OVERVIEW OF SMARTPHONES CHARACTERISTICS ON THE TECHNODOM.KZ WEBSITE

Prepared by: Daniyal Yerkesh, Meyerbekova Merey

**The project's main goal is to determine the most optimal and cost-effective types of smartphones, according to their characteristics.**

STEPS TO ACHIEVE OUR GOAL:

1. **Web scraping a page of smartphones, on the technodom website:**

1) Importing necessary libraries, such as requests, BeautifulSoup, and pandas

2) Define our main url and list of our future product links, to gather links for each smartphone

3) Identifying the parent tag and determining where links are located

<div class=”category-page-list\_\_list”> <div>

4) Writing a loop to collect these links, since there were 3 links for each product, getting rid of these links:



5) Counting the number of our links. Detect where name, price, and reviews are laid out. Inspecting the HTML page and finding out where they are, then applying it in the code.



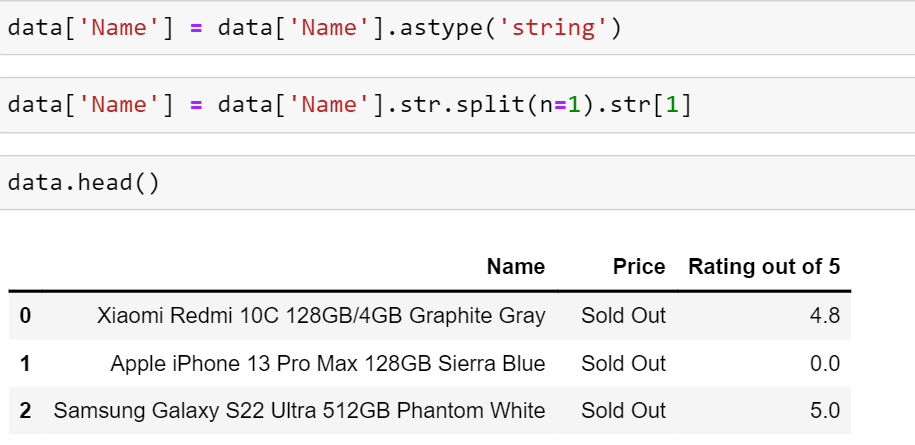
6) Final scraped data by dataframe:

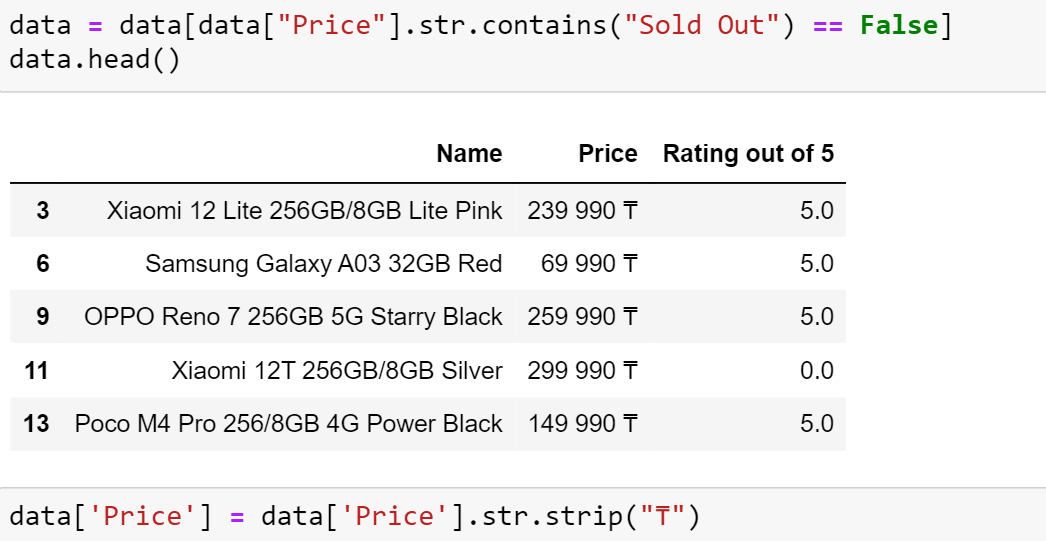
data = pd.DataFrame(itemlist)

