Mobile Phones Selling Price Report

Proposal for final project (MDSA Winter 2023)

Yongpeng Fu (10182778), Rudy Brown (10057171), Jose Palacios (30190988),
Stuart Finley(30191070), Andrii Voitkiv(30199373)

2023-02-18

Contents

Introduction

Mobile phones are everywhere, so are the prices. Despite still having the word “phone” in the name, a typical modern smartphone has much more features than just to make and receive calls. They are boasting a staggering range of applications like brand, memory, storage, camera, resolution, just to name a few. All these cutting edge technology and features packed in one little device does not come without a cost. A 2020 review of premium mobile phones shows a staggering *490%* rise in the last two decades.

With so many mobile phones on the market, it can be difficult to decide which one you want to buy. As a customer, we are particularly interested in finding some relation between all these features and its selling price. To this purpose, we collected the MobilePhone’s dataset from Kaggle and apply a set of statistical analysis hoping to answer some guiding questions:

1. Can we estimate the average price for mobile phones?
2. What is the impact of each mobile phone’s feature on the selling price?
3. Can a classification model to distinguish the selling price range?
4. Can we build a decent model to predict the selling price for a mobile phone?

Dataset

The initial dataset consists of 8 columns and 28,036 rows and no missing values. These 8 columns are:

- **Model**: categorical variables with sub-classes. These names include the color of the unit and its storage capacity. The latter being also listed as a separate column. - **Independent Variable**
- **Company**: categorical variable. Name of the phone’s manufacturer. - **Independent Variable**
- **Price**: continuous variable. Units in Indian Rupees. - **Dependent Variable**
- **Rating**: continuous variable. Units in Indian Rupees. - **Independent Variable**
- **Number of ratings**: discrete variable: a simple count. - **Independent Variable**
- **Total reviews**: discrete variable: a simple count. - **Independent Variable**
- **RAM size**: categorical variable. RAM specification of the phone. - **Independent Variable**
- **ROM size**: categorical variable. Storage (non-volatile memory) capacity of the phone. - **Independent Variable**

Some initial steps can be completed to clean the dataset and create new variables which can be used in our analysis. The initial steps for cleaning the dataset are as follows:

1. Remove any duplicates in the dataset;
2. Because **Model** column contains sub-class of a mobile phone, we decide to further break it down to *Model* and *Color*;
3. Convert all units from **RAM size** and **ROM size** measure to GB and then remove unit suffix;
4. Add additional column to segment the **Price** into 4 different levels;
5. Add additional column to determine if a phone has %G feature or not based on **Model** information.

```
mobile_dataset <- read.csv("./Updated_Mobile_Dataset.csv")
#Break down Model column into Model only and Color columns
#Step 1: remove the Company name from Model
Model_no_Company <- stringr::str_remove(mobile_dataset$Model,mobile_dataset$Company)
↪  %>% trimws(., which = c("both"))
#Step 2: remove anything after parentheses to get Model only info
mobile_dataset$Model_Only <- stringr::str_replace(Model_no_Company, "
↪  \\s*\\([^\\)]+\\)", "")
#Step 3: Get the color information from inside the last ()
#Step 3.1: Get the parenthesis and what is inside from the last ()
```

```
Model_no_Company_parenthesis <- stringr::str_extract(Model_no_Company,
↪  "(?<=\\()([^()]*?)(?=\\)[^()]*$)")
#step 3.2: Get the color by just retaining the info before ,
mobile_dataset$Color <- gsub(",.*$", "", Model_no_Company_parenthesis)
#Step 4: Remove duplicated rows in the dataset
mobile_dataset <- mobile_dataset[!duplicated(mobile_dataset),]
#Step 5: cut the price based on the percentile into 4 different levels
mobile_dataset <- mobile_dataset %>% mutate(Price_Level = ntile(Price, n = 4))
#Step 5.1: map each number level to the character
from <- c(1,2,3,4)
to <- c("Low", "Medium", "High", "Very High")
mobile_dataset$Price_Level <- mapvalues(mobile_dataset$Price_Level, from = from, to
↪  = to)


#Step 6: Get the numeric part of RomSize (remove GB and MB, but convert MB to GB),
↪  discard any record that no numeric in RomSize
#Step 6.1: there are some data input errors for RamSize and RomSize. In the records
↪  where RomSize is "Not Known" are swapped with RamSize, so we need to correct
↪  that.
RamSize_temp <- ifelse(mobile_dataset$RomSize == "Not Known", "0 GB",
↪  mobile_dataset$RamSize)
mobile_dataset$RomSize <- ifelse(mobile_dataset$RomSize == "Not Known",
↪  mobile_dataset$RamSize, mobile_dataset$RomSize)
mobile_dataset$RamSize <- RamSize_temp
#Step 6.2: split RomSize into two columns with size number and unit, and convert MB
↪  to 1/1000GB, KB to 1/1000000GB
mobile_dataset$RamSize_Ori <- mobile_dataset$RamSize
mobile_dataset$RomSize_Ori <- mobile_dataset$RomSize
mobile_dataset <- mobile_dataset %>% separate(RomSize, c("RomSize_num",
↪  "RomSize_Unit")) %>% mutate(RomSize_Unit= mapvalues(.$RomSize_Unit, from =
↪  c("GB", "MB", "KB"), to = c(1, 1/1000, 1/1000000)))
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 1 rows [573].
```

```
#step 6.3: remove any rows that are not numeric value for RomSize
mobile_dataset <- mobile_dataset[!is.na(as.numeric(mobile_dataset$RomSize_num)),]
```

```
## Warning in
## `[.data.frame`(mobile_dataset, !is.na(as.numeric(mobile_dataset$RomSize_num)), :
## NAs introduced by coercion
```

```
mobile_dataset$RomSize_num <- as.numeric(mobile_dataset$RomSize_num)
mobile_dataset$RomSize_Unit <-
↪  ifelse(is.na(as.numeric(mobile_dataset$RomSize_Unit)), 0,
↪  as.numeric(mobile_dataset$RomSize_Unit))
#Step 6.4: generate the final column RomSize_inGB
mobile_dataset$RomSize_inGB <- mobile_dataset$RomSize_num *
↪  mobile_dataset$RomSize_Unit


#Step 7: Get the numeric part of RamSize (remove GB and MB, but convert MB to GB),
↪  discard any record that no numeric in RamSize
```

```
#Step 7.1: split RamSize into two columns with size number and unit, and convert MB
↪  to 1/1000GB
mobile_dataset <- mobile_dataset %>% separate(RamSize, c("RamSize_num",
↪  "RamSize_Unit")) %>% mutate(RamSize_Unit= mapvalues(.$RamSize_Unit, from =
↪  c("GB", "MB"), to = c(1, 1/1000)))
#step 7.2: remove any rows that are not numeric value for RamSize
mobile_dataset <- mobile_dataset[!is.na(as.numeric(mobile_dataset$RamSize_num)),]
mobile_dataset$RamSize_num<- as.numeric(mobile_dataset$RamSize_num)
mobile_dataset$RamSize_Unit <- as.numeric(mobile_dataset$RamSize_Unit)
#Step 7.3: generate the final column RamSize_inGB
mobile_dataset$RamSize_inGB <- mobile_dataset$RamSize_num *
↪  mobile_dataset$RamSize_Unit


#Step 8: Create a new column to determine if the phone is 5G or not
mobile_dataset$Is_5G <- ifelse(str_detect(mobile_dataset$Model_Only, "5G"), "Yes",
↪  "No")
#Step 9: only keep the columns we need
column_names <- c("Model", "Company", "Price", "Rating", "No_of_ratings",
↪  "TotalRevivews", "Model_Only", "Color", "Price_Level", "RamSize_inGB",
↪  "RomSize_inGB", "RamSize_Ori", "RomSize_Ori", "Is_5G" )
mobile_dataset <- mobile_dataset[column_names]
write.csv(mobile_dataset, file = './Cleaned_Mobile_Dataset.csv', row.names = F)
```

After cleaning and breaking down columns, the dataset now consists of 11 columns and 736 rows and no missing values. These 11 columns are:

- **Model**: categorical variables with sub-classes. These names include the color of the unit and its storage capacity. The latter being also listed as a separate column. - **Independent Variable**
- **Company**: categorical variable. Name of the phone's manufacturer. - **Independent Variable**
- **Price**: continuous variable. Units in Indian Rupees. - **Dependent Variable**
- **Rating**: continuous variable. Units in Indian Rupees. - **Independent Variable**
- **Number of ratings**: discrete variable: a simple count. - **Independent Variable**
- **Total reviews**: discrete variable: a simple count. - **Independent Variable**
- **Model_Only**: categorical variable: only contains the model information of a mobile phone. - **Independent Variable**
- **Color**: categorical variable: color of a mobile phone. - **Independent Variable**
- **Price_Level**: The price level of a mobile phone, with levels of "Low", "Medium", "High", "Very High". - **Independent Variable**
- **RamSize_inGB**: continuous variable. RAM specification of the phone in GB. - **Independent Variable**
- **RomSize_inGB**: continuous variable. Storage (non-volatile memory) capacity of the phone in GB. - **Independent Variable**

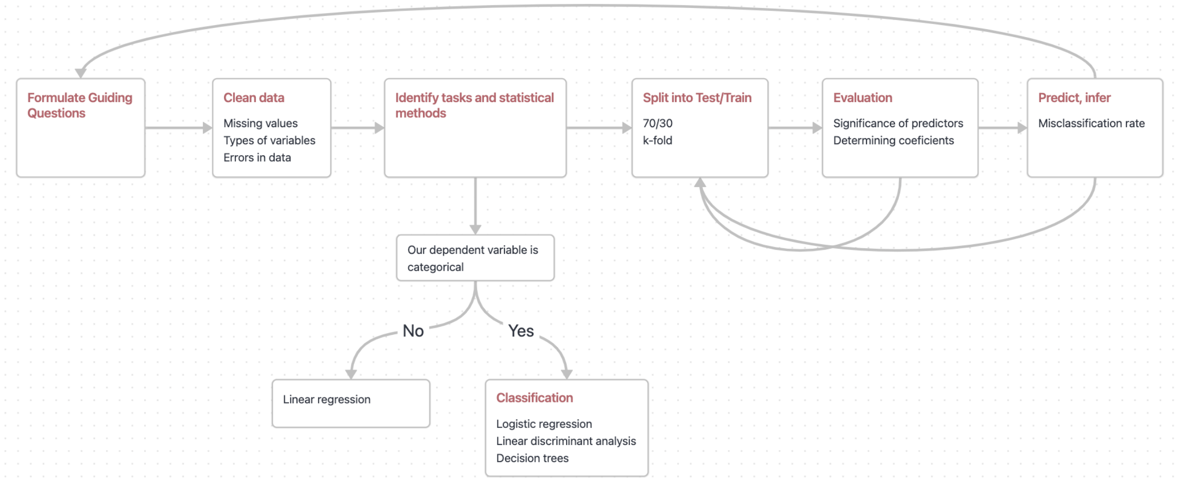
```
mobile_dataset <- as_tibble(read.csv("./Cleaned_Mobile_Dataset.csv"))
mobile_dataset %>% head(4)
```

```
## # A tibble: 4 x 14
##   Model        Company Price Rating No_of~1 Total~2 Model~3 Color Price~4 RamSi~5
##   <chr>        <chr>   <int>  <dbl>   <int>   <int> <chr>   <chr> <chr>     <dbl>
## 1 Infinix HO~ Infinix  8199    4.3     505      52 HOT 20~ Luna~ Medium        4
## 2 MOTOROLA e~ MOTORO~  7999    4.1   56085    5600 e40     Carb~ Medium        4
## 3 MOTOROLA e~ MOTORO~  7999    4.1   56085    5600 e40     Pink~ Medium        4
## 4 POCO C31 (~ POCO     7499    4.3  183688   11185 C31     Shad~ Medium        4
## # ... with 4 more variables: RomSize_inGB <dbl>, RamSize_Ori <chr>,
## #   RomSize_Ori <chr>, Is_5G <chr>, and abbreviated variable names
## #   1: No_of_ratings, 2: TotalReviwes, 3: Model_Only, 4: Price_Level,
## #   5: RamSize_inGB
```

The dataset and detailed analysis can be found at this repository.

Scope of Analysis

Our team is finalizing what the full analysis of the dataset will look like, but a preliminary template and breakdown of work by team member has been included below. The different colors represent which components of the project different team members would take on. It is anticipated that all members will assist in the finalization of the report.

