Hochschule Rhein-Waal
Rhine-Waal University of Applied Sciences
Faculty of Communication and Environment

Degree Program Information Engineering and Computer Science, M. Sc.

Prof. Dr. Timo Kahl

Smartphone Recommendation System

Term Paper

SS 2019

Module “Applied Research Project A”

by

Anup Katuwal	-	26557
Bhuwan Acharya	-	26129
Bikash Paudel	-	26192
Md. Monsur Ali	-	24547
Musifique Ahmed Isha	-	26532
Sudarshan Awasthi	-	26558

Acknowledgement

The team assigned for Applied Research Project "A" would like to express sincere gratitude to Professor Timo Kahl, the project supervisor, for his guidance, enthusiastic encouragement and beneficial critiques for the applied research project. The team would also like to thank him for his regular advice in making progress on schedule and conducting progress check meetings. Heartful thanks to all those who helped the team in the survey distribution because of which survey was performed successfully.

Abstract

The smartphone recommendation system recommends smartphones based on user's age, gender, annual income, willing amount, profession, predefined possible reasons for purchase, and features preferences. The questionnaire for the survey was created using the google forms, and the study was conducted to get the data for the recommendation system. Data visualization and analysis were achieved using Tableau. The data was complicated, and to handle such complicated data for learning the underlying trend, a deep learning mechanism was needed. That was achieved using Artificial Neural Networks. The information was trained using the Artificial Neural Networks using Rectified Linear Unit and SoftMax function in the hidden layer and the output layer, respectively. The recommendation system outputs the probabilities of the possible outputs and recommends the highest probability one. Based on that significant recommendation, the system suggests similar featured alternatives. Afterwards, the website was built using HTML, JavaScript, CSS, and flask to make system interactive. A cloud-based Firebase Realtime database was created to store smartphone's specifications data. Finally, for the complete system, the website, the database, and the recommendation system were integrated.

Table of Contents

List of abbreviation

ANN	-	Artificial Neural Networks
HTML	-	Hyper Text Markup Language
CSS	-	Cascading Style Sheets
OS	-	Operating System
URL	-	Uniform Resource Locator
IT	-	Information Technology
JSON	-	JavaScript Object Notation
SDK	-	Software Development Kit
API	-	Application Programming Interface
SQL	-	Structured Query Language
ReLU	-	Rectified Linear Unit
UAR	-	Usability Aspects Report
BI	-	Business Intelligence

LIST OF FIGURES

1. Introduction

Most of the globe's population is well acquainted with the term "Smartphone". Regular phones are capable of phone calls and text messages. However, a smartphone is an integrated computer with an operating system, web browsing and capability of handling third party application for more features, including general features like calls and text messages. Nowadays, Smartphones are not just for communication. Instead, they have created their need in our lives, and used for daily life purposes such as navigation, shopping, banking, and many others.

According to Statista, a leading German statistics company shows that in 2018, approximately 2.7 billion people are using smartphones and numbers are increasing rapidly. Due to a wide variety of applications and usability, more than one-third of the population is already smartphone users. Statista also found that only in 2018, globally, around 1.56 billion smartphones were sold. Moreover, Statista also forecasted that almost half of the global population would have a smartphone by 2021 ("Cell Phone Sales Worldwide 2007-2018 | Statista").

These days, we can find a variety of smartphones with different specifications, and also similar features with different price ranges. GSM Arena is a website that shows you the smartphone details, including comparison features so that you could compare the smartphone before purchasing one. GSM Arena is the reference for this project. The database of GSM Arena contains a total of 114 distinct brands of smartphones. Among those 114 brands, around 80% of the smartphone runs on the Android operating system (OS). Approximately 15% of the total brands has the iOS operating system, and remaining have other operating systems such as windows, blackberry and many more. As we can see here, 114 different brands are there where most of the smartphone users are not even familiar with more than 50% of brands. Although there exists such a large variety of smartphone vendors, Samsung, Apple, Huawei has an enormous market of all, followed by Xiaomi and Oppo.

The Statista revealed some of the new studies regarding the smartphone's market share until the first quarter of 2019. Samsung was the winner with the percentage of approximately 23.1 %, followed by the Huawei, with a 19% share. Next two are Apple and Xiaomi with about 11.7% and 8% respectively. Oppo and Vivo have approximately 7.5% market share each. The remaining other smartphones all together have around 23.3% of the market share.

In spite of the different brands of smartphones present in the market, people have personal reasons for buying a smartphone. As mentioned in the statistics, people mostly prefer Samsung, Huawei, and Apple. There are different trends for purchasing the smartphone in the market.

The most prominent factors are the brand and the service provider, which makes consumers trust them and select their respective brand. The smartphone's design, computing power, operating platform, price, and some other special features attract some customers. Besides those, some only consider their needs such as email, web browsing and document reading.

People have personal reasons for buying a smartphone, but due to the vast diversity of smartphones in the market, people are caught in a dilemma while selecting a phone. If they have the same price but different feature or same features at different rates. Either to purchase a branded one or regular phone to save some investments. Moreover, many consumers regret after buying a smartphone. With the increasing trend of smartphones, different companies are launching new smartphones almost every day with new features and a variety of price tags. Here strikes the need for a recommendation system, that will assist people in optimizing their choices. The system will suggest few suitable options based on user inputs and the user can select from those fewer appropriate options. The system should be able to compare smartphones that were in the market for years and also with the recently released ones. So that the user can make the best choice as per the needs, reducing the possibilities of regret.

2. Literature Review

Smartphone recommendation systems are that system that can assist the user in selecting a smartphone. It is not a completely innovative idea today as we can find many attempts that have been made to create one. The collaborative and content-based filtering is the most popular techniques for building a recommendation system, which recommends based upon the interest of a particular user. Generally, this system collects the information of the attribute of an item and user's interest and with that information system analyzes the user's preferences and profile to assist the user to get the right product. Due to the combined growth of internet and e-commerce, recommendation systems are facing challenges resulting in the overloaded information for the suitable identification of products or services for the user.

According to E.J Salazar and O. Ortega, most of the study conducted on evaluating the recommendation systems depends upon the quality of the recommendations and the recommendation algorithm for decision making. L. Chen and H.K. Tsai stated that decisions taken by the different users in different situation need to be studied. Also, the interface created to interact between the user and the system influences the recommendation system. D.N. Chin said system's design, development and usage depends on user experience, on the characteristics of the user and in the context in which the user is using the recommendation system. Later on, heuristics evolution was formed to figure out the good and bad features of an interface that was developed using ten effective heuristics rules. A particular document called Usability Aspects Report (UAR) contains all these rules and guidelines. Later, J. Nielson state that analyst must focus on what users do instead of designing a practical and easy to use interface. F.J. Martin conducted a study on the user interface design to implement the recommendation system effectively. By analyzing that study thoroughly, the author found out that the essential component in the design and development of the recommendation system is the user interface as it represents 50% of the user experience. Srivastava. J. et al. said that from a long-time user interest and behaviour mining based on the details of web-usage has been used. Based on this cluster of users are formed concerning the website's perspective and create URL groups. G. Xuie et al. introduced a collaborative system based on traditional simple K-means clustering algorithm to smoothen the unrated data details for individual users concerning clusters. J. Lang innovates a new technique called combinatorial vote where a group of agents or voters express their preferences and come to the common conclusion concerning a set of non-independent variables to assign. E. Ephrati and J Rosenschein said that the voting theory principles had been successfully implemented in a multi-agent system for many years to maximize user benefits (A

Mobile Phone Recommendation System with User-Centric Voting Approach, 18, no. 2, ser. 91-102).

Different approaches are taken under consideration to create a better recommendation system, whereas there does not exist an excellent, intelligent recommendation system till now. Many factors, features, and criteria must be considered to make a better decision concerning functionality, features, price, screen size, memory capacity, Internet facility, camera and video, and so on. All these factors were considered to plan and develop a better intelligent system, which is explained in the report thoroughly.

3. Problem statement

As stated earlier in the Literature Review section, there is a lack of proper intelligent systems to the date. Different aspects were considered to develop many smartphone recommendation systems. But for a better system, many factors, features, and criteria must be considered. Even though the use of smartphone and its application are rocketing every day, every month, every year, Study conducted in this field is very less. Only a few ideas have been implemented to develop a smartphone recommendation system. With the increase in smartphone's demand and need in our lives as well as the variety and popularity of now a day's recommendation system is quite necessary to assist the user in making a smart decision.

4. Aims

This project aims to assist the users with an intelligent system based on Artificial Neural Networks to make them invest wisely for the most valued smartphone meeting their needs.

5. Data Collection

5.1. Survey Conduction using Google Forms

One of the crucial parts of our project was to survey to collect the data which would later help in training the recommendation system. It is much more efficient to distribute the survey using social media because of the activeness of people in social media. One of the best tools available for creating an online survey questionnaire is Google Forms, and it was used to create a survey questionnaire in this project as well. Google Forms is an online tool developed by Google to create online forms. Different types of question types are available in google forms such as multiple choices, dropdowns, checkboxes, short answer, paragraph answer, and other options as well. Images and videos can be added to make questions clearer. It allows us to create different sections as well, which helps in organizing those questions in the proper topic. Google Forms also gives an overview of the response data by simple visuals such as pie-charts, bar graphs, and others. Most importantly, the survey responses are in the google sheet with questions in the rows and the corresponding answers in the column.

5.2. Creating Questionnaire using google forms

A google account is required to access the Google Forms. After having the account, you can access the google forms and create a new google form using available options. In that form, we can create the required section using the add section option. After giving the proper name to that section, questions can be added using the add question option. While adding those questions, we can choose the question type (such as multiple choices or checkboxes or short answer or other required ones). We can then type in the question's area and keep the answer option or leave it blank depending upon the question type. Finally, after the creating of a questionnaire, we can get the link for that questionnaire and distribute it online to conduct an online survey.

5.3. Smartphone Recommendation System Survey

There are three sections in the survey questionnaire:

- Personal Section: it was used to collect personal information. It contained questions regarding age, gender, willing amount, annual income and profession.
- Smartphone Section: This section was for collecting data such as which smartphone model they use, the significant reasons for using a smartphone and their feature preferences in a smartphone.
- Consumer Influences Section: This section was introduced to understand the consumer's psychology behind purchasing a smartphone. It included collecting data regarding whether people buy smartphones based on personal research or just based on some social influence such as friends or based on the current trend.

The survey recorded a total of 368 responses.

