EDA

Nicholas Allen, Surya Maddali, Jake Adams

Introduction:

In recent decades, cell phones have become a hot commodity around the world. The idea of calling with the tips of your fingers was a revolutionary idea that continues to set the standard for telecommunications. With advancements to cell phones, one question that is always present is pricing There may be certain factors that affect cell phone pricing such as storage, camera capabilities, and battery power. The goal of this project is to assess that, seeing if certain features of phones affect pricing in a significant way. It is an interesting and important question to answer because it can inform others about what phone features matter the most to companies that make phones as well as inform us about what features matter the most to a phone's functionality when looking to buy one. Machine learning is a reasonable approach to tackle this question because it can give us insight into why or how phones are purchased. Moreover, it can help us predict phone prices in the future based on what features they possess, which would be informed by past data on this exact matter. In other words, it would help readers assess what features are continuing to affect the price of the phone the most in the present.

Illustration:

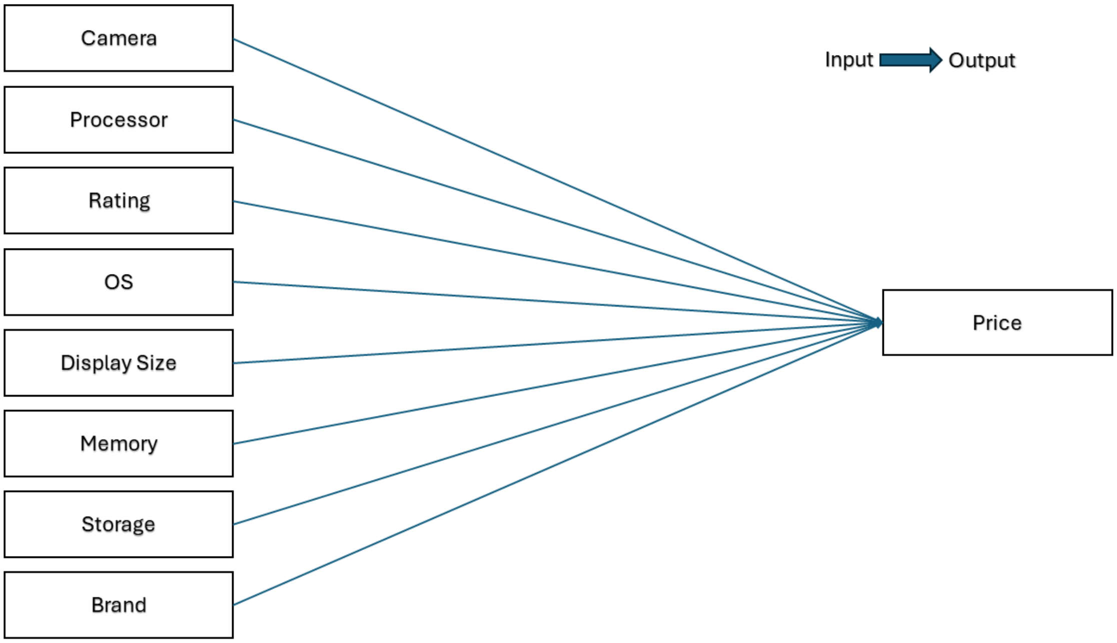


Figure 1: Elephant

Background and Related Works:

We looked at an article from IEEE Xplore. This article was about predicting mobile phone prices using a data set from kaggle. This article differed from ours because they were predicting phone prices with classification. They had their y variable in as a factor with 4 levels. The levels were form "low cost" to "very high cost". Some examples of their x variables were battery power and clock speed. They used several different models to predict phone price such as a decision tree and SVM. Their most accurate model was SVM with an accuracy of 94.8%.

Reference: N. Hu, "Classification of Mobile Phone Price Dataset Using Machine Learning Algorithms," 2022 3rd International Conference on Pattern Recognition and Machine Learning (PRML), Chengdu, China, 2022, pp. 438-443, doi: 10.1109/PRML56267.2022.9882236. keywords: {Support vector machines;Machine learning algorithms;Random access memory;Machine learning;Feature extraction;Mobile handsets;Batteries;computer science;machine learning;classification;price prediction},

Data Processing:

We loaded in the data sets though the readxl package

```
library(readxl)
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.4.3     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```
df <- read_excel('smartphones_-_smartphones.xlsx')
df2 <- read_csv('Sales.csv')
```

```
Rows: 3114 Columns: 12
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (7): Brands, Models, Colors, Memory, Storage, Camera, Mobile
dbl (5): Rating, Selling Price, Original Price, Discount, discount percentage

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

First Dataset

The first data set looked like this before processing.