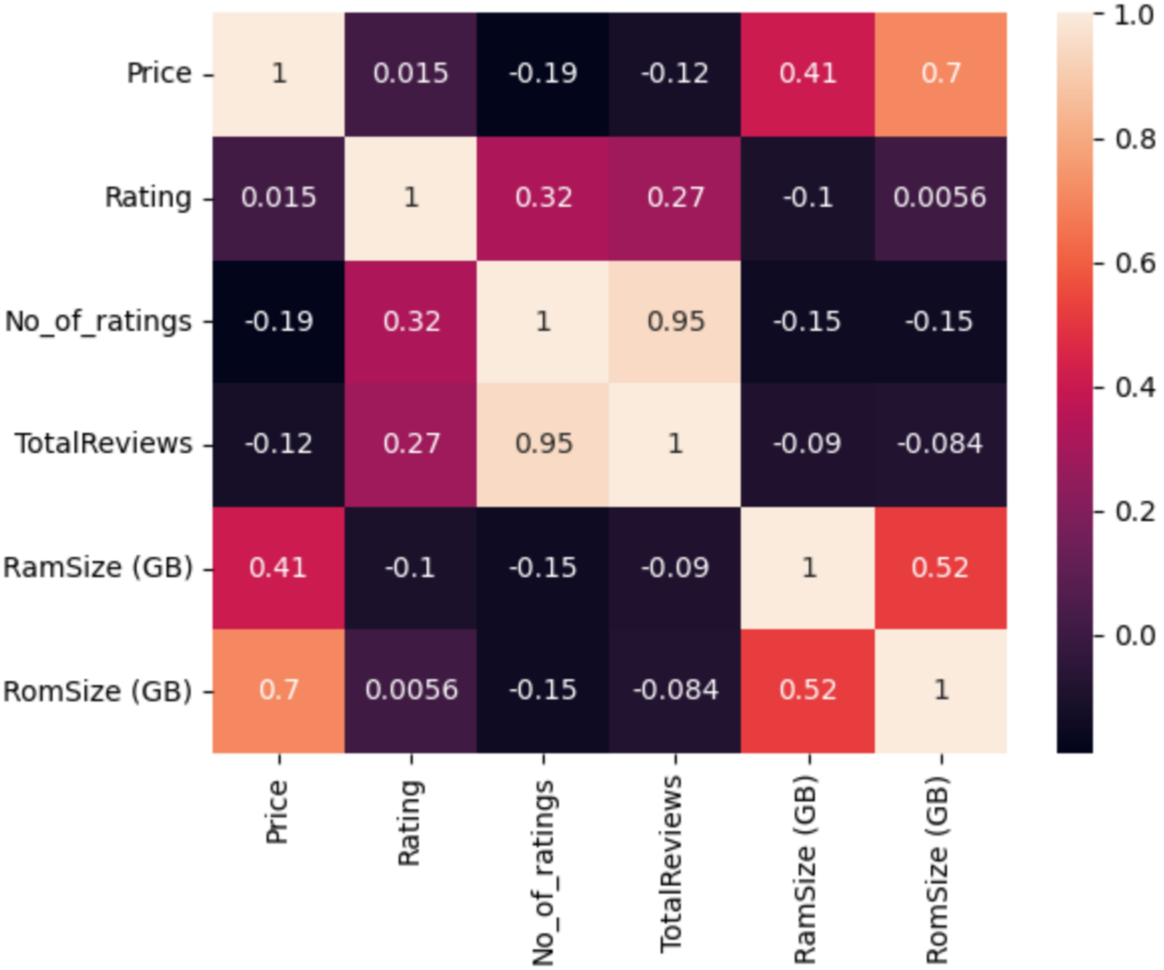


Dimensionality reduction

We should emphasize that this part of the analysis is not particularly useful in a dataset with less than 10 predictors like ours. It was still carried for its value as a learning exercise. This point was brought up and addressed during our presentation. With that in mind let's go through the motions.