6. Visualization:

A survey was conducted to collect the data required for the recommendation system. The data was processed and cleaned for visualization and analysis, which was done using Tableau. Tableau is a BI tool used for visualization and analysis. It was designed for people with fewer coding skills and to make data analysis efficient and faster. Simply load data and then filter, join, merge data as per need. In the worksheet also, only dragging and dropping is enough for creating different types of graphs. Legend, colour, different shapes can be given easily from the dashboard. Some of those critical visualizations and analysis of this project were done using Tableau are explained below:

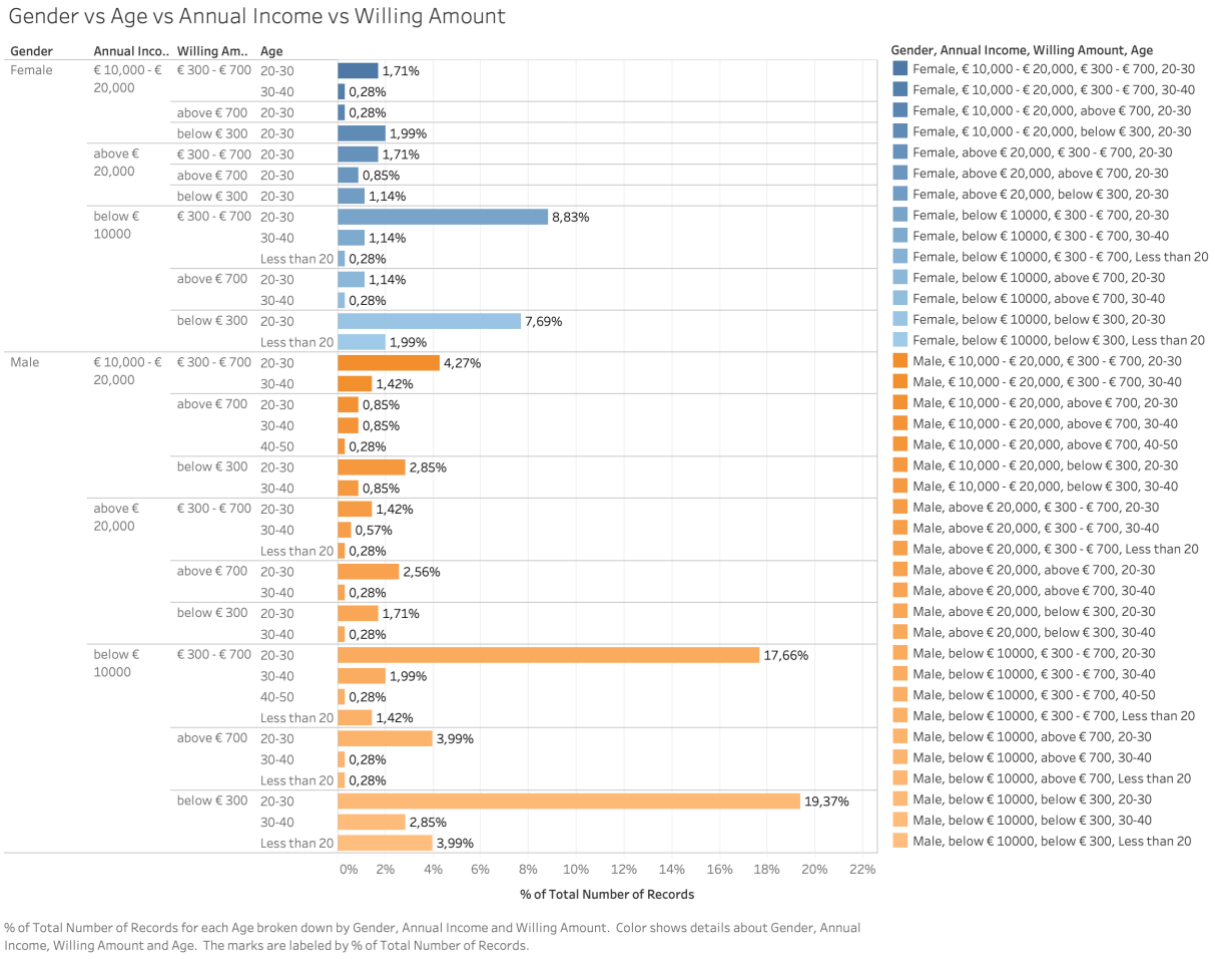


Figure 1: Graph showing the relation between the Gender, Age, Annual income and willing amount

The graph shows the relationship between gender, age, annual income, and the willing amount. In the survey, male and female genders were the only respondents. Annual income per annum has three categories: 10,000 euros, 10,000-20,000 euros, and above 20,000 euros. Age has three categories: less than 20, 20-30, 30-40, and 40-50. The willing amount in the whole project represents the amount of money the user is willing to spend on purchasing a smartphone.

The graph shows that 19.37% of the male of the age group 20-30 with income below 10,000 euros are ready to pay below 300 euros for a smartphone. It is also noticeable that 17.66% of males of the age group 20-30 are willing to pay between 300-700 euros whose annual income is below 10000 euros and being a male.

About the females, approximately 7.69% of the age group 20-30 with annual income below 10,000 euros and are interested in below 300 euros smartphone, and 8.83% of the respondents are interested in smartphone between 300-700 euros.

Moreover, more than half of the male as well as the female population of age group 20-30 with income below 10,000 euros per annum are willing to spend between 0 to 700 euros for a smartphone. Around 44% of respondents want to pay less than 300 euros for a smartphone, and 43% are willing to between 300-700.

One crucial observation is there are people with annual income more than 20,000 euros a year and, only less of them want to spend more than 700 euros on a smartphone. But on the other hand, there are more respondents interested in paying above 700 euros even though they have an income of below 10,000 euros per year.

Also, the females of age group 20-30 are more interested in buying the smartphone of range 300 to 700 euros. Whereas in the case male of age group 20-30, it's just opposite. Most of them are interested in paying below 300 euro and then between 300 to 700 euro and only very few of them above 700 euro.

About the males, the ones with income below 10,000 euros per annum, most of them are willing to pay below 300 euro, and they belong to the age group of 20-30. Whereas Males with an income above 20,000 euros are interested in investing above 700 euros and if we check the data of males with income between 10,000 - 20,000 euros most of them want 300-700 euros smartphone.

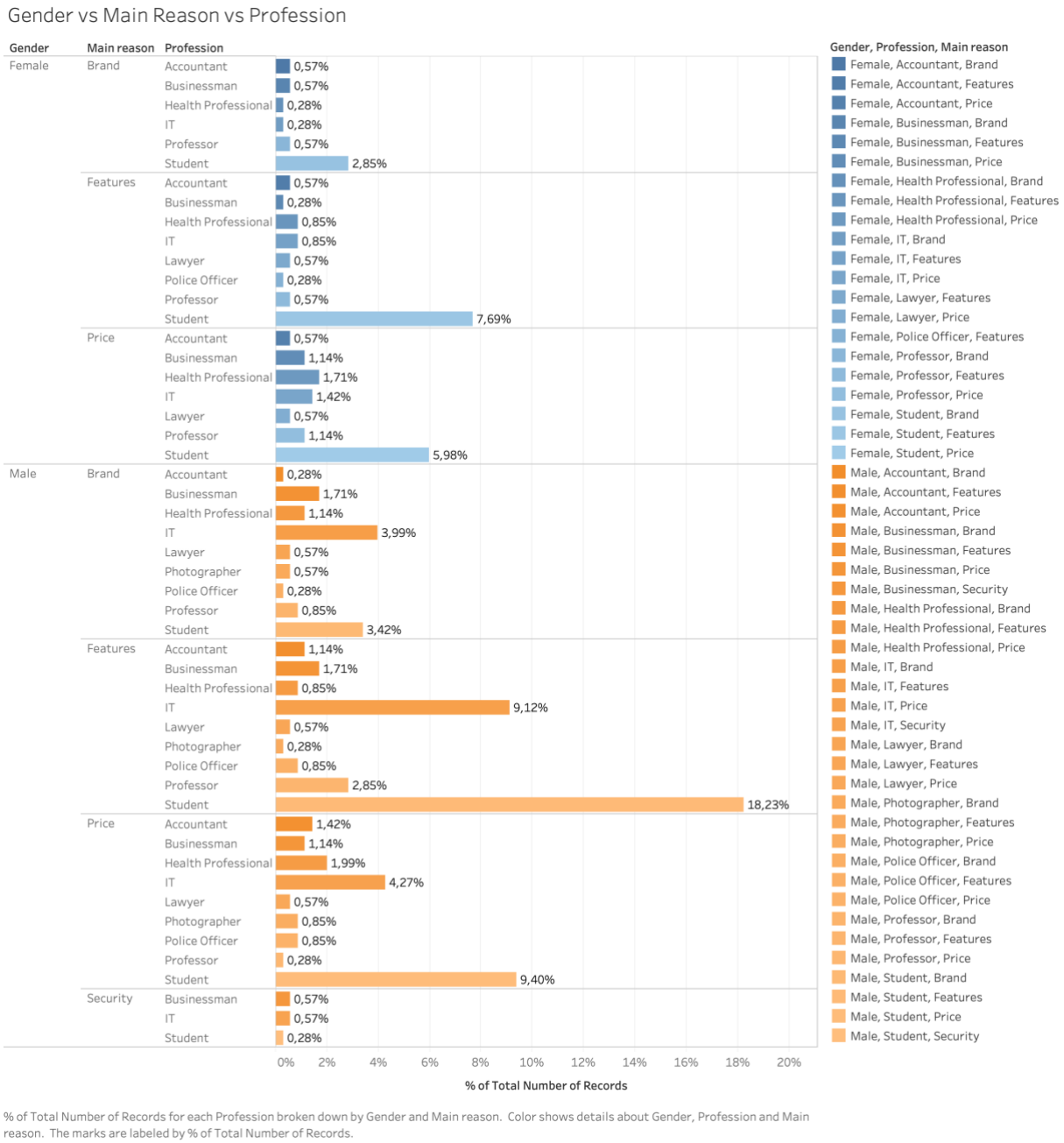


Figure 2: Graph showing the relation between Gender, Profession and Main reason

Figure 2 graph shows the relationship between gender, Main reason, and Profession. The main reason means the basis for one to buy a smartphone which has four categories Brand, Features, Price, and Security. The Profession is categorized into nine categories, as shown in figure 2.

After analyzing the complete graph, one critical analysis is that the highest percentage of 18.23% are male students who purchase a smartphone considering the main reason as features. The next are students going with the price and IT professional considering features.

The female data is found to be similar to that of male data. Most of the respondents are students and go with features. Secondly, they consider price and then brand. But no female was found to be concerned about security in the survey whereas male of profession businessman, IT and

very few students are focusing on the security. Female respondents are very less concerned about brand, and then around 11% are considering price factor and 11% are considering features. That most of the non-student female respondents go with price rather than brand and features.

Considering non-student males, most of them choose features, and maximum are IT professionals. Then they prefer the price of the smartphone, and very few are concerned about the security of the smartphone.

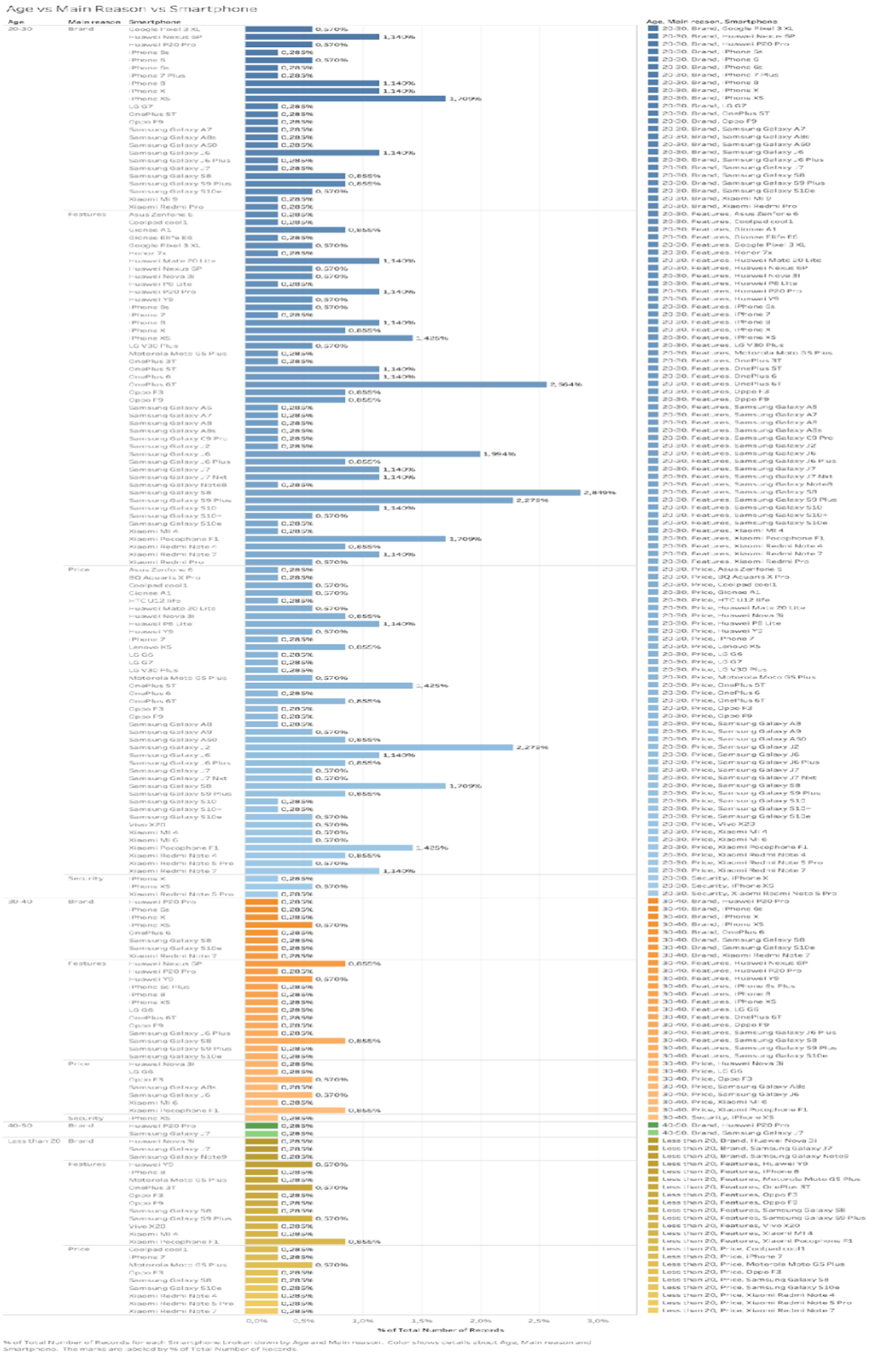


Figure 3: Graph showing the relation of Age, Main reason and Smartphone

Figure 3 above is huge, so its sectioned and its parts are in Figure 4, 5,6,7 and 8.