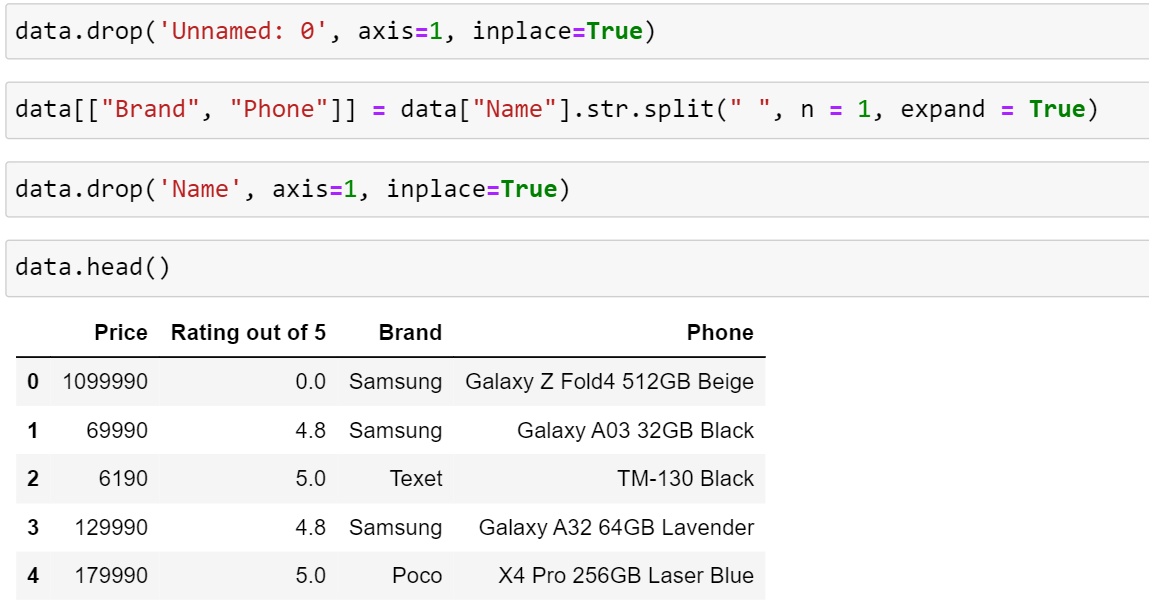
data

**2. Data cleaning :**

1. Changing name column: change the type to a string so that it can be easily used in the analysis, and also remove the first word (Смартфон)

