**Group Name: Py Scrapers**

**Group Members:**

|  |  |  |
| --- | --- | --- |
| **First name** | **Last Name** | **Student number** |
| **Akash Reddy** | **Burri** | **C0910253** |
| **Akshitha** | **Ralla** | **C0912936** |
| **Jagadeesh Krishna** | **Mattapalli** | **C0912884** |
| **Kameswara** | **Sai Pranit** | **C0913039** |
| **Sathwick** | **Sadanala** | **C0903053** |
|  |  |  |

**Submission date: 17 March 2024**

Contents

[Abstract 3](#_gjdgxs)

[Introduction 3](#_30j0zll)

[Methods 3](#_1fob9te)

Data scraping…………………………………………………………………………………………………………………………………………..4

Data wrangling…………………………………………………………………………………………………………………………………………5

Data visualization…………………………………………………………………………………………………………………………………….6

pandas profiling………………………………………………………………………………………………………………………………………8

Encoding methods…………………………………………………………………………………………………………………………………..9

outlier identification………………………………………………………………………………………………………………………………..9

Addressing outliers…………………………..…………………………………………………………………………………………………...11

unsupervised learning methods…………………………………………………………………………………………………………….13

**Abstract:**

The fast paced growth of the industry of smartphones is a subject of frequent examination, one in which the status of consumer trends, market trends, and rivalry environments should be regarded. The primary intended purpose of our project is to interrogate and extract smartphone data from a globally renowned and popular internet marketplace known as eBay. We get the information on mobile display listings (title, conditions, and prices) via Python (requests and beautifulsoup) using web scraping. Through the Pandas library, we later structure and investigate the data to fathom what are the price fluctuations and market trends. In order to assist shoppers, retailers, and the leaders in this industry to be well informed, we intend to analyze the market and provide useful content to help them have a better judgment about smartphones in this ever-growing market. This study has shown how the web scraping and data analysis methods, characterized by effectiveness, precision and statistical rigorousness, may serve for information extraction.

**Introduction**

The primary objective of this project is to extract and analyze smartphone data from eBay, a popular online marketplace. By scraping information such as titles, conditions, and prices of smartphones listed on the platform, we aim to gain insights into the current trends and pricing patterns in the smartphone market. This analysis can provide valuable information for consumers, retailers, and industry analysts, helping them make informed decisions in a rapidly evolving market.

**Importance of Analyzing Smartphone Data in the Current Market:**

The smartphone industry is characterized by rapid technological advancements, frequent product launches, and fluctuating prices. Analyzing smartphone data from a major marketplace like eBay can offer several benefits:

* **Consumer Insights:** Understanding the most sought-after features, brands, and price ranges can help consumers make better purchasing decisions.
* **Market Trends:** Identifying popular smartphone models and their pricing trends can provide insights into consumer preferences and market demand.
* **Competitive Analysis:** Retailers and manufacturers can assess their competitors' pricing strategies and market positioning.
* **Inventory Management:** By analyzing the condition and pricing of smartphones, sellers can optimize their inventory levels and pricing strategies.

**Tools and Libraries Used:**

The analysis is conducted using the following Python libraries:

* **Requests:** This library is used to send HTTP requests to the eBay website, enabling us to retrieve the HTML content of the search results pages.
* **BeautifulSoup:** A library for parsing HTML and XML documents, BeautifulSoup is utilized to extract relevant data from the HTML content obtained via Requests.
* **Pandas:** A powerful data manipulation and analysis library, Pandas is employed to organize the scraped data into a structured format, facilitating further analysis and visualization.

By leveraging these tools, we can efficiently scrape, process, and analyze large volumes of smartphone data from eBay, providing valuable insights into the current state of the smartphone market.

**Data Scrapping:**

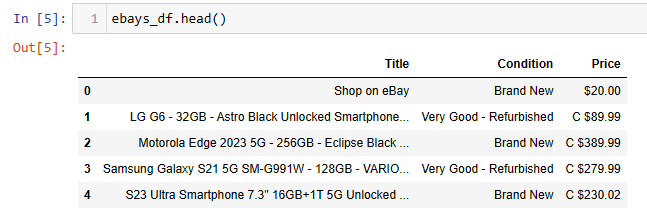
This Python script is designed to scrape information about smartphones listed on eBay. It utilizes the Requests library to send HTTP GET requests to eBay's search results pages and BeautifulSoup to parse the HTML content of the pages and extract relevant data such as titles, conditions, and prices of the listed smartphones. The scraped data is then organized into lists and stored in a Pandas DataFrame.

The scrape\_ebay\_info function takes two parameters: base\_url, which represents the base URL of the eBay search results page, and num\_pages, which specifies the number of pages to scrape. Inside the function, a loop iterates through each page of the search results, sending a GET request to each page's URL, and parsing its HTML content.

For each page, the script finds all the item containers using BeautifulSoup and extracts the title, condition, and price of each listed smartphone. It handles cases where the information may not be available by assigning default values such as "Title not found", "Condition not found", or "Price not found".

The extracted data is then appended to separate lists (titles, conditions, and prices). Once all pages are scraped, these lists are used to create a Pandas DataFrame (df). Finally, the data frame is saved to a CSV file named 'smartphones\_in\_ebay.csv' using the to\_csv method.

Overall, this script provides a flexible and automated approach to gather data from eBay's marketplace, enabling further analysis and insights into the smartphone market trends and pricing patterns.



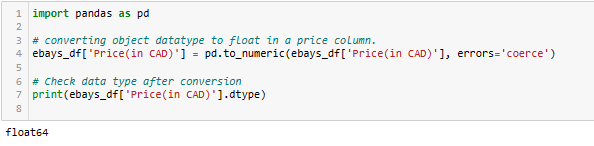
**Data Wrangling:**

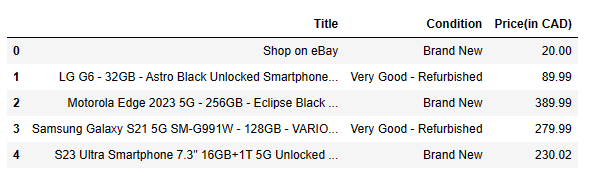
This step includes data cleaning, validating, preprocessing and exploring

The Price column was converted from string values into floating point numbers. Also discarding any special characters and non-numeric characters from the prices to ensure the column is purely numerical. This allows us to accurately analyze and work with the pricing data, facilitating easier comparisons and calculations.





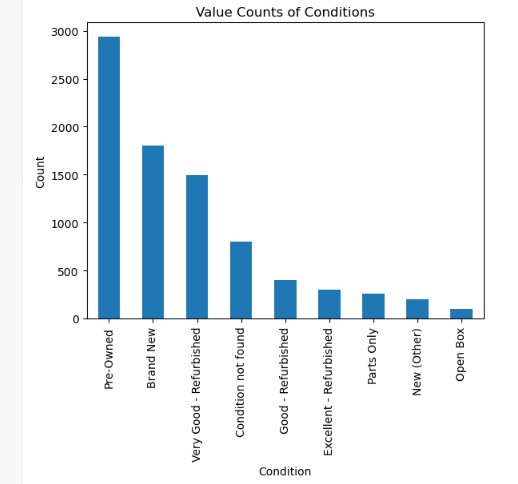




After performing Datavalidation process finally Price Column converted into Floating datatype.

**Data Visualization:**

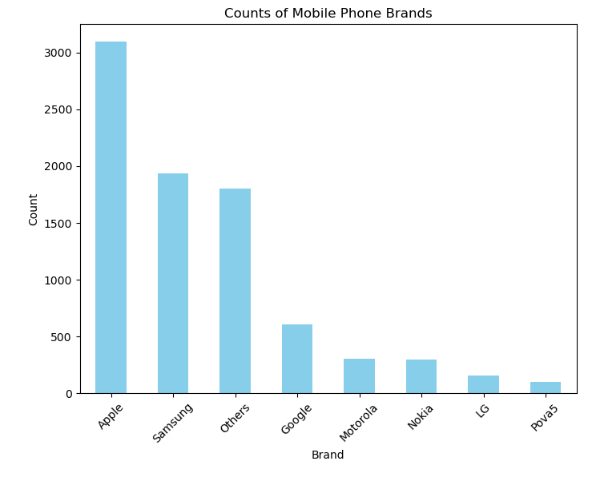
imported matplotlib libraries required for visualization and plotted a bar plot against conditions and value counts of condition



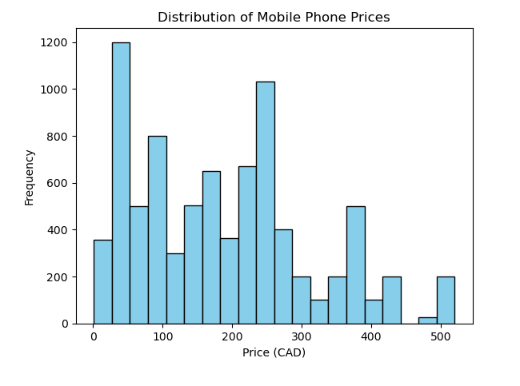
Overall, the prominence of preowned items on eBay reflects the platform's role as a marketplace that caters to both sellers looking to offload their used items and buyers seeking affordable alternatives to new products.

As there are a lot of mobile phones with different names in their title column, some of the popular brands were considered into a list called ‘brands’ and the other names were grouped into ‘others’ category

Below is a barplot with brands on its x label and count of brands on y label



Looking at the bar graph, it's evident that most folks in North America have a strong inclination toward buying Apple and Samsung gadgets. The preference is shining through in the graph, where the number of listings for Apple and Samsung products stands out prominently compared to other brands. This tells that the majority of consumers in North America lean towards choosing mobile devices made by Apple and Samsung.



The graph titled "Distribution of Mobile Phone Prices" is a histogram that illustrates the frequency of mobile phones across different price ranges in Canadian dollars (CAD). Based on the histogram:

* The most frequent price range for mobile phones on eBay falls between CAD 0 to 100, suggesting that lower-priced phones are more commonly listed or that this price range is more popular among eBay listings.
* There is a significant number of phones in the CAD 200 to 300 price range, indicating a healthy market for mid-range priced phones.
* The histogram shows a roughly multimodal distribution, with noticeable peaks in the lower and mid-range price categories. This might suggest that both budget and moderately priced phones have substantial representation on the platform.
* Higher-priced phones (above CAD 400) are less frequent, implying either a lower supply or demand for premium or high-end mobile phones in the eBay marketplace.
* The frequency steadily decreases as the price increases, which is typical in consumer electronics markets where premium devices are less commonly purchased compared to budget and mid-range devices.

**Pandas profiling:**

**Encoding Methods**

**1) Label Encoding:**

Label encoding is applied using the factorize function from pandas. It assigns a unique numerical label to each unique category in the 'Condition' column.

The label-encoded values are stored in a new column named 'Condition\_LabelEncoded' in the DataFrame ebays\_df.

**2)One-Hot Encoding:**

One-hot encoding is applied using the get\_dummies function from pandas. It creates binary dummy variables for each unique category in the 'Condition' column.

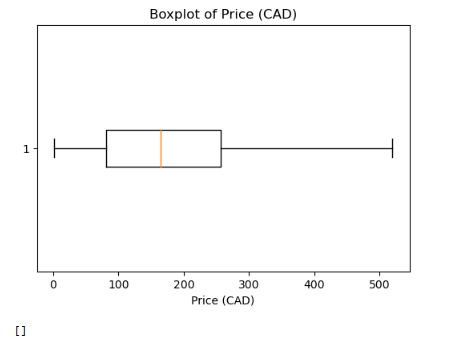
The prefix 'Condition' is added to the column names of the dummy variables to indicate their association with the 'Condition' column.

The resulting dummy variables are concatenated with the original DataFrame ebays\_df along the columns axis (axis=1), effectively expanding the DataFrame with one-hot encoded columns.

By performing both label encoding and one-hot encoding, the code transforms the categorical 'Condition' column into numerical representations suitable for machine learning algorithms. Label encoding provides a single numerical representation for each category, while one-hot encoding creates binary variables to represent each category separately. These encoded features can be used as input for machine learning models to analyze or predict smartphone pricing trends based on their condition.

**Outlier Identification**

To identify outliers within the 'Price(in CAD)' column of the DataFrame ebays\_df. To visualize the distribution of prices, we employ a boxplot, a graphical representation that provides a summary of the data's central tendency and dispersion. The boxplot is created using matplotlib's plt.boxplot() function, which displays the quartiles, median, and potential outliers. Following the visualization, we compute the interquartile range (IQR) to establish the boundaries for identifying outliers. The IQR, the difference between the third quartile (Q3) and the first quartile (Q1), helps us gauge the spread of the data. Using 1.5 times the IQR rule, we determine the lower and upper bounds for outlier detection. Any data point falling below Q1 - 1.5 \* IQR or above Q3 + 1.5 \* IQR is considered an outlier. Subsequently, we filter the outliers from the original DataFrame and store them in a new DataFrame named outliers. Finally, we print out the prices of the identified outliers for further examination. This step provides crucial insights into potentially anomalous data points that may require additional scrutiny or preprocessing before further analysis.



The image presents a boxplot depicting the distribution of mobile phone prices in Canadian dollars (CAD).

"The boxplot provides a statistical summary of mobile phone prices on eBay, delineating the central tendency and variability of the dataset. Key observations from the boxplot are as follows:

* The **median** price, indicated by the orange line within the box, appears to be slightly above the 100 CAD mark, suggesting that half of the smartphone prices are above this value, while the other half falls below.
* The **interquartile range (IQR)**, represented by the box, is relatively narrow, which implies that the middle 50% of the prices are concentrated within a limited range around the median.
* The **whiskers**, which extend from the box, suggest that the prices have a **wide distribution**, with the lowest prices starting close to 0 CAD and the highest reaching up to approximately 500 CAD.
* There are no visible outliers, as there are no points that lie beyond the whiskers or outside the expected range of variation.

**Addressing outliers**

**Quantile-based Flooring and Capping:**

Mean: The mean price after capping slightly decreases from 195.39 to 192.56 CAD. This indicates that extreme high values were reduced, leading to a decrease in the overall average price.

Standard Deviation: The standard deviation decreases noticeably from 131.05 to 123.12 CAD. This reduction indicates a decrease in the dispersion or spread of prices, as extreme values are limited.

Minimum and Maximum: The minimum price remains the same at 27.00 CAD, indicating that the lower bound was not changed. However, the maximum price decreases significantly from 519.99 to 430.30 CAD, demonstrating the effectiveness of capping in reducing extreme high values.

**Trimming:**

Mean: The mean price after trimming decreases more noticeably to 187.34 CAD compared to capping. This indicates a more significant reduction in the average price due to the removal of extreme values.

Standard Deviation: The standard deviation also decreases significantly from 131.05 to 123.12 CAD, similar to capping. However, trimming may result in a slightly narrower spread of prices compared to capping.

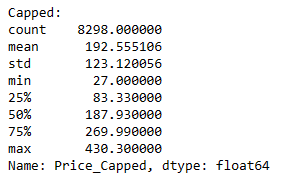
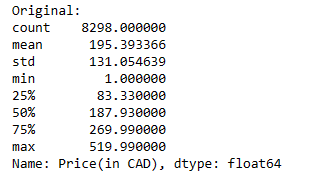
Minimum and Maximum: Both the minimum and maximum prices remain the same as in the capped dataset, indicating that trimming removes extreme values from both ends of the price distribution.

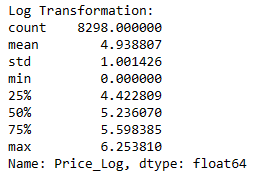
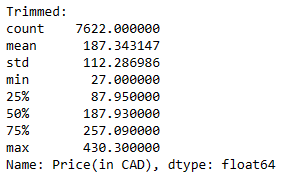
**Log Transformation:**

Mean: The mean log-transformed price is substantially different from the original mean, indicating a significant change in the average price representation. Log transformation compresses large values and expands small values, altering the distribution.

Standard Deviation: The standard deviation of log-transformed prices is relatively small (1.00), indicating less variability compared to the original data. Log transformation often reduces the impact of extreme values on the variability of the data.

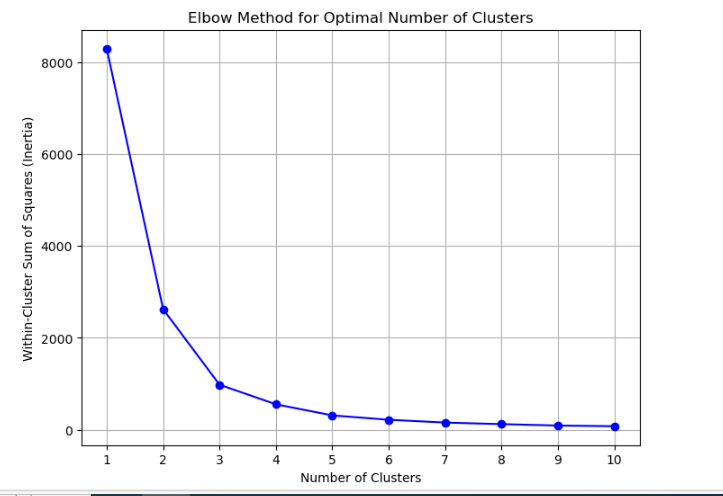
Minimum and Maximum: The minimum log-transformed price is 0.00, indicating that some original prices were close to or equal to zero. The maximum log-transformed price (6.25) represents the upper limit of the transformed data, which may have compressed extreme high values.





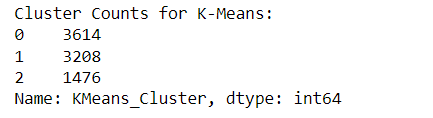
**Unsupervised Learning Methods:**

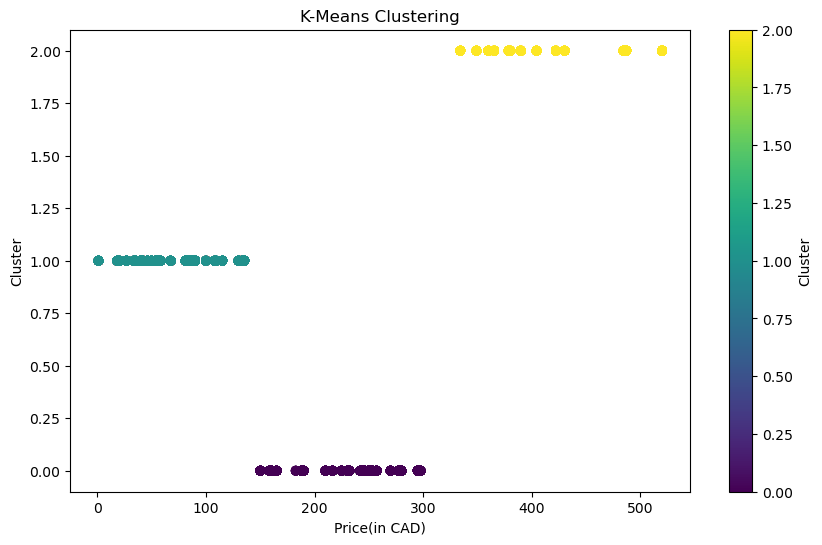
Elbow Method to determine the optimal number of clusters for KMeans clustering. First, numerical columns, specifically 'Price(in CAD)', are selected for clustering. Then, the data is standardized using StandardScaler to ensure that all features have a mean of 0 and a standard deviation of 1, which is a prerequisite for KMeans clustering. The inertia, which represents the within-cluster sum of squares, is calculated for different numbers of clusters ranging from 1 to 10. For each number of clusters, a KMeans model is fitted to the scaled data, and its inertia is recorded. The Elbow Method plots the number of clusters against the corresponding inertia. The "elbow" point, where the inertia starts to decrease at a slower rate, indicates the optimal number of clusters. In this plot, we aim to identify the number of clusters that provides a good balance between minimizing inertia (i.e., compactness of clusters) and avoiding excessive fragmentation of data into too many clusters.



**1) K means**

Firstly, the numerical column 'Price(in CAD)' is selected for clustering. Next, the data is standardized using StandardScaler to ensure that all features have a mean of 0 and a standard deviation of 1, which is a prerequisite for K-Means clustering. K-Means clustering is then applied with the specified number of clusters, in this case, 3 clusters. The 'KMeans\_Cluster' column is created in the eBay dataset to store the cluster assignments for each data point. Finally, the distribution of clusters is analyzed by counting the occurrences of each cluster using value\_counts(), providing insight into how the data points are distributed across the clusters. This analysis helps in understanding the segmentation of the dataset based on the price of smartphones.





**2) Gaussian Mixture Model**

The numerical column 'Price(in CAD)' is selected for clustering. Next, the data is standardized using StandardScaler to ensure that all features have a mean of 0 and a standard deviation of 1, which is a prerequisite for GMM clustering. GMM clustering is then applied with the specified number of components, in this case, 3 clusters. The 'GMM\_Cluster' column is created in the eBay dataset to store the cluster assignments for each data point. Finally, the distribution of clusters is analyzed by counting the occurrences of each cluster using value\_counts(), providing insight into how the data points are distributed across the clusters. This analysis helps in understanding the segmentation of the dataset based on the price of smartphones using Gaussian Mixture Models.