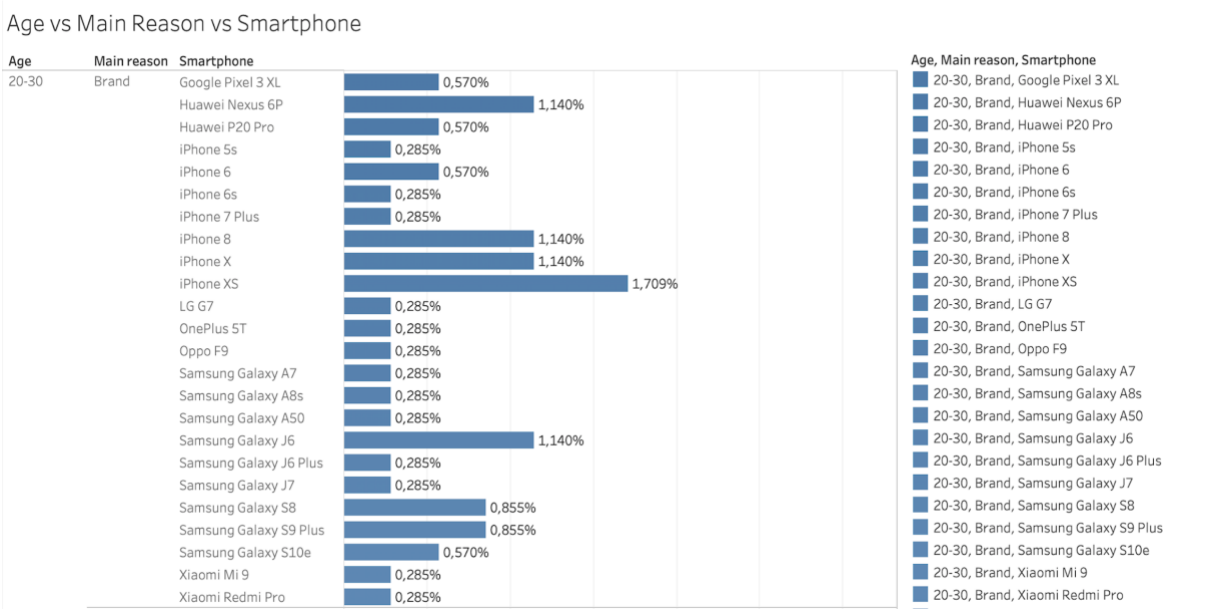


Figure 4: Zoomed in part of figure 3 showing age group 20-30 with main reason as brand and smartphone

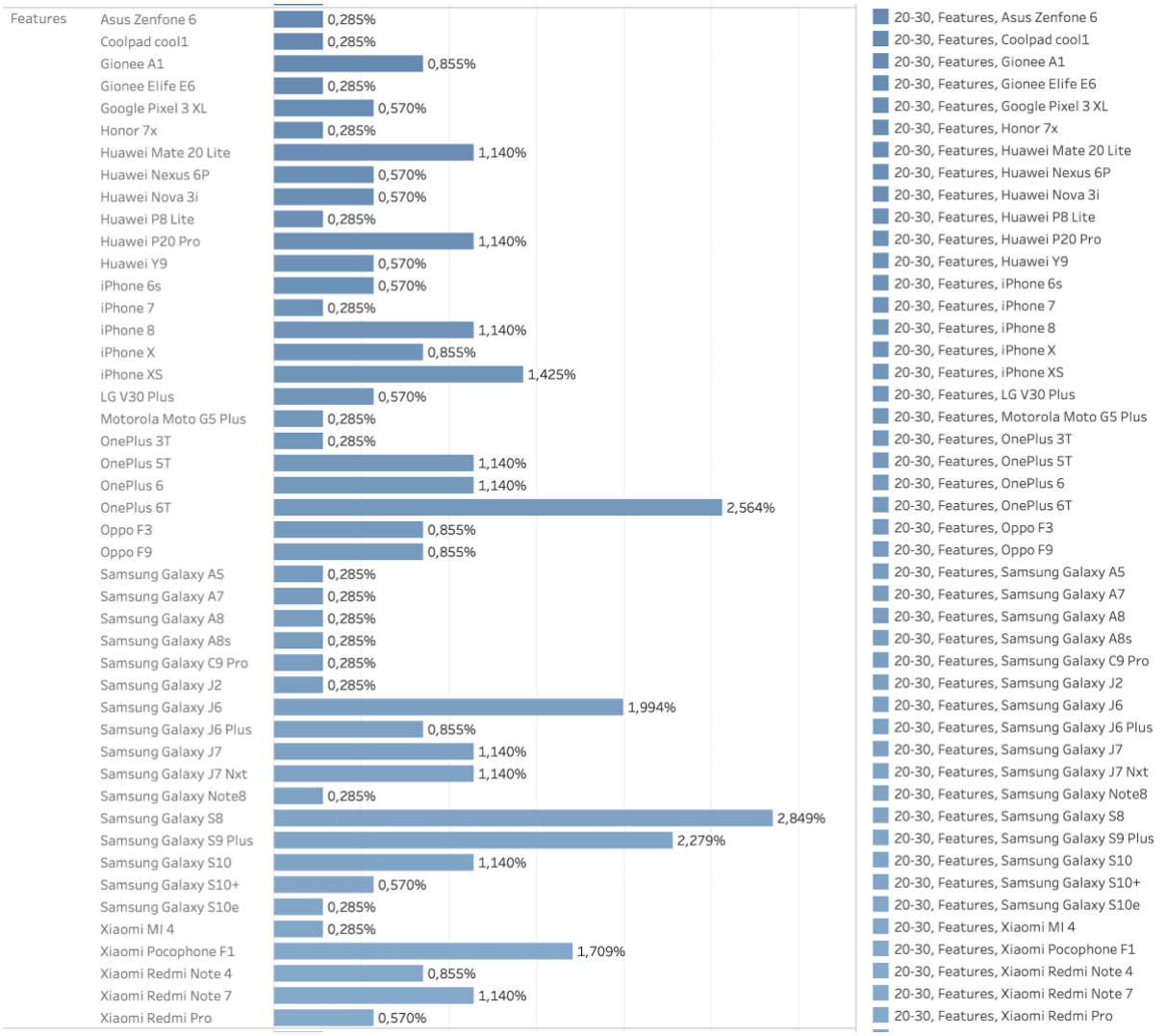


Figure 5: Zoomed in part of figure 3 showing age group 20-30 with main reason as features and smartphone

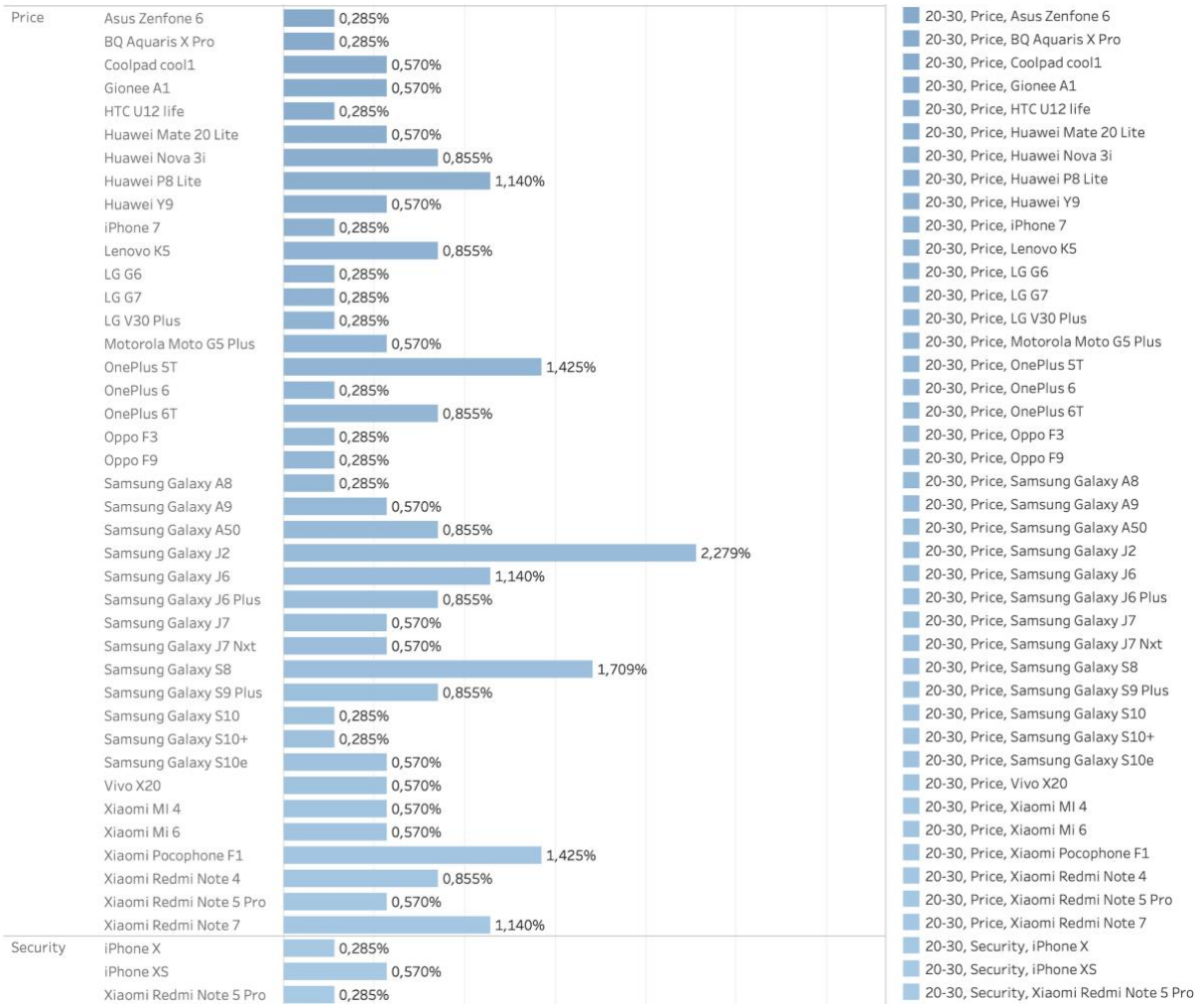


Figure 6: Zoomed part of fig: 3 showing age 20-30 with main reason as price and security and smartphone