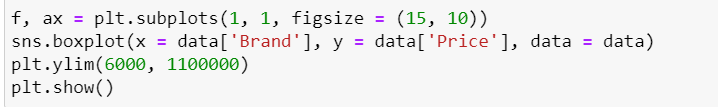


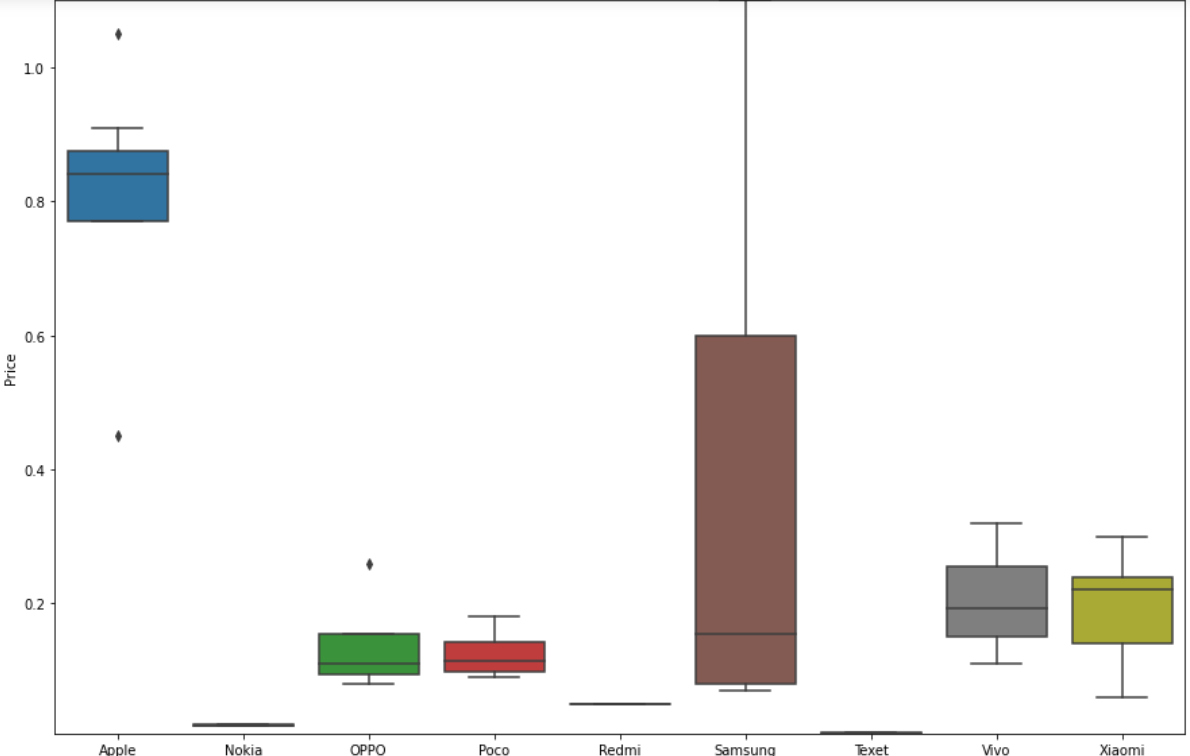
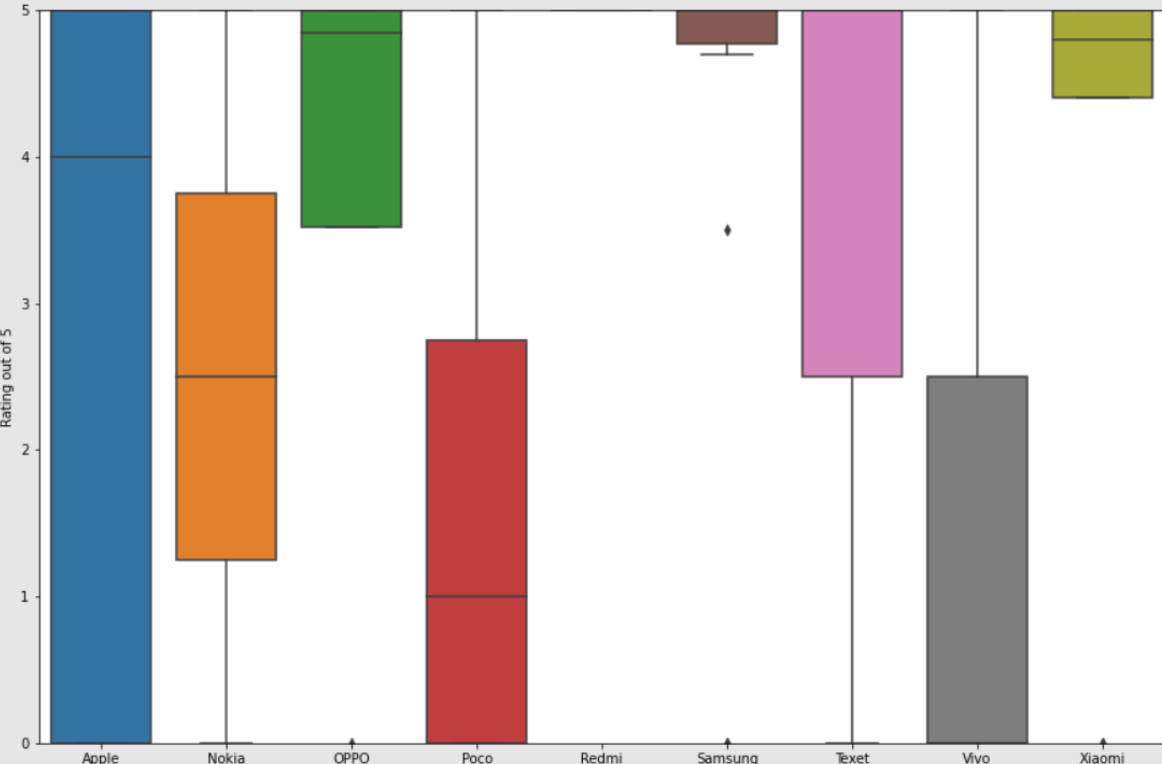
1. Changing price column: deleting “Sold Out” items, removing “₸” character 
2. Changing the type of “Rating out of 5” column to float
3. Creating “technodom.csv”: data.to\_csv("technodom.csv", encoding="utf-8")
4. Reading csv file, deleting an unnecessary column, creating new column - brand (to use it as a categories later)