```
head(df)
```

```
# A tibble: 6 x 11
  model    price rating sim   processor ram   battery display camera card  os
  <chr>    <chr>  <dbl> <chr> <chr>     <chr> <chr>   <chr>   <chr>  <chr> <chr>
1 OnePlus~ 54,~      89 Dual~ Snapdrag~ 12 G~ 5000 m~ 6.7 in~ 50 MP~ Memo~ Andr~
```

```
2 OnePlus~  19,~     81 Dual~ Snapdrag~ 6 GB~ 5000 m~ 6.59 i~ 64 MP~ Memo~ Andr~
3 Samsung~  16,~     75 Dual~ Exynos 1~ 4 GB~ 5000 m~ 6.6 in~ 50 MP~ Memo~ Andr~
4 Motorol~  14,~     81 Dual~ Snapdrag~ 6 GB~ 5000 m~ 6.55 i~ 50 MP~ Memo~ Andr~
5 Realme ~  24,~     82 Dual~ Dimensit~ 6 GB~ 5000 m~ 6.7 in~ 108 M~ Memo~ Andr~
6 Samsung~  16,~     80 Dual~ Snapdrag~ 6 GB~ 5000 m~ 6.6 in~ 50 MP~ Memo~ Andr~
```

It is a tabular data set on some mobile phones. Some examples of columns in the data set are mobile which represents the name of the phone and the price of the phone.

To start off we took out the model column because it represented the names of the phones which will not impact the price. We also took out the sim column.

```r
df <- df %>% drop_na() %>% select(!model) %>% select(!sim)
```

Cleaning battery column

We extracted the battery life of each phone in mAH and made the column numeric

```r
df <- df %>%
  mutate(battery = gsub(pattern = "mAh Battery|with|(?:[0-9]){1,3}W|Fast Charging", replaceme
```

```
Warning: There was 1 warning in `mutate()`.
i In argument: `battery = .Primitive("as.double")(battery)`.
Caused by warning:
! NAs introduced by coercion
```

Cleaning processor variable

We extracted the power of the processor in GHz. We then made the column numeric

```r
df$processor <- str_extract(df$processor, "\\d+\\.?\\d*\\s*GHz|\\d+\\s*GHz")

df$processor <- gsub("GHz", "", df$processor)

df <- df %>% drop_na() %>% mutate_at('processor', as.numeric) %>%rename('processor GHz)'='pr
```

Cleaning os column

We noticed that because the data was unclean, some of the values that should be in the os column were in the card column. We put these value in the os column and removed the card column after. We also made the os column a factor.

```r
for (i in 1:nrow(df)){
  if (df[i,9] == 'No FM Radio'){
    df[i,9] <- df[i,8]
  }
  else if (df[i,9] == 'Bluetooth'){
    df[i,9] <- df[i,8]
  }
}

df <- df %>% select(!card) %>% mutate_at('os', as.factor)
```

Cleaning camera column

We extracted the amount of mega pixels in the front camera of each phone. We made this column numeric

```r
df$camera <- str_extract(df$camera, '[0-9]{1,2} MP Front Camera')
df$camera <- str_extract(df$camera, '[0-9]{1,2}')
df <- df %>% mutate_at('camera', as.numeric) %>% rename('f camera MP'='camera') %>% drop_na(
```

Cleaning ram column

We extracted the ram of the phones in GB and made it a factor because phones only have a few preset values for their ram

```r
df$ram <- str_extract(df$ram, '[0-9]{1,2} GB')
df <- df %>% mutate_at('ram', as.factor)
```

Cleaning Display column

We extracted the display size and the Hz of the display and turned that into two new columns. We made these new columns numeric and removed the original

```
df <- df %>% mutate(displaySize = as.numeric(str_extract(df$display, "\\b\\d+\\.\\d+\\b")))
```

```
df <- df %>% mutate(displayHz = as.numeric(str_extract(df$display, "\\b\\d+(?=\\s*Hz)")))

df <- df %>% select(!display)
```

Cleaning Price column

We converted the value in rupees to dollars to make it easier to understand for our audience

```
df <- df %>%
  mutate(price = gsub(",", "", price))
df$price <- sub("\\ ", "", df$price)
df$price <- as.numeric(df$price)

df <- df %>%
  mutate(price = round(price / 83.41, digits = 2)) %>% rename('price $'='price') %>% drop_na
```

General Analysis

After Cleaning:

```
head(df)
```

```
# A tibble: 6 x 9
  `price $` rating `processor GHz)` ram   `battery mAh` `f camera MP` os
      <dbl>  <dbl>            <dbl> <fct>         <dbl>         <dbl> <fct>
1      659.     89              3.2 12 GB          5000            16 Android v~
2      240.     81              2.2 6 GB           5000            16 Android v~
3      198.     75              2.4 4 GB           5000            13 Android v~
4      180.     81              2.2 6 GB           5000            16 Android v~
5      300.     82              2.6 6 GB           5000            16 Android v~
6      204.     80              2.2 6 GB           5000             8 Android v~
# i 2 more variables: displaySize <dbl>, displayHz <dbl>
```

```
summary(df)
```

```
    price $            rating       processor GHz)        ram
 Min.   :  88.71   Min.   :62.00   Min.   :1.600   8 GB   :231
 1st Qu.: 191.81   1st Qu.:78.00   1st Qu.:2.200   6 GB   :144
 Median : 275.73   Median :82.00   Median :2.400   4 GB   : 81
 Mean   : 345.50   Mean   :81.37   Mean   :2.526   12 GB  : 38
 3rd Qu.: 395.62   3rd Qu.:85.00   3rd Qu.:2.900   16 GB  :  7
 Max.   :2877.34   Max.   :89.00   Max.   :3.220   3 GB   :  3
                                                   (Other):  2
  battery mAh     f camera MP                 os       displaySize
 Min.   : 3095   Min.   : 1.00   Android v12  :255   Min.   :5.900
 1st Qu.: 4600   1st Qu.:13.00   Android v11  :150   1st Qu.:6.500
 Median : 5000   Median :16.00   Android v13  : 68   Median :6.600
 Mean   : 4934   Mean   :18.16   Android v10  : 17   Mean   :6.593
 3rd Qu.: 5000   3rd Qu.:20.00   Android v10.0:  4   3rd Qu.:6.670
 Max.   :22000   Max.   :60.00   iOS v15      :  3   Max.   :6.950
                                 (Other)      :  9
   displayHz
 Min.   : 90.0
 1st Qu.: 90.0
 Median :120.0
 Mean   :110.5
 3rd Qu.:120.0
 Max.   :240.0
```

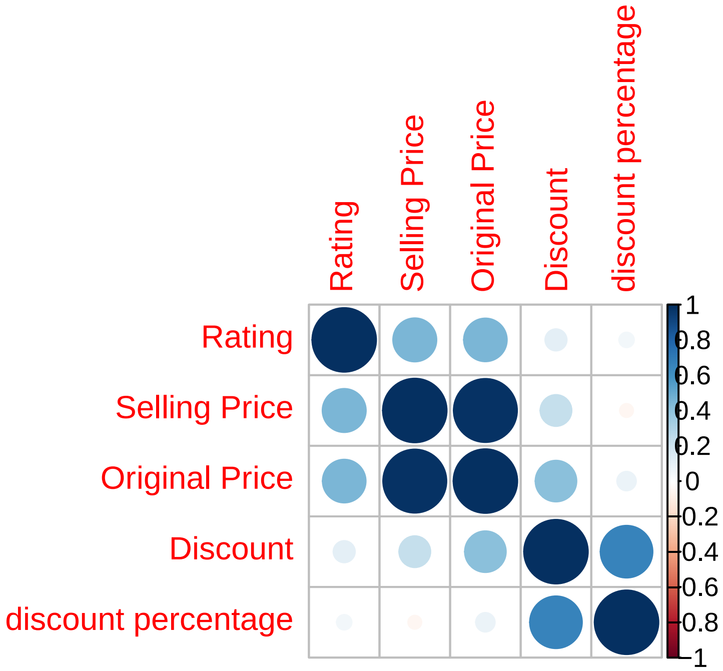
```
library(corrplot)
```

```
corrplot 0.92 loaded
```

```
numeric_data <- df2 %>%
  select_if(is.numeric) %>% drop_na()

correlation_matrix <- cor(numeric_data)

corrplot(correlation_matrix, method = "circle")
```



Second Dataset

The first dataset looked like this before processing