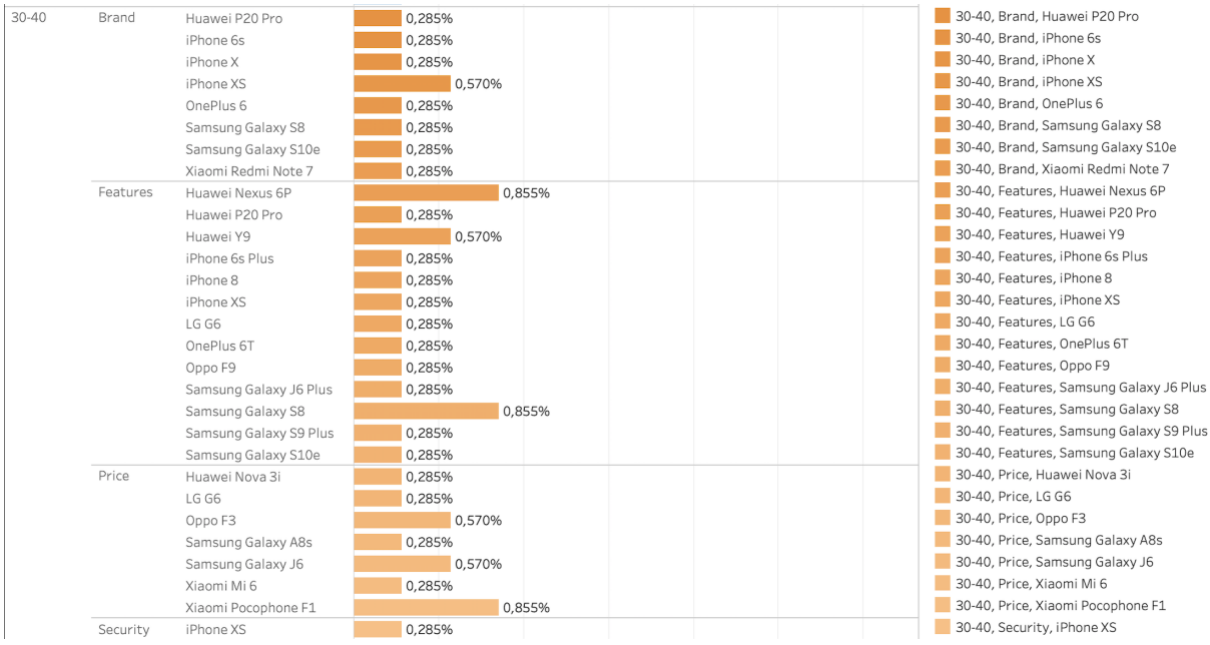


Figure 7: Zoomed in part of figure 3 showing age group 30-40 with main reason and smartphone

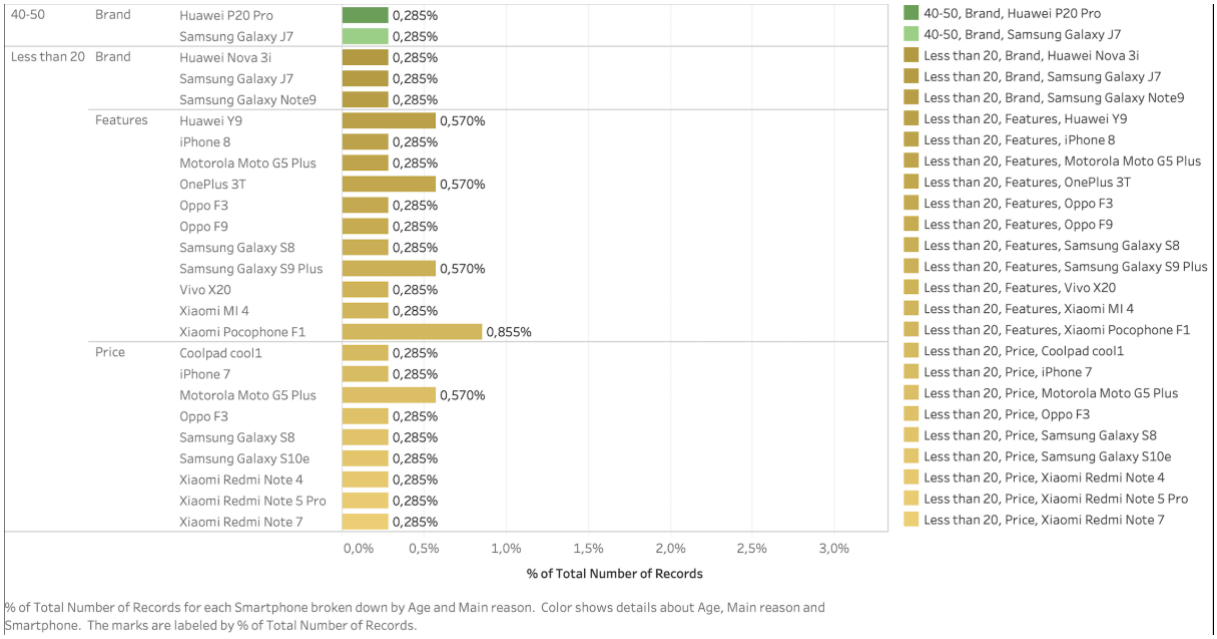


Figure 8: Zoomed in part of figure 3 showing age group 40-50 and less than 20 with main reason and smartphone

Figure 3 consists of the age of category less than 20, 20-30, 30-40, 40-50 and Main reason as Brand, Features, Price, and Security. There are 62 different smartphones found in this survey where most of them are Samsung, Apple, Xiaomi and Huawei.

From figure 3,4,5,6,7 and 8, it can be found that most of the respondents are of age 20-30 and then 30-40 and less than 20 and finally very few are of age group 40-50. Considering the age group 20-30, most of them choose features and are mostly using a smartphone such as Samsung Galaxy S8/S9/J6/J7, OnePlus 6T, Xiaomi, Pocophone F1 and also Apple brand phones like iPhone 6s/8/X/XS. Besides features, they prefer the price and then brand, which was evident in Figure 2 as well. Moreover, the respondents choosing the main reason as the brand uses the iPhone more than other phones, whereas those who prefer features select Samsung phones followed by OnePlus and Xiaomi. Respondents are seen interested in Samsung, where the price is considered the main reason. iPhone is the choice for those with security as a priority. The age group 30-40 people are more interested in the iPhone if they consider brand and Samsung and Huawei for the feature. Respondents who consider price factor are seen more attracted by Oppo, Xiaomi, and Samsung. An almost similar observation in respondents with age below 20 also.

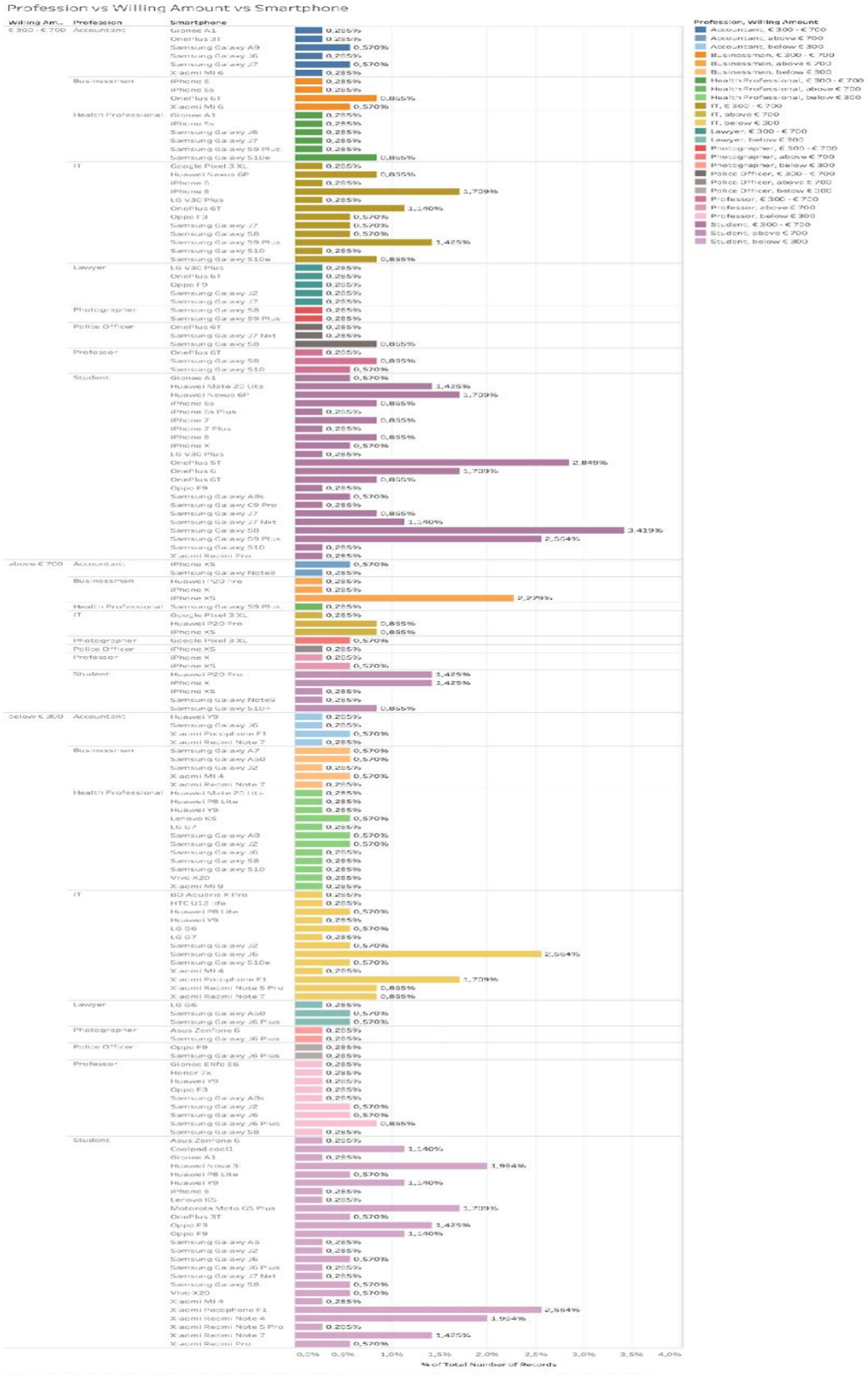


Figure 9: Graph showing the relation of Profession, willing amount and Smartphone

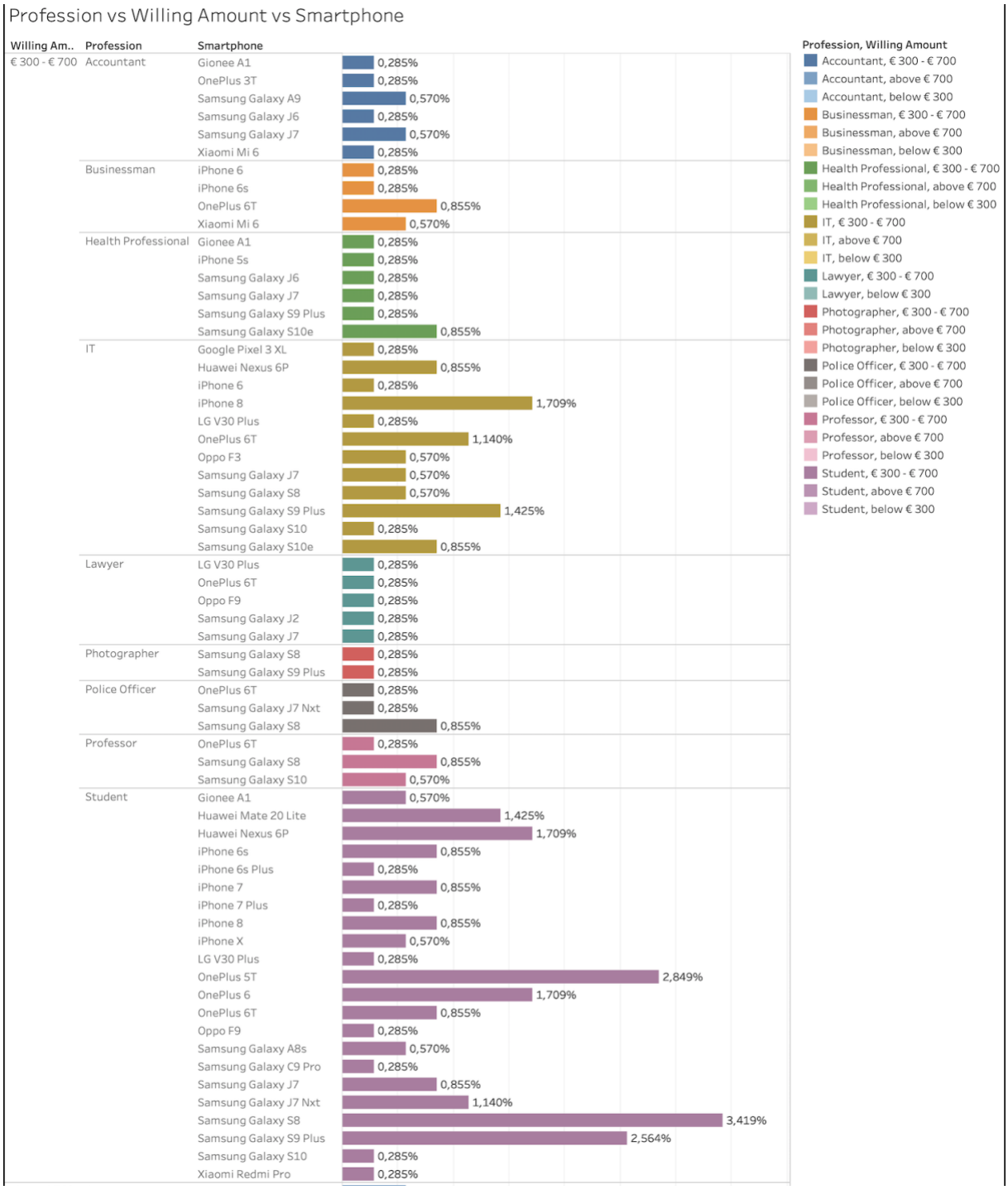


Figure 10: Zoomed part of figure 9 showing willing amount of 300-700 euro with profession and smartphone

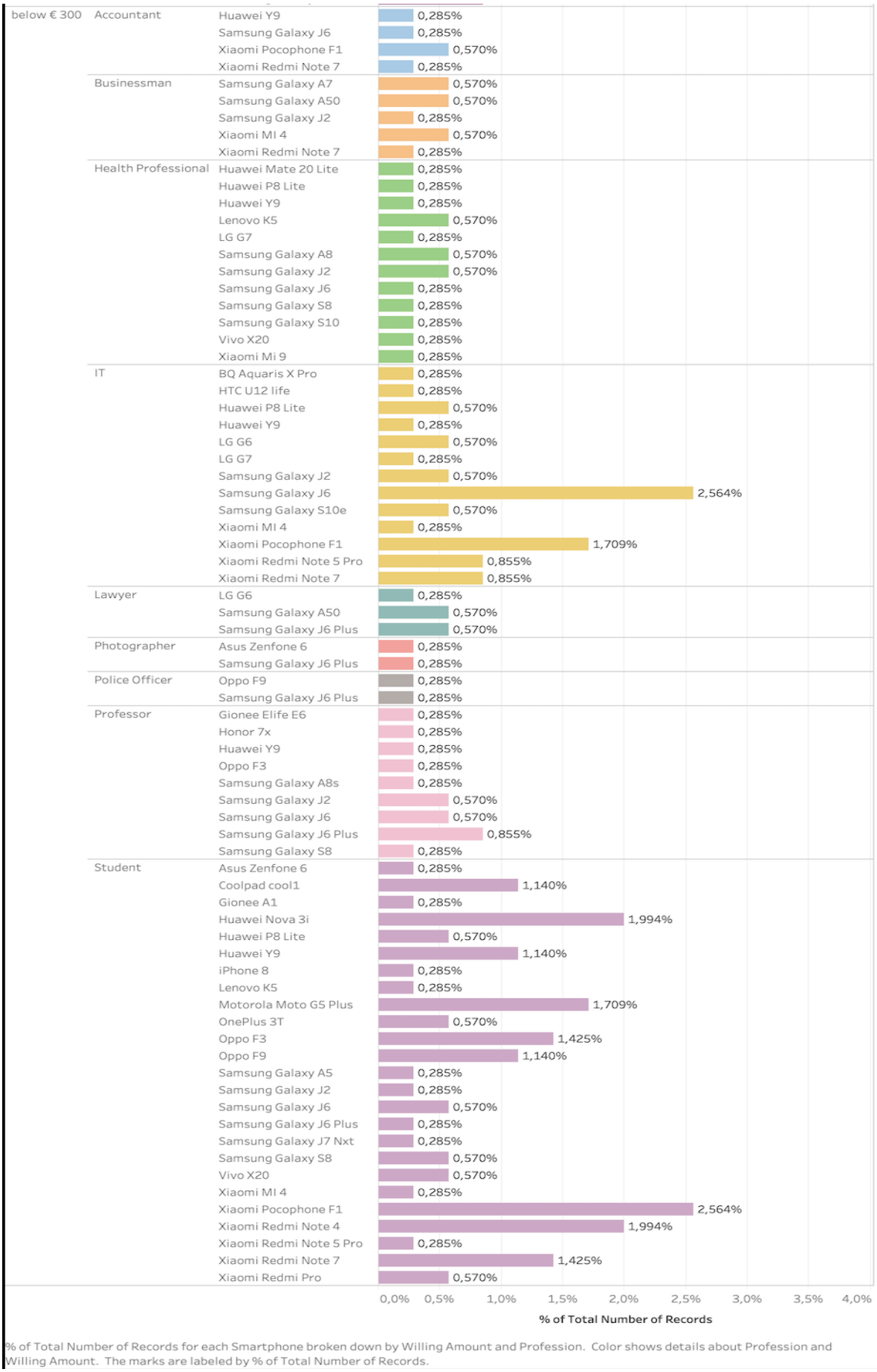


Figure 11: Zoomed part of figure 9 showing willing amount below 300 euro with profession and smartphone

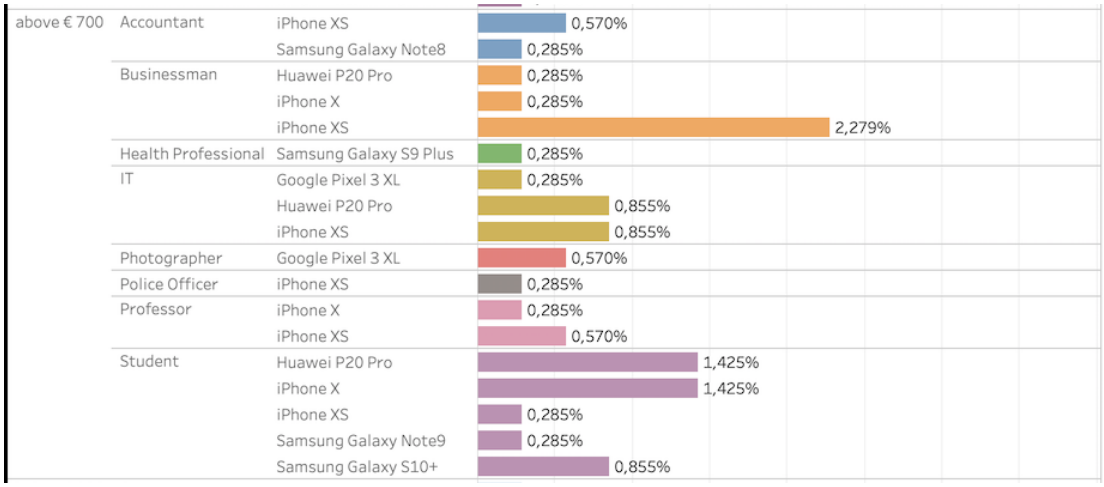


Figure 12: Zoomed part of figure 9 showing willing amount above 300 euro with profession and smartphone

Figure 9,10,11 and 12 shows that most of the students and IT professionals are interested in below 300 euros smartphone. Considering the students willing to pay below 300, most of them are using Xiaomi smartphones of different models leaving behind the market leader, Samsung. Around 3.5% of the respondents use Huawei, which is more than the Samsung users, which are about 2 %. Other smartphones like Oppo and Coolpad are also the choice of the students. With IT professionals, Samsung is still a popular one followed by Xiaomi.

With the willing amount greater than 700 euros, most of them are associated with the iPhones followed by Huawei and Samsung. iPhones XS is specifically most popular with the professional willing to pay more than 700 euro.

By studying the willingness to pay between 300-700 euros on the smartphone, students are found to use Samsung, OnePlus, and Huawei rather than iPhone and Xiaomi. But IT professionals are using iPhone and Samsung. Among the other professions, Samsung is the most popular.

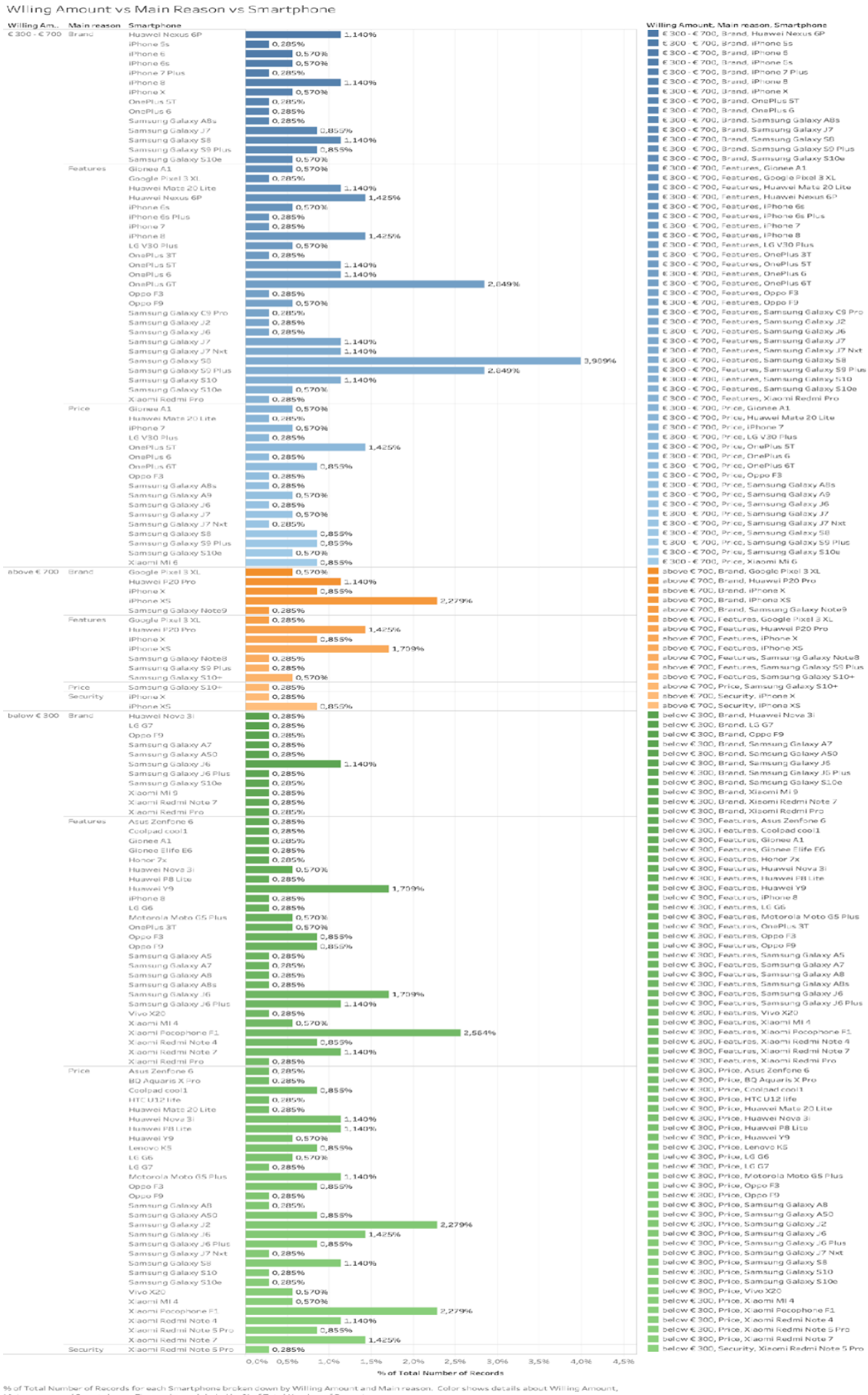


Figure 13: Graph showing the relation of Main reason, willing amount and Smartphone

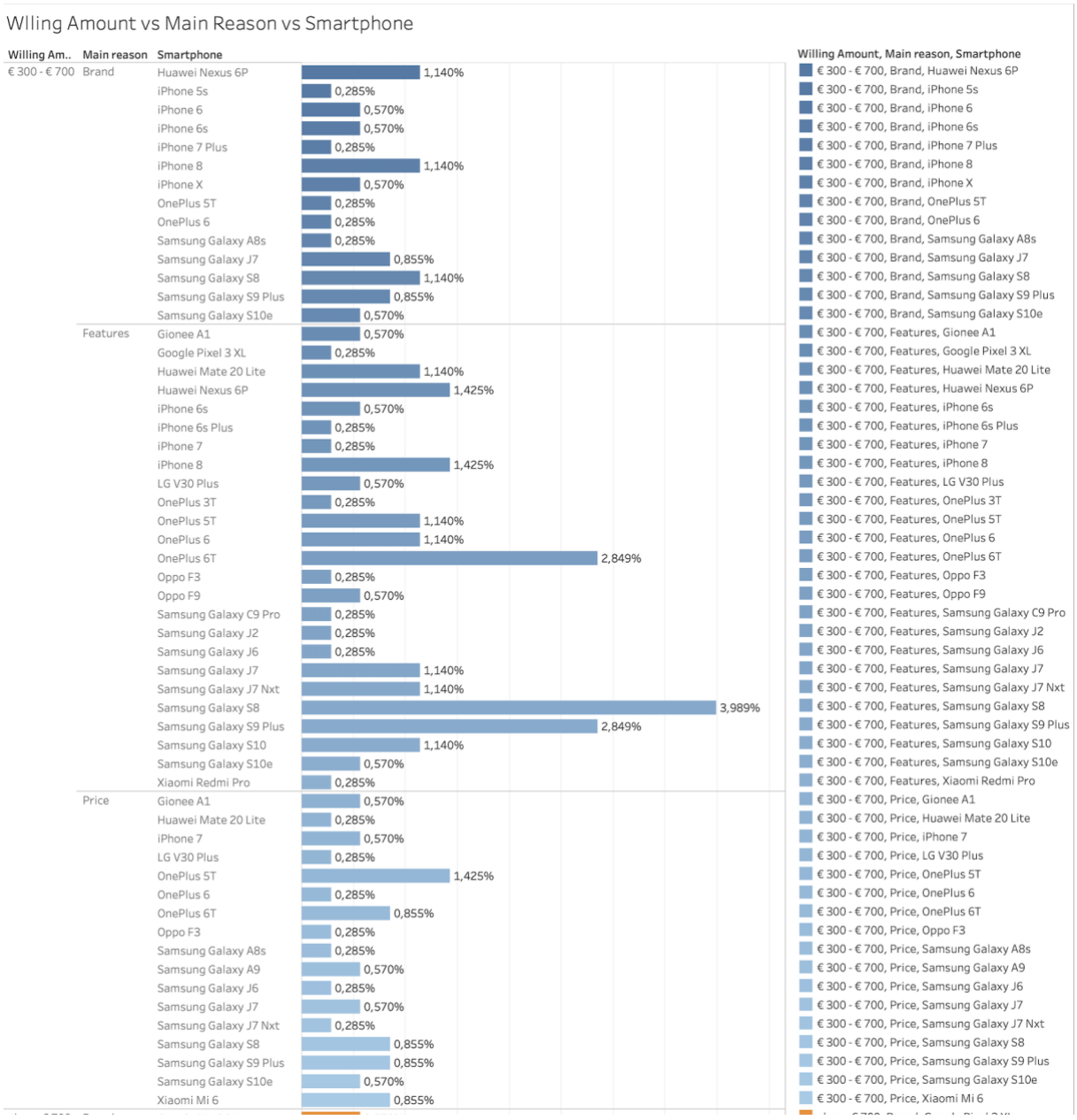


Figure 14: Zoomed part of figure 13 showing willing amount between 300-700 euro with main reason and smartphone

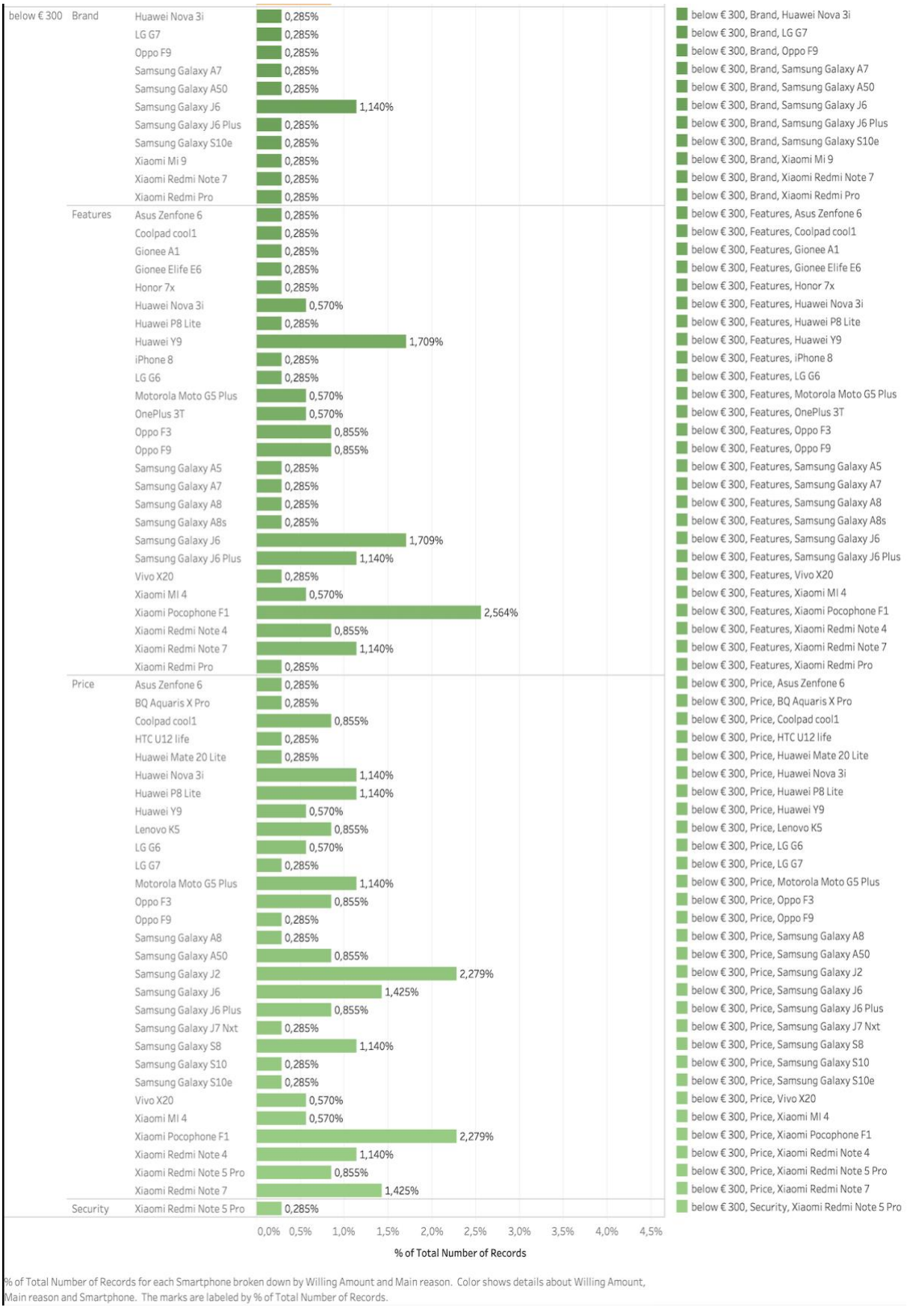


Figure 15: Zoomed part of figure 13 showing willing amount below 300 euro with main reason and smartphone

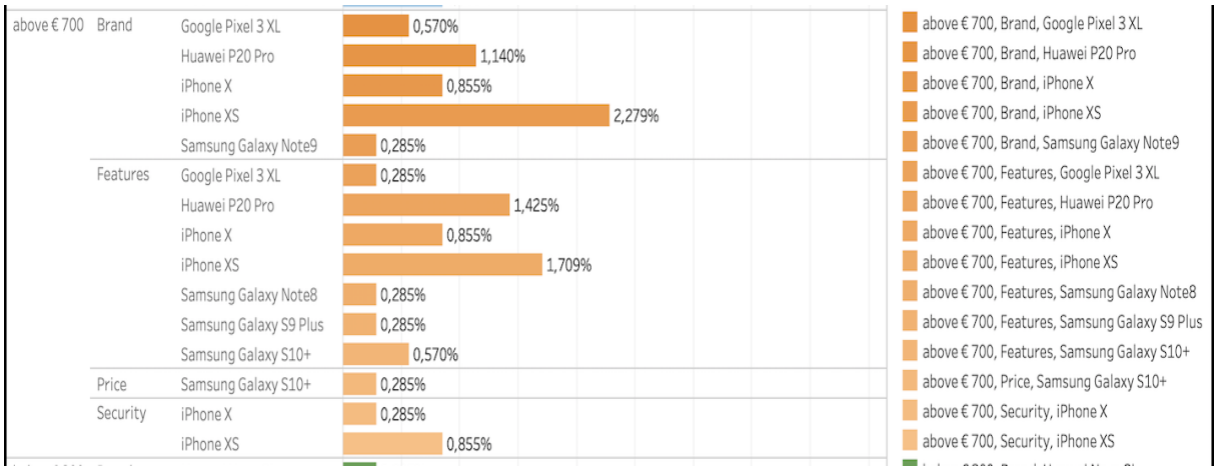


Figure 16: Zoomed part of figure 13 showing willing amount above 300 euro with main reason and smartphone

Figure 13,14,15 and 16 show that those who are willing to pay below 300 euros and considering price are found to be using Samsung the most followed by Xiaomi. Respondents with the amount as a priority are not much fond of the iPhone; instead, they chose Huawei and Oppo and some other brands. Those willing to pay below 300 euros are generally using Xiaomi, Huawei, and Samsung considering features as a priority. But the respondents are found to be using Samsung when considering Brand as the main reason for buying a smartphone.

Samsung and OnePlus are more popular among the respondents willing to pay 300-700 euros and preferring features and price. Those who consider BBrand are found to be using iPhone and Samsung and also there exists Huawei Nexus 6P competing with them.

Observing the data of respondents willing to pay above 700 euro, iPhone is the clear winner. Most of them are using the iPhone, either they are considering Brand or Feature or Security. However, respondents are seen sticking to Samsung when the factor is price.