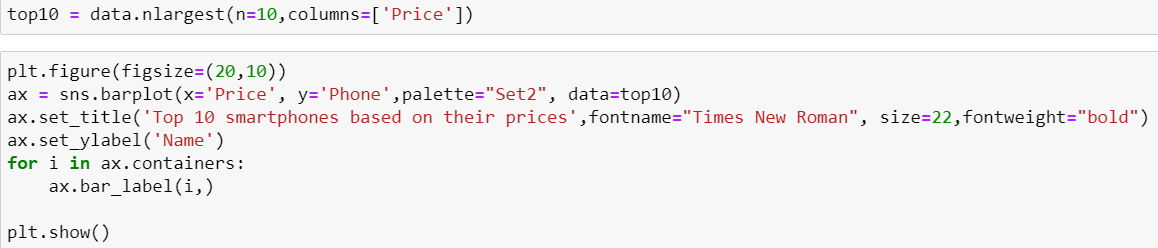


**3. Data Visualization for data analysis:**

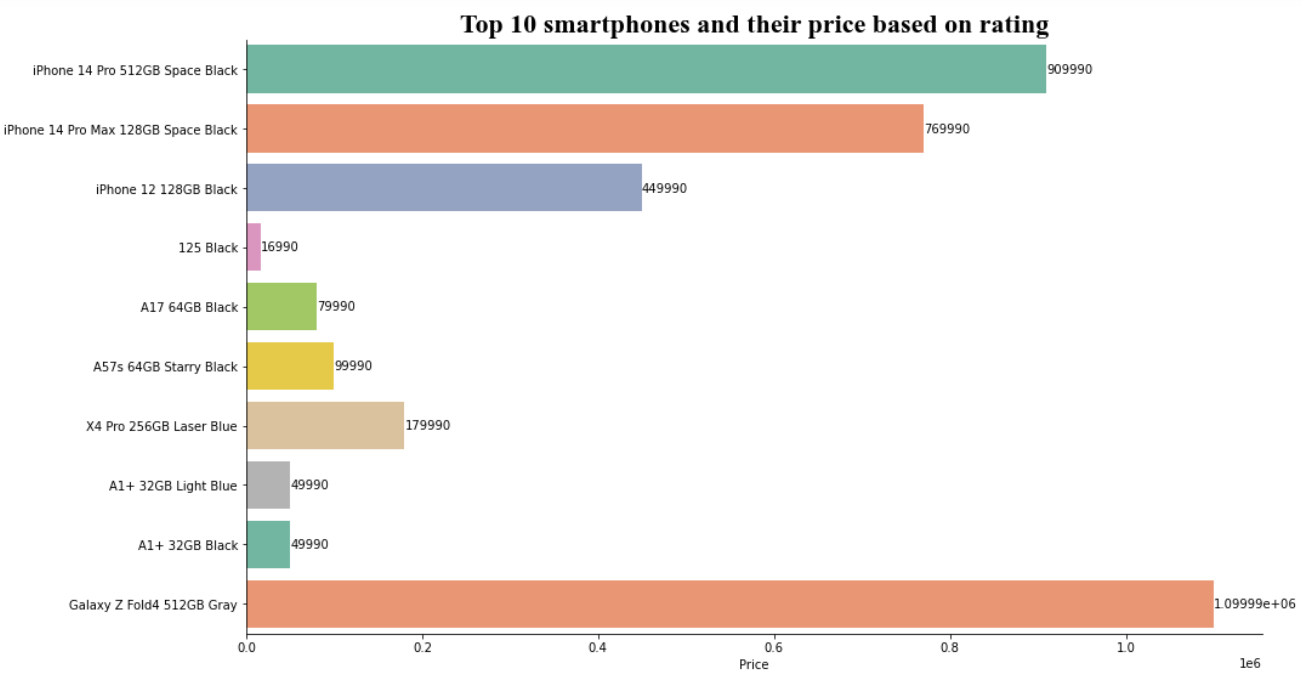
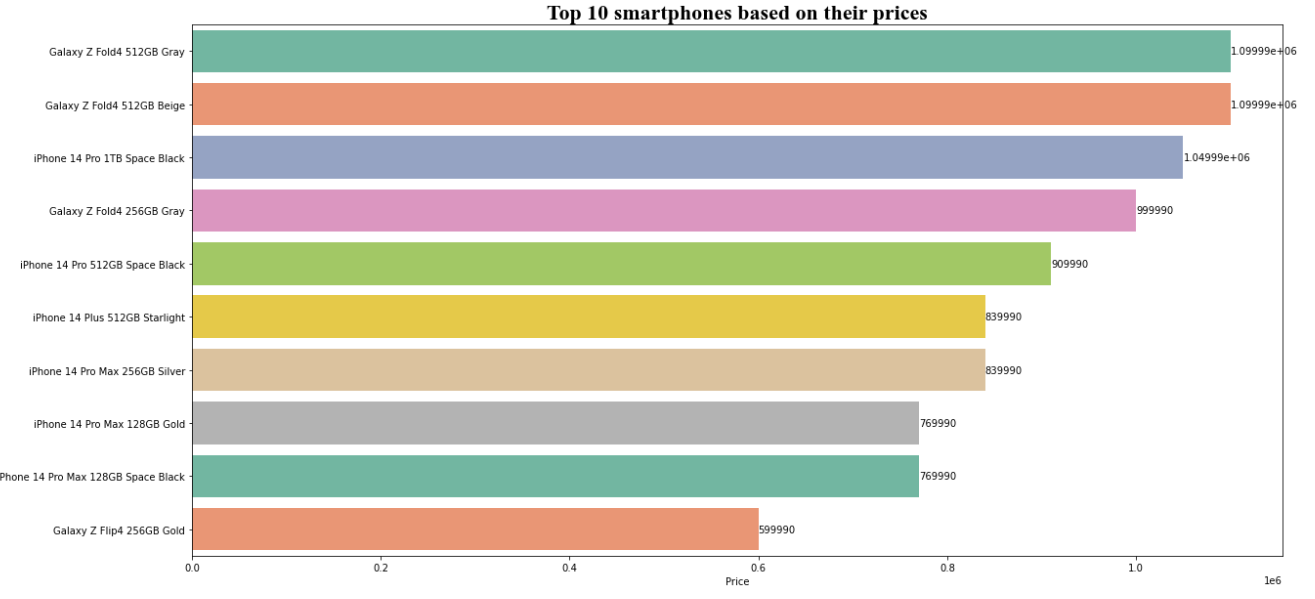
Determine which smartphone brand is better in terms of rating and price, looking at the charts.

1. Importing libraries: seaborn, matplotlib.pyplot, numpy
2. Determine outliers by price and rating by building boxplots, immediately created boxplots of all brands:

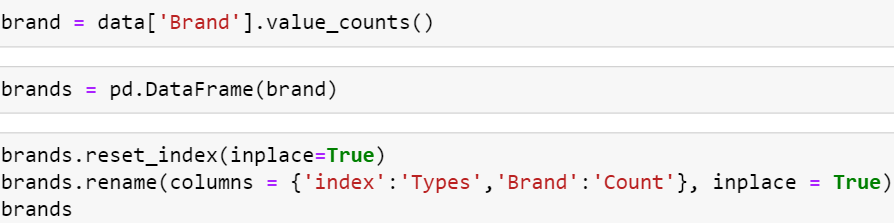


1. By barplots finding top-10 smartphones, based on rating and price:

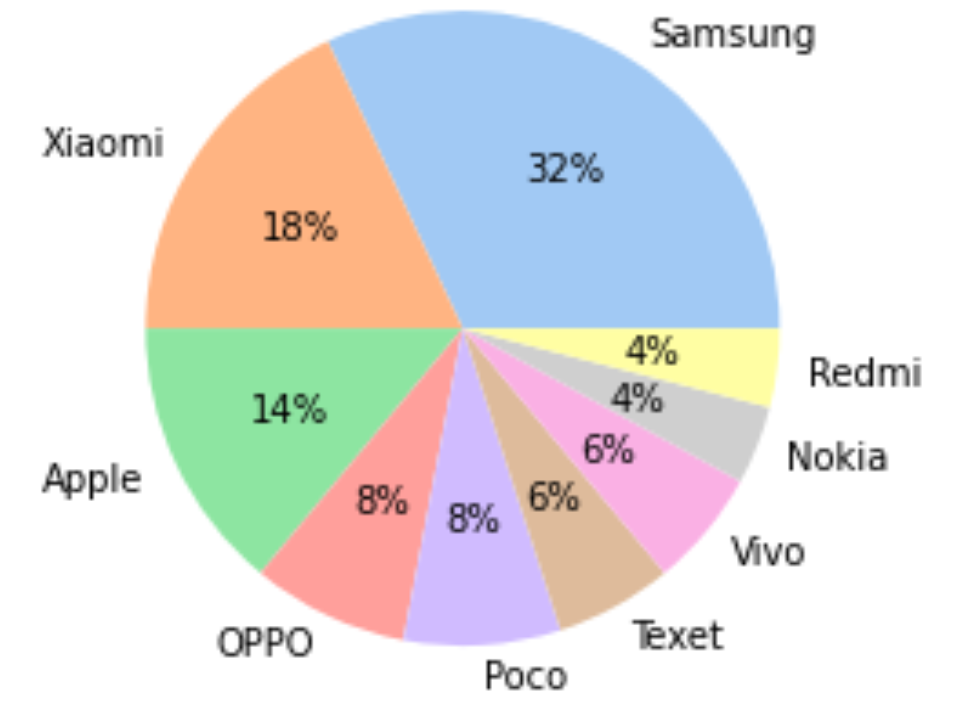
Visualizations:



1. Visualization of pie chart, to determine which brand is more popular: creating table of brand, by pandas dataframe, and then their count availability on website:





**4. Data Analytics using ML:**

1. To compare the price and the rating, we reduce the price:

data['Price'] = (data['Price'] // 100000)

1. Importing libraries: linear\_model, train\_test\_split from sklearn to divide our data for training and testing data
2. Dividing our data for training and testing, and then visualizing it with scatterplot:

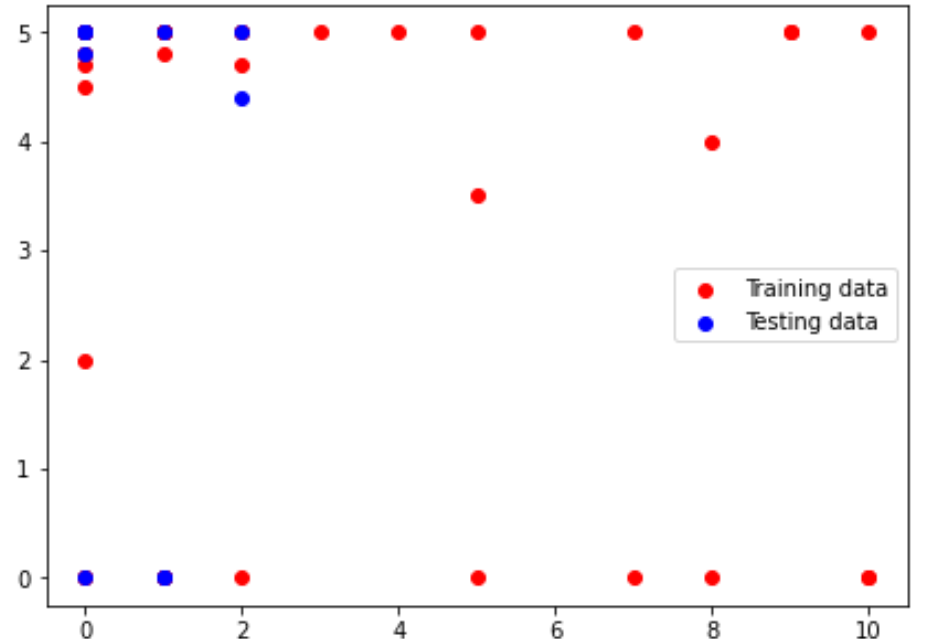
training\_data, testing\_data = train\_test\_split(data, test\_size=0.2, random\_state=25)