```
head(df2)
```

```
# A tibble: 6 x 12
  Brands  Models      Colors        Memory Storage Camera Rating `Selling Price`
  <chr>   <chr>       <chr>         <chr>  <chr>   <chr>   <dbl>           <dbl>
1 SAMSUNG GALAXY M31S Mirage Black  8 GB   128 GB  Yes       4.3           19330
2 Nokia   3.2         Steel         2 GB   16 GB   Yes       3.8           10199
3 realme  C2          Diamond Black 2 GB   <NA>    Yes       4.4            6999
4 Infinix Note 5      Ice Blue      4 GB   64 GB   Yes       4.2           12999
5 Apple   iPhone 11   Black         4GB    64 GB   Yes       4.6           49900
6 GIONEE  L800        Black         8 MB   16 MB   Yes       4              2199
# i 4 more variables: `Original Price` <dbl>, Mobile <chr>, Discount <dbl>,
#   `discount percentage` <dbl>
```

It is also a tabular data set with information on mobile phones. This data set differs from the first because it has less columns that are useful for predicting price but it has more rows.

Cleaning P1

To start we removed unneeded columns. These were models, Camera, selling price, mobile, discount, and discount percentage. We then make all the column names lowercase. We then made all the data in the colors and brands columns lowercase. We then removed the underscore from the original_price column name. We then converted the price to dollars. We them made the memory, brands, and storage columns factors.

```
df2 <- df2 %>% drop_na()

df2 <- df2[,-c(2,3,6,8,10,11,12)]

names(df2) <- tolower(names(df2))

df2$brands <- tolower(df2$brands)

df2<- rename(df2, original_price = "original price")

df2 <- df2 %>% mutate(original_price = df2$original_price * 0.012)

df2$memory <- as.factor(df2$memory)

df2$storage <- as.factor(df2$storage)

df2$brands <- as.factor(df2$brands)
```

General Analysis

After cleaning:

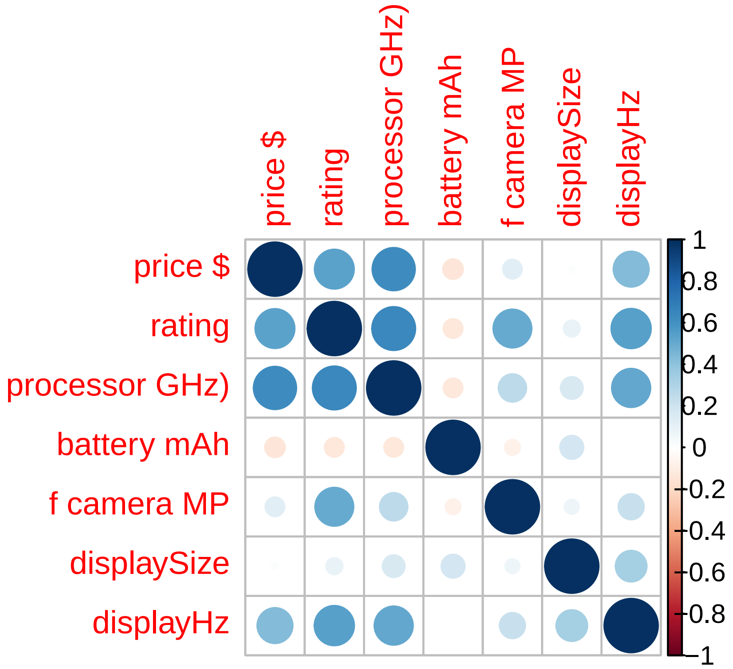
```
head(df2)
```

```
# A tibble: 6 x 5
  brands  memory storage rating original_price
  <fct>   <fct>  <fct>    <dbl>          <dbl>
1 samsung 8 GB   128 GB     4.3          252.
2 nokia   2 GB   16 GB      3.8          122.
3 infinix 4 GB   64 GB      4.2          156.
4 apple   4GB    64 GB      4.6          599.
5 gionee  8 MB   16 MB      4             26.4
6 apple   3 GB   64 GB      4.6          575.
```

```
summary(df2)
```

```
     brands          memory          storage          rating        original_price
 samsung:685   4 GB    :711   64 GB  :757   Min.   :2.300   Min.   :  12.0
 apple  :319   3 GB    :479   128 GB :720   1st Qu.:4.100   1st Qu.: 124.7
 realme :281   6 GB    :444   32 GB  :545   Median :4.300   Median : 195.6
 oppo   :251   2 GB    :376   16 GB  :312   Mean   :4.241   Mean   : 319.9
 xiaomi :191   8 GB    :326   256 GB :216   3rd Qu.:4.400   3rd Qu.: 360.0
 nokia  :184   1 GB    :193   8 GB   :133   Max.   :5.000   Max.   :2280.0
 (Other):986   (Other):368   (Other):214
```

```
numeric_data2 <- df %>%
  select_if(is.numeric)
correlation_matrix2 <- cor(numeric_data2)
corrplot(correlation_matrix2, method = "circle")
```


price $
rating
processor GHz)
battery mAh
f camera MP
displaySize
displayHz
price $
rating
processor GHz)
battery mAh
f camera MP
displaySize
displayHz