7. Website

The Website has been created to make the system interactive. The Website has different pages like Home, about us, our team, Contact us, Visualization, and Recommendation. The home page has some smartphone details and some offers. About Us and Our Team section displays the information of the project developing team and their responsibilities. In the visualization page, user can see the visuals of the data from the survey. The visualization page displays different kinds of graphs showing the relations in the data. The next one is the recommendation page consisting of different questions to be answered by the user for getting the smartphone recommended. There are three classifications in the inputs: Personal information input, Personal preference input and last is features input. In the personal information section, there are some questions regarding general information of the user. The Personal Preference section consisted of questions regarding the user's preferences in smartphones and the feature section make some inquiries about features required by users. Most of the factors for the recommendation, including the limitations in previous research and recommendation system have been considered to achieve the best results. Besides this, there is an output page that displays five outputs, i.e. five smartphones are recommended. Among them, the first one is the most suitable for the user, and the other four are similar featured alternatives to have more choices. The image and the specification of the most suited smartphone are displayed side by side on the same page. The other four smartphones have image, name, and price on that page and click on the icon will open the next page with the bigger picture and specification of the smartphone. Another significant aspect of the website is its responsiveness, i.e. the compatibility with all screen sizes such as smartphones, computers, and tablets. The tools of web-development used in creating such an interactive website are in the next section.

7.1. Web Development

Web development requires:

- Front End Development
- Back End Development

7.1.1. Front End Development

Front-end development is also known as client-side development and is the interactive part of the website design. It helps users to interact with the website. Everything a user can see on any web page is some combination of HTML, CSS, and JavaScript. HTML, CSS, and JavaScript are popular and powerful tools for front-end development. The website for this project was built using those tools as well. Brief introduction of them is below.

7.1.1.1. Hyper Text Markup Language (HTML)

The primary use of HTML is to design the layout of a web page. The browser reads the .html file and renders its content in the webpage. The layout design of every web page is done using HTML in this project. Index.html file is our homepage. We also have visualization.html, recommendationInput.html, and recommendationOutput.html.

7.1.1.2. Cascading Style Sheets (CSS)

Cascading style sheets (CSS) are used to style the document written in a markup language. A selector selects an HTML element, and then the CSS rule is applied to that selector. In this project, the style.css file contains code to style our HTML documents. One CSS file can handle multiple HTML documents. CSS is used to design font size, colour, background, images and other styling in the project website.

7.1.1.3. Bootstrap

Bootstrap is a CSS, HTML and the JavaScript framework for creating responsive web pages. Responsive web design allows a webpage to adapt to the screen of the user's device. This project uses some bootstrap class like forms, buttons, navigations for the responsiveness of our website.

7.1.1.4. Font Awesome

Font Awesome has been used to get some icons like a Facebook icon, Twitter icon, LinkedIn icon, and others on the Website.

7.1.2. Back End Development

Back end development is also referred to as server-side development. It creates communication between the browser and the database. The Flask framework has been used to develop the back end.

7.1.2.1. Flask

Flask is a micro web framework based on Python. On the Website, there are several questions for the user, and the answers to those will act as input to the recommendation system. Flask was used to handle these inputs. Flask push method sends data to the server. In Python, data is fetched from the server and fed to the recommendation system. After the evaluation of data using ANN (Artificial Neural Networks), the result is passed to the browser using Flask. Flask retrieved the information of the recommended phone with four other alternative recommendations to the browser from the server, and the output page displays the suggested smartphones.

8. Database

A database is an intelligent collection of classified information which is easy to access, manage, and update. In this project, it is used to store and retrieve mobile phone specification data. Firebase Realtime Database, a cloud-hosted database provided by Google, was well suited for this project and used.

Data is in JSON (JavaScript Object Notation) format. The database is connected to link all the clients and servers in real-time. While constructing cross-platform applications with iOS, Android, and JavaScript SDKs, all clients share only one Realtime Database instance and automatically obtain updates of the latest information.

Moreover, the Firebase Realtime Database allows creating wealthy and cooperative apps through safe customer code access to the database. Data remains local and, while real-time events are still offline, the end-user will be able to react to the situation. The Realtime Database synchronizes local information modifications, when the unit is connected again, with remote updates, which happened during the offline client and automatically merged any conflict.

8.1. Overview of Firebase database is shown below: P

Figure 17: Overview of Firebase database

("How Using Firebase Can Help You Earn More")

The Realtime Database is a NoSQL database. Compared to a relational database, it has distinct optimizations and functions. The Realtime Database API is intended only to allow quick activities. This API enables creating an excellent experience in real-time that can serve millions of customers without compromising their reaction.

8.2. Authentication and Data Access from Firebase Database

The Firebase service allows account's authentication programmatically. A unified admin SDK is used to authenticate several functions, including database, storage, and authentication. The Administration SDK enables you to communicate with Firebase in privileged settings for actions such as:

- Read and write Realtime database information with complete privileges for administration.
- Create and verify tokens for Firebase authentication.
- Access resources on the Google Cloud Platform, such as Cloud Storage Buckets and Firestore databases for your firebase initiatives.

For the authentication, Admin SDK configuration provides the option for generating the new private key for different platform like Node.js, Java, Python and Go. That private key consisting of the URL of the database with a credential for the authentication as shown below.

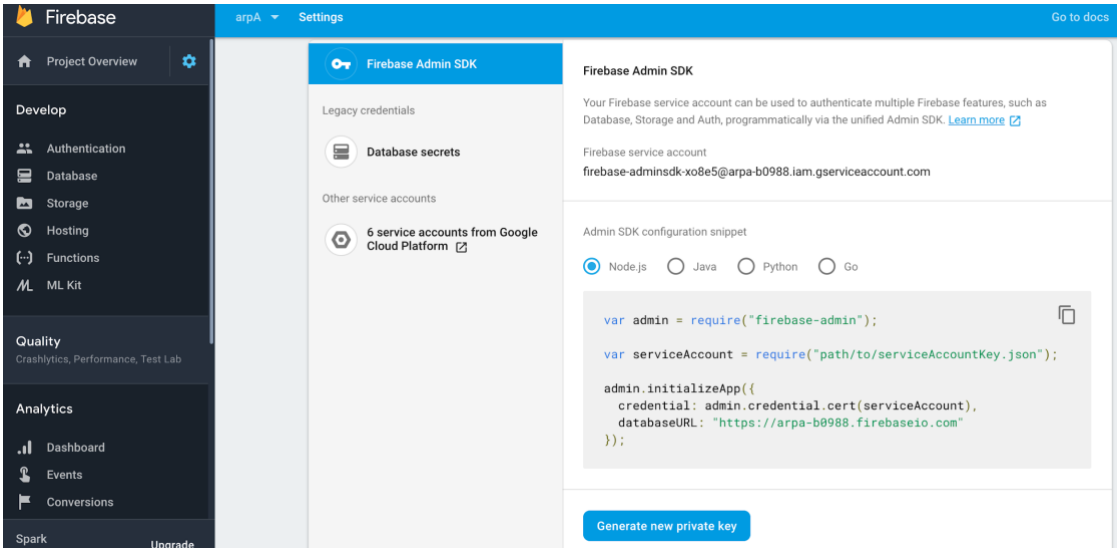


Figure 18: Authentication on firebase

On the smartphone recommendation system, when the system needs to access the database, it is done with the generated private key from Admin SDK, which holds the information about the authentication. Then Firebase verifies the database credentials and after the successful authentication only, accessing the database is possible from the recommendation system. The flow diagram of the process is in Figure 19.

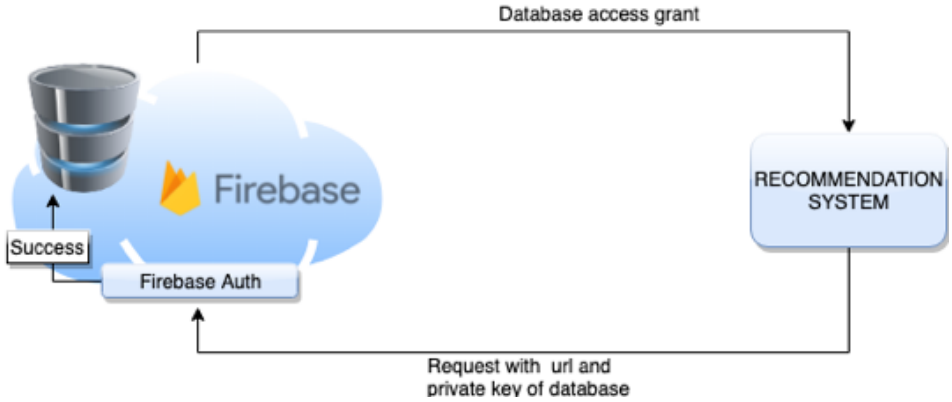


Figure 19: Process flows for the authentication

8.3. Database Structure

All data in Firebase Realtime Database are JSON objects. Unlike the SQL databases, there are no tables or records. JSON is an open, standard format of lightweight text-based data exchange. In JSON, there are four fundamental and integrated data types. They are strings, numbers, Booleans (true and false) and null. Besides, there are two structured data types as objects and arrays. '{', and '}' wraps the object and are known as a list of label-value pairs. Arrays are enclosed by '[', and ']' and are a list of values.

On this project, the Smartphone name is an object of hash and specification as a key-value pair, as shown in figure 20. For instance, Gionee A1 as an object of hash and detail information like Battery, Battery_Tag, Display_Resolution, Display_size, Main Camera tag, MainCamera, OS, OS_Tag, Price, Price_Tag, RAM, Selfie_Camera, Sensors, Storage and URL is the list of values in an array.

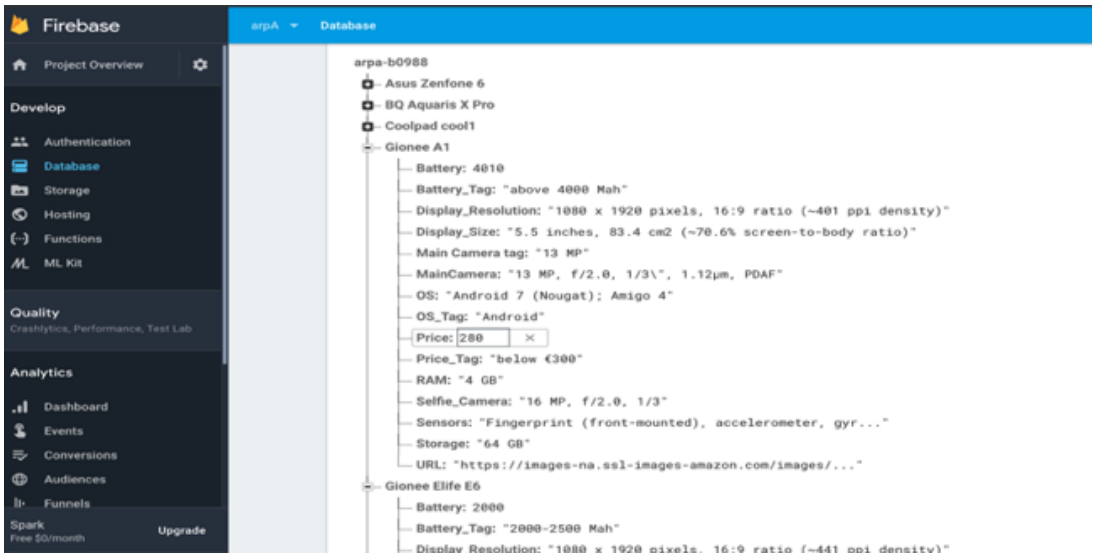


Figure 20: Database structure

8.4. Pros and Cons of Firebase Database

8.4.1. Pros

As the data is hosted on the cloud, accessing the data with secure authentication is easy, and no need to invest in the hardware infrastructure. With the massive storage potential, it can plan for long. The storage size can increase and decrease in real-time.

8.4.2. Cons

This project uses the free version of the firebase database with some limitations on database storage and the number of connections. The Firebase authenticates only 10,000 times per month. It provides 1 GB of storage, which is insufficient for massive databases. It only allows 100 concurrent connections, that means only 100 clients can access the database simultaneously.

9. Data Encoding

Data here represents the input to the recommendation system. Most of the time, those inputs are in the categorical formats such as “iPhone”, “Samsung” and many others. Computers cannot directly understand data in such form and encoding of data is required.

Data Encoding means to encode the data into understandable computer format so that a computer can recognize the data later.

Computers can well understand binary formatted data of 0’s and 1’s.

If there are only two inputs, we could assign 0 to the one Input and 1 to another, but most of the times, there are more than two inputs. In those case combinations of 0’s and 1’s is used to encode multiple Inputs.

Figure 21: sample question for data encoding

In Figure 21, the four possible outcomes are “Brand”, “Features”, “Price”, and “Security”.

“Brand” is encoded as 100,

“Features” is encoded as 010,

“Price” is encoded as 001, and

“Security” is encoded as 000.

The position of 1 is determining the outcome.

If the 1’s is at first position, then the computer understands it as “Brand”.

If the 1’s is in the second position, then the computer understands it as “Features”.

If the 1’s is in the third position, then the computer understands it as “Price”.

And if there are only zeroes computer understands it as “Security”.

If there were only three inputs “Features”, “Price”, and “Security”.

Then,

“Features” would be encoded as 01,

“Price” would encode as 10, and

“Security” would be encoded as 00.

To generalize this concept,
if there are N number of inputs, then we can encode each Input with N -1 binary values.
In the figure above, we have four inputs, and each input is encoded using
N -1 = 4 - 1 = 3 binary values.
“Brand” = 100 (3 binary values 1,0 and 0).

If we have N inputs then with N -1 binary values, we can represent N -1 number of Inputs with different 1’s positions. And the Last one is encoded with all the zeroes.
Let’s take the example of our encoded sample. We have
“Brand” is encoded as 100,
“Features” is encoded as 010,
“Price” is encoded as 001,
“Security” is encoded as 000
The first three representations have different 1’s position, and the last one has only zeroes. This is the concept of Data Encoding.

10. Recommendation System

10.1. Introduction

The recommendation system is the trained machine learning model which recommends the user needs (such as a movie, a food item or anything) based on some inputs (such as their interest, age, preferences, and others). There are different systems for recommending movies, food items, and the system created for this project is the **SMARTPHONE RECOMMENDATION SYSTEM**. **The smartphone recommendation system was implemented using Artificial Neural Networks.**

10.2. TOOLS USED for smartphone recommendation system:

- **Programming language: python**
- **Libraries:**
    - **Pandas:** The primary task is to import data files.
    - **Numpy:** it is used to implement mathematical functions.
    - **Sklearn:** it is used to process the data, splitting the training and the test sets, and others.
    - **Keras with Tensorflow backend:** This library is used to implement deep learning models such as Artificial Neural Networks.

11. Artificial Neural Networks

To understand the concept behind the Artificial Neural Network, let us imagine the structure of the neural network in the biological brain as it is the blueprint for Artificial Neural Networks. The natural brain's neural network is the interconnected network of neurons with different functions that take information and pass around different neurons for the learning process. Likewise, Artificial Neural Network is a network of interconnected artificial neurons(nodes). They perform information processing and passing of data to different nodes in different layers to figure out the underlying trend and learn from them.

11.1. Reason for using Artificial Neural Network

The data collected from the survey consisted of a high number of inputs and a large number of outputs. There was no linear relation between them. For the proper Output, the system needed to understand the underlying trend in the data correctly. Using Artificial Neural Networks, it is possible to study the vast data with the help of a high number of interconnected artificial neurons. The neurons pass and process the information to other nodes. This passing of information in the network helps the network to study and figure out the trend efficiently.

11.2. Artificial Neuron and Activation Function

An artificial neuron is the unit of the Artificial Neural Network. It takes input information and provides outputs. These neurons may pass data to another node, or process information and provide an output based on that. The one which only forwards information does not have any activation function, but the one which processes the data has an activation function.

An activation function is a mathematical function associated with the artificial neural nodes. First, the information is input to the neuron, and the input parameters are placed on that mathematical formula and calculated to give output. Below figure shows a neuron implemented with an activation function.

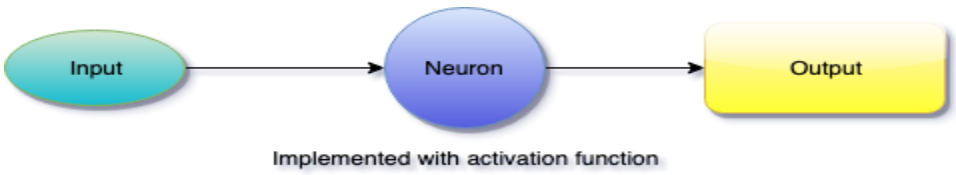


Figure 22: Artificial Neuron with Activation Function

11.3. LAYERS IN ARTIFICIAL NEURAL NETWORKS

There are three layers in Artificial Neural Networks.

- Input Layer
- Hidden Layer
- Output Layer

11.3.1. INPUT LAYER

It is a collection of input nodes that gather information from the outside world and pass to the network. It takes the inputs in the encoded form. Nodes in this layer are not implemented with an activation function because this layer does not perform calculations. The total count of nodes in the input layer is the number of inputs from the data.

11.3.2. HIDDEN LAYER

Generally, the hidden layer performs calculations on the data passed from the input layer and transfers the data to the output layer. To perform the calculation, the nodes in the hidden layers are implemented with an activation function. Sometimes there might be a huge number of inputs, and it might be quite complicated to find relations between the inputs. There are a few more challenges when it comes to the hidden layer. There is one input layer and one output layer but, how many hidden layers we need? The number of inputs in input nodes is the number of taken inputs, and the number of outputs nodes are expected outputs, but how many nodes in each hidden layer? There are a few rules to answer these questions.

11.3.2.1. Defining number of layers

Defining the correct numbers of hidden layers is extremely important. If we define the lesser number of layers than required, then the system may not be able to figure out the underlying trend of the data. That leads to unusual predictions, and the situation is called underfitting. And if we define more than required, then it fits the data very well but starts capturing noises in data as well, which causes high variance in the output. That again leads to inaccurate predictions, and the mechanism is called overfitting. Preventing both is a high priority for the model to work efficiently. There are no specific rules to prevent underfitting and overfitting, but there are some factors to be considered, which can help us. Some of them are:

- After data analysis, if a linear relationship exists between the data, then no hidden layer is required because the output layer itself can process such data.
- If there is a possibility of mapping the data with some relation may be and data is not so massive, then one hidden layer is required.
- Based on the complexity and randomness present in the data, the total number of inputs and outputs, and required complexity in learning two or more hidden layers are required.

It was about defining the number of hidden layers. The next part is defining the number of nodes in the hidden layer.

11.3.2.2. Defining number of Nodes:

The next challenge after defining the number of layers is defining the number of nodes. Similar to defining the number of layers, defining too less than the required number of nodes might lead to underfitting and too much than required might lead to overfitting which makes important factor to consider for machine learning model to work efficiently. The total number of nodes in each hidden layer is the same. Here as well, there are no hard and fast rules for defining the number of nodes, but one of the best approaches is:

number of nodes = (number of input nodes + number of output nodes)/2

The number of nodes in the hidden layer should be less than or equal to this value of the number of nodes we got from the calculation.

Example:

Case 1: **Result is even**

Number of inputs nodes = 20

Number of output nodes = 12

Then, numbers of nodes in hidden layer = (20 + 12)/2 = 16

Here 16 or would be the best choice.

Case 2: **Result is odd**

Number of inputs nodes = 15

Number of output nodes = 12

Now we get 15+12 = 27/2 = 13.5. In these cases, the rule is to choose the lesser value and chose 13 rather than going for 14.

So, the best choice here is 13.

Using this is how the number of nodes in each hidden layer is defined.

11.3.3. OUTPUT LAYER

It is the set of nodes that are responsible for the final calculation, and an activation function is used to implement. After the final calculation, this layer transfers the information to the outside world. The total number of nodes in the output layer is the number of expected outputs.

12. Learning process of a Neural Network

To understand what is the learning process of an Artificial Neural Network let us see the diagram below:

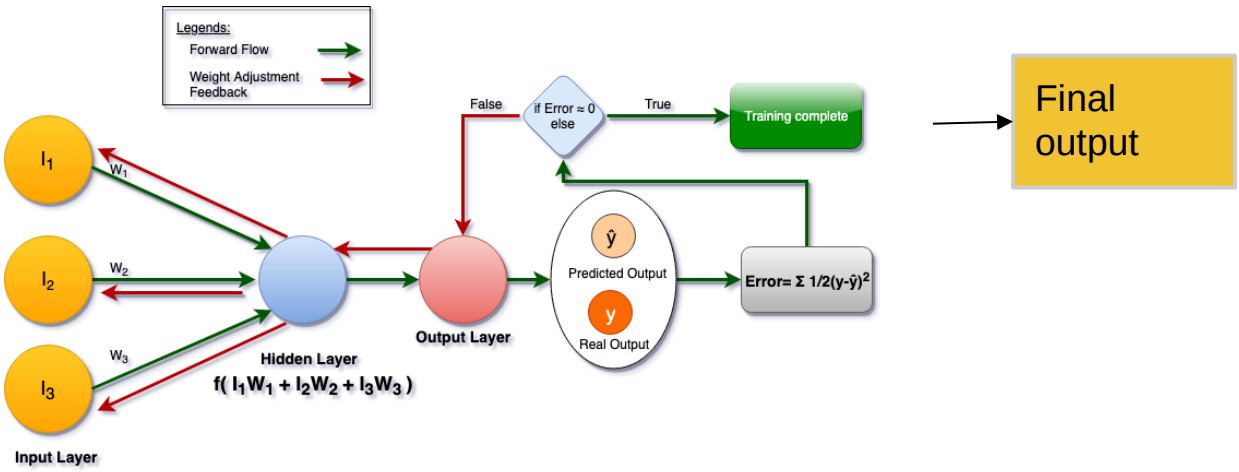


Figure 23: Learning process workflow

The I_1, I_2, and I_3 are the input nodes and as a whole called input layer. W_1, W_2, and W_3 are called weights, and they are randomly assigned numerical values. As the input layer is supposed to transfer the information to the network, it just transfers input values to the network. However, these inputs are multiplied with corresponding weight and added before they are given to the hidden layer as the input value.

First multiplication to corresponding weights

I_1 with $W_1 = I_1W_1$

I_2 with $W_2 = I_3W_3$

I_1 with $W_1 = I_1W_1$

Then all are added, and it becomes

$I_1W_1 + I_2W_2 + I_3W_3$.

Now, the hidden layer receives the data. Hidden layer nodes have activation functions, which are mathematical functions associated with the nodes to perform the required calculations. The value $\mathbf{I_1W_1 + I_2W_2 + I_3W_3}$ acts as the input for the activation function of the hidden node.

After the calculation of $\mathbf{f\ (I_1W_1 + I_2W_2 + I_3W_3)}$ in the hidden layer, the output layer receives the calculated result.

In the output layer, output generates after the final calculation. The result from the hidden layer acts as the input to the activation function associated with the output node. The output generates after the calculation. ŷ represents the predicted output. The next step to compare it with the real

output y. The predicted output is the output based on some inputs to the neural network, and real output is the output based on the actual data for model training. Then the predicted output is compared with the actual output based on the formula:

$$\text{error} = \Sigma\frac{1}{2} * (y - \hat{y})^2$$

After the calculation of the error, if the error is not very near to 0, the feedback is sent back. Feedback is given to the output layer to send feedback to the hidden layer node and then to input layer nodes, to readjust the assigned weights. Then the flow is backward for giving feedback to every node for new weight assignment reassigning the weights. After reassignment again, inputs are multiplied and added, sent to the hidden layer, and the output layer. Again, a new prediction is made and compared to the real output. Again, if the error value is not approximately equal to 0, the process repeats, but if the error value is approximately equal to 0, then that is the final output value. It is how they are learning using the strategy of comparing the predicted output with real output, and the reassignment of the weights.

13. Architecture of the Smartphone Recommendation System