X\_train = pd.DataFrame(training\_data['Price'])

y\_train = pd.DataFrame(training\_data['Rating out of 5'])

X\_test = pd.DataFrame(testing\_data['Price'])

y\_test = pd.DataFrame(testing\_data['Rating out of 5'])



1. Building a linear regression model, prediction by X\_test values using scatter:

X1 = data['Price'].values.reshape(-1,1)

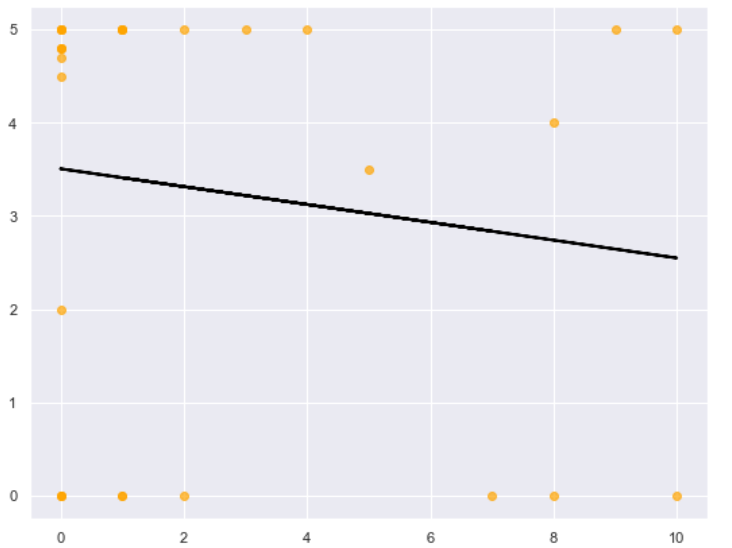
y1 = data['Rating out of 5'].values.reshape(-1,1)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X1, y1, test\_size=0.4, random\_state=0)

regression = lm.LinearRegression()

regression.fit(X\_train, y\_train)

y\_pred = regression.predict(X\_test)



**5. SVM classifier:**

1. Importing libraries: SVC, accuracy\_score from sklearn
2. Spliting our data, to obtain categories meaning smartphone brands:

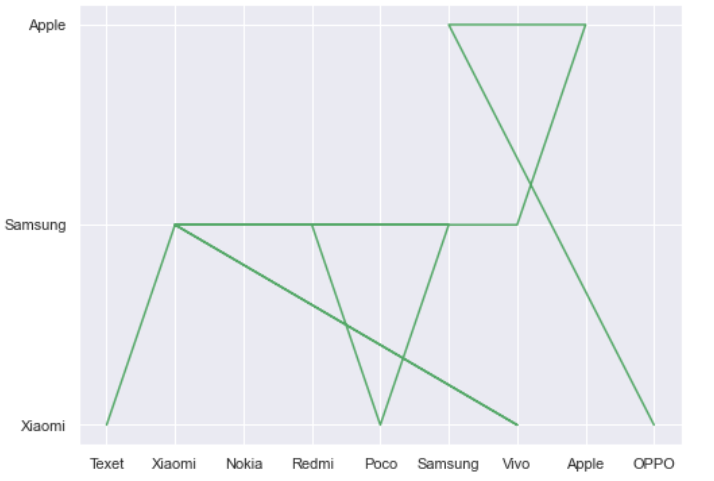
x=data[['Price', 'Rating out of 5']]

y=data['Brand']

