*nrow(lap))
testset=lap[indexes,]
trainset=lap[-indexes,]
View(trainset)
View(testset)

#Analysis
#Univariate Analysis
# 1) Distribution of price
ggplot(trainset, aes(x=Price_euros)) +
  geom_histogram(aes(y=..density..), colour="black", fill="white",bins=30)+
  geom_density(alpha=0.2, fill="blue") +
  labs(x="Price(Euros)", y = "Density")+
  scale_x_continuous(expand = c(0, 0)) + scale_y_continuous(expand = c(0, 0))+
  theme(
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.background=element_blank(), axis.line = element_line(colour = "black")
  )


# 2) Bar Plot - Company
ggplot(trainset, aes(x=factor(Company)))+
  geom_bar(stat="count", width=0.7, alpha=0.3, fill="blue")+
  geom_text(aes(label = ..count..), stat = "count",vjust = -0.2)+
  labs(x="Company", y = "Count")+
  theme(
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.background = element_blank(), axis.line = element_line(colour = "black")
  )

#Bivariate Analysis
# 1) Scatter Plot- PPI vs Price
ggplot(trainset, aes(x = PPI, y = Price_euros)) +
  geom_point(alpha=0.3,colour = "blue")+
  labs(x="PPI", y = "Price(Euros)")+
  theme(
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.background=element_blank(), axis.line = element_line(colour = "black")
  )

# 2) Bar Plot- Memory vs Average Price
avg_price_SSD=mean(trainset$Price_euros[trainset$SSD!=0])
avg_price_HDD=mean(trainset$Price_euros[trainset$HDD!=0])
avg_price_FS=mean(trainset$Price_euros[trainset$FlashStorage!=0])
avg_price_HB=mean(trainset$Price_euros[trainset$Hybrid!=0])

mem=data.frame(Memory=c('SSD','HDD','FlashStorage','Hybrid'),
               AvgPrice=c(avg_price_SSD,avg_price_HDD,avg_price_FS,avg_price_HB))

ggplot(mem, aes(x = Memory, y =AvgPrice))+
  geom_bar(stat= 'identity' , alpha=0.2, fill="blue")+
  geom_text(aes(label = round(AvgPrice,2), vjust = -0.2))+
  labs( x="Memory", y = "Average Price(Euros)")+
  theme(
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.background = element_blank(), axis.line = element_line(colour = "black")
  )

# 3) Boxplot- Company vs Price
ggplot(trainset, aes(x=Company, y=Price_euros)) +
  geom_boxplot(alpha=0.2,fill='blue')+
  stat_summary(fun.y=mean, geom="point", shape=20,
               size=4,col='dark blue',fill='dark blue')+
  labs(x="Company Name", y = "Price(Euros)")+
  theme(
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.background = element_blank(), axis.line = element_line(colour = "black"),
  )
```

```
# 4) Boxplot-TypeName vs Price
ggplot(trainset, aes(x=TypeName, y=Price_euros)) +
  geom_boxplot(alpha=0.2,fill='blue')+
  stat_summary(fun.y=mean, geom="point", shape=20, size=4, col='blue',fill='blue')+
  labs(x="Type Name", y = "Price(Euros)")+
  theme(
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.background = element_blank(), axis.line = element_line(colour = "black")
  )

# 5) Bar Plot- Touch Screen vs Average Price
TS=c('No','Yes')
avg_price_TS=numeric(0)
for(i in 1:length(TS)){
  avg_price_TS[i]=mean(trainset$Price_euros[trainset$TouchScreen==TS[i]])
}
touchScreen=data.frame(TS,avg_price_TS)

ggplot(touchScreen, aes(x = TS, y =avg_price_TS))+
  geom_bar(stat= 'identity' , alpha=0.2, fill="blue")+
  geom_text(aes(label = round(avg_price_TS,2), vjust = -0.2))+
  labs(x="Touch Screen", y = "Average Price(Euros)")+
  theme(
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.background = element_blank(), axis.line = element_line(colour = "black")
  )

# 6) Box Plot- CPU vs price
ggplot(trainset, aes(x=CpuBrand, y=Price_euros)) +
  geom_boxplot(alpha= 0.2, fill='blue')+
  stat_summary(fun.y=mean, geom="point", shape=20, size=4, col='blue',fill='blue')+
  labs(x="Cpu", y = "Price(Euros)")+
  theme(
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.background = element_blank(), axis.line = element_line(colour = "black")
  )

# 7) Boxplot- RAM vs price
ggplot(trainset, aes(x=as.character(Ram), y=Price_euros)) +
  geom_boxplot(alpha=0.2, fill='blue')+
  stat_summary(fun.y=mean, geom="point", shape=20, size=4, col='blue',fill='blue')+
  labs(x="Ram(GB)", y = "Price(Euros)")+
  theme(
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.background = element_blank(), axis.line = element_line(colour = "black")
  )


# 8) Boxplot- Operating system vs Price
ggplot(trainset, aes(x=OpSys, y=Price_euros)) +
  geom_boxplot(alpha=0.2,fill='blue')+
  stat_summary(fun.y=mean, geom="point", shape=20, size=4, col='blue',fill='blue')+
  labs(x="Operating System", y = "Price(Euros)")+
  theme(
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.background = element_blank(), axis.line = element_line(colour = "black"),
  )

#Multivariate Analysis
#Advanced Analysis Techniques
#Extract only numeric variables
numericVar=subset(trainset, select=c(Price_euros,PPI,ClockSpeed,
                                     Ram,SSD,HDD,FlashStorage,
                                     Hybrid,Weight))
View(numericVar)
#corr matrix
corrplot(cor(numericVar),method='number',diag = FALSE, type='lower')

#log-price
ggplot(trainset, aes(x=log(Price_euros))) +
  geom_histogram(aes(y=..density..), colour="black", fill="white",bins=30)+
  geom_density(alpha=.2, fill="blue") +
  labs(x="Price(Euros)", y = "Density")+
  scale_x_continuous(expand = c(0, 0)) + scale_y_continuous(expand = c(0, 0))+
  theme(
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.background=element_blank(), axis.line = element_line(colour = "black")
  )


#Detect Outliers
#Graphical representation of Outliers
d2=outlier(numericVar)

#Check what are the rows containing outliers
md = mahalanobis(numericVar, center = colMeans(numericVar),
                 cov = cov(numericVar))
cutoff = (qchisq(p = 0.999, df = ncol(numericVar)))
names_outliers = which(md > cutoff)
names_outliers
length(names_outliers)
```