Principal component analysis

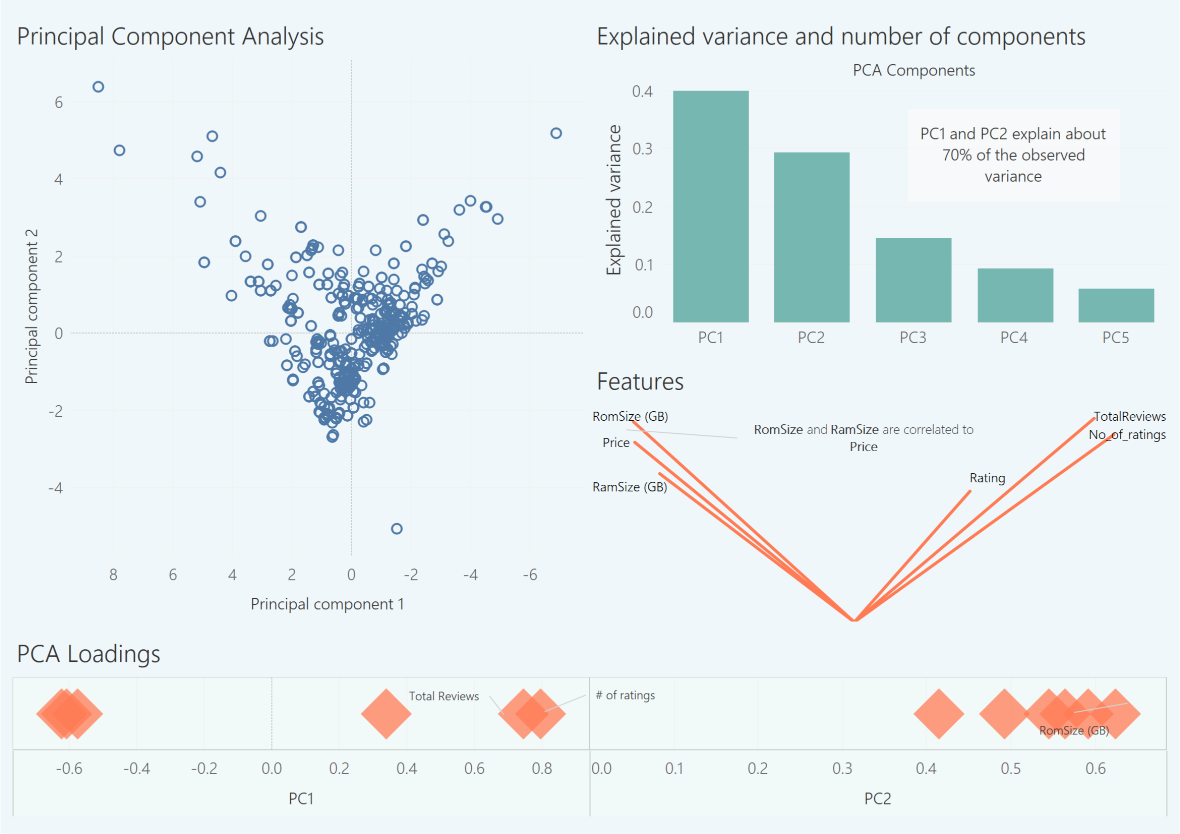
PCA is a tool that projects a collection of vectors into a a new vector space of fewer orthogonal dimensions. This projection can be simpler to fit using a linear model but much harder to interpret: the new predictors are going to be linear combinations of the original predictors and may not have a direct interpretation. To carry out a PCA one must have correlation among features,



PCA is sensitive to outliers and scale. This means that if our numerical predictors have different units or span different orders of magnitude we need to re-scale it. For every predictor, this is accomplished by simply subtracting the mean and dividing by its standard deviation. In python, the library *scikit learn* support different types of scaling out of the box. It can also perform PCA,

```
from sklearn.decomposition import PCA
pca = PCA(n_components = 2)
principal_components = pca.fit_transform(df_scaled)
```

The following figure summarizes the results,



We can see that about 70% of the variance can be explained by projecting onto a 2D space. To use this in a prediction scenario we'd have to setup a pipeline that follows this workflow,

1) Rescale
2) Map from the original predictor/response space to the new PC space using a a linear transformation
3) Apply the model (predict)
4) Rescale result
5) Map from PC to original predictor/response space

This is less of a hassle when dealing with many predictors and the gains from dimensionality reduction are tangible.

References

2020 review. "High-End Mobile Phones Price Have Soared 490% in 20 Years | This Is Money." This Is Money, This Is Money, 23 July 2020, https://www.thisismoney.co.uk/money/bills/article-8548235/High-end-mobile-phones-price-soared-490-20-years.html.

MobilePhone's dataset. "MobilePhone's Dataset | Kaggle." Kaggle: Your Machine Learning and Data Science Community, Kaggle, 20 Dec. 2022, https://www.kaggle.com/datasets/sudhanshuy17/mobilephone.