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(x, y, test\_size=0.3, random\_state=25)

np.unique(Y\_train)

Output: array(['Apple', 'Nokia', 'OPPO', 'Poco', 'Redmi', 'Samsung', 'Texet',

'Vivo', 'Xiaomi'], dtype=object)

1. Realising a SVM classifier:

clf = SVC()

clf.fit(X\_train, Y\_train)

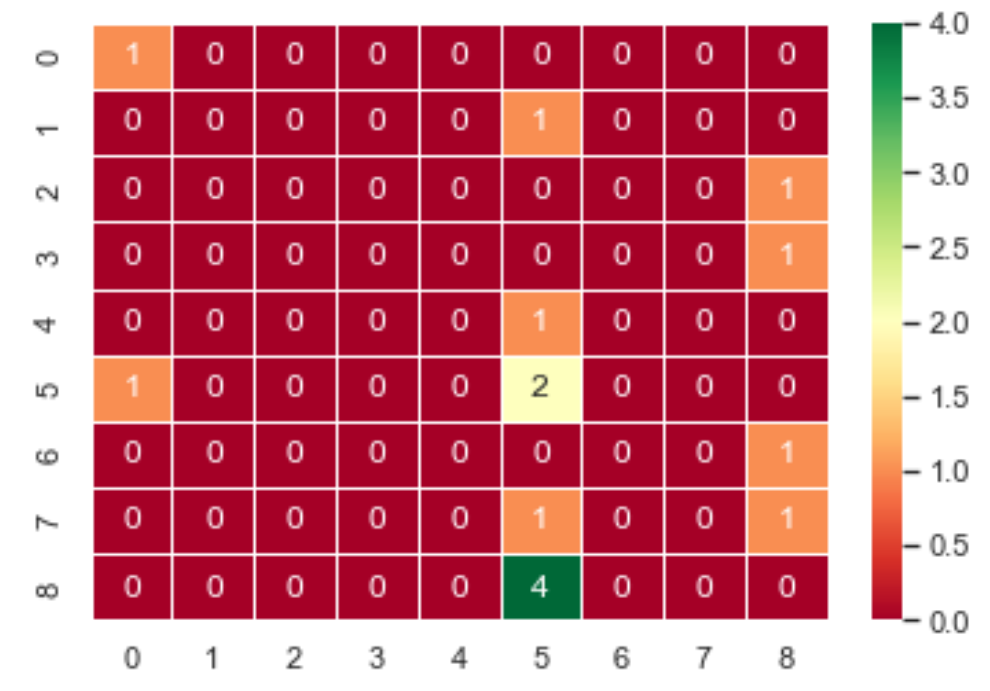
plt.figure(figsize=(8,6))

predictions=clf.predict(X\_test)

plt.plot(Y\_test, predictions, c = 'g')

print('Accuracy of training: ',accuracy\_score

(Y\_train,clf.predict(X\_train)))

1. Defining the performance of a SVM classification algorithm by confusion matrix, visualising it with heatmap.

from sklearn.metrics import confusion\_matrix

cm = confusion\_matrix(Y\_test, predictions)

sns.heatmap(cm,annot=True, linewidths=1,

cmap="RdYlGn")

plt.show()

We wanted to show how the brand of a phone is predicted by rating and price, for example: how to predict a high rating and price for choosing a phone. But since data having a small context and not a lot of data samples accuracy is 51%.

**Conclusion**: Taking data from the site of the technodom, we analyzed what types of smartphones to take. The Samsung brand has some of the best smartphones, with subplots ranking closer to 5, and prices have a wide range of choices. But also if you take a phone with a high rating and a low price, we advise you to take the Xiaomi brand. According to the top 10 phones that have a relatively good rating and a low price, for example phones: OPPO A17 64GB Black (79990 tenge and 5/5) and Poco X4 Pro 256GB Laser Blue (179990 in 5/5). If we look at the linear regression, we see that the price and rating variables do not really depend on each other and are close to negative dependence.