Mobile Mentor: Your Mobile Phone Comparison Tool Documentation

---

3,793,1.00 - Skills: Programming - Introduction Level

Submitted by a_h

21.12.2023

Content

About the application

This web program will compare any mobile phone from a dataset. With a set size of 8775 entries the user can have a look at many characteristics and details of the devices and will be able to ask for an automatic comparison. The application will analyse any given selection of devices and display their differences and strengths and decides over the best device.

Language and Resources

This is an application using python and streamlit. It is deployed to the streamlit cloud and can be accessed with the following url: https://mobilementor.streamlit.app/

If you are interested to run it locally on your machine, please install streamlit (pip install streamlit), download the csv file and run the program with your anaconda cmd (/path/ streamlit run mobilementor.py)

The application imports four different modules and libraries. These are the following:

- import streamlit as st
- import pandas as pd
- import altair as alt
- import plotly.express as px

Streamlit is used to display the content (such as: dataframes, text and charts) as a web-based application. It uses its own commands such as:

- st.write()
- on1 = st.toggle()
- see more: https://docs.streamlit.io/library/get-started

The dataset used for this application is provided with the submission. Below is the link to the original source and the license to use it.

- **Source:**https://www.kaggle.com/datasets/deepann/4000-laptops-data-from-gadgets360?select=mobiles.csv
- **License:** CC BY-NC-SA 4.0 DEED, Attribution-NonCommercial-ShareAlike 4.0 International
    - https://creativecommons.org/licenses/by-nc-sa/4.0/

Structure

The app splits into different parts which are separated through titles and toggles:

- Full dataframe (see first toggle)
- Comparison of the mobile devices (Main Part, see title)
- Further analysis (see header and toggle)

Functions

The code has four functions:

- def modify_df()
- def display_images()
- def create_charts()
- def calculate_points()

How the program works

Firstly, the application will load the dataset. This original dataset will then be modified because it is not consistent and cannot be directly used to make the comparisons. Therefore, it will be adjusted to fit this project's needs. The new dataset contains "Price in CHF" instead of "Price in India" and in addition has the columns "Overall Rating", "Amount of Pixels", "Main Camera (megapixel)" and "Points".

- If there is any NaN in the "Products Name" column, this row will be dropped
- The price in Swiss francs is calculated by converting Indian currency to Swiss: Remove currency character, multiply it by * 0.011
- The overall rating is calculated by ([amount of ratings of one star] * 1 + [amount of ratings of two stars] * 2 +…) / amount of ratings
- The Main Camera is defined with the "Rear Camera" column: The attribute gets split up at the first "-" to receive the first number, which is the main camera. Some values do not contain numbers, they are set to 0
- The amount of pixels is calculated by removing any dots or commas from the "Resolution" column, separating the string at the "x" and multiplying these two numbers

- The points column will be used in the calculations later and is set to 0 in the beginning.
- If there is any cell without a value, it receives a value of 0

To be able to have a look at the dataset, an additional section was added which can be found under the first toggle. Using the search bar, the user can look for any specific product name and the list below will be filtered for all the items that contain the searched string. To have a better overview of the entries, the user can select the columns that will be display with a multiselection bar.

For the main part of the application the user can select as many mobile phones as they want. To do so they use the multiselection bar and enter or select their devices. The program will create a simple table to display images of the devices on the left and their name on the right. The images are being taken out of the column "Picture URL". To not overload the user's device the images are displayed separately in the toggle below the user's selection.

After selecting the first device, the application will start the personalised comparison. The comparison takes five categories into consideration:

- Price
- Overall Rating
- Main Camera
- Amount of Pixels
- Battery Capacity (mAh)

The program automatically determines a winner device using its ranking system. The devices receive points for their ranking in each category. The amount of points depends further on the amount of devices that are being compared and whether the devices are sharing a rank or not.

Example: Let's say we compare four devices: A has the best camera, B the second best and C and D share the last rank. The ranking system would look like this:

1. Device A – 64 MP – 4 Points
2. Device B – 48 MP – 3 Points
3. Device C – 12 MP – 1 Points
4. Device D – 12 MP – 1 Points

Calculation steps: First the program sorts the values in ascending or descending order, giving the devices a rank. Generally, the rule is the higher a value the better, but this rule is flipped when comparing the devices' price. If e. g. two devices share a value, then they will also share their rank and receive the higher one out of the two possible ranks. Then the program will take the number of devices in the comparison and subtract the rank of one device, add one back to it and add the calculated points to said device. Let's calculated the points for Device A in the example above: 4 [Devices] – 1 [Rank] + 1 = 4 [Points]. The points are calculated for every single comparison and are summed up in the end to determine the overall winner. This result is displayed in a bar chart, sorting the devices by performance.

Afterwards the user can have an insight into the individual comparisons. Every category has its own bar chart to visualize the differences of the devices.

In the very end the user can have a look at the devices' strengths. The strengths are displayed in a polar/spider chart and show their performances in the categories. The values for the spider chart are the ones given by the ranking system.

For the last section of the program the user can inspect different brands and how the decide to produce and sell their devices. This shows some important product design decisions of the manufacturer. The user first selects a brand from the dataset and then selects a category. A bar chart will be created showing a distribution on how often a value is chosen by the manufacturer when designing a product. Further e. g. the user can be interested in the ratings of Apple devices. After selecting "Apple" as the brand and the category "Rating", the user receives information about the rating distribution and values such as min, max and average rating.

Have fun using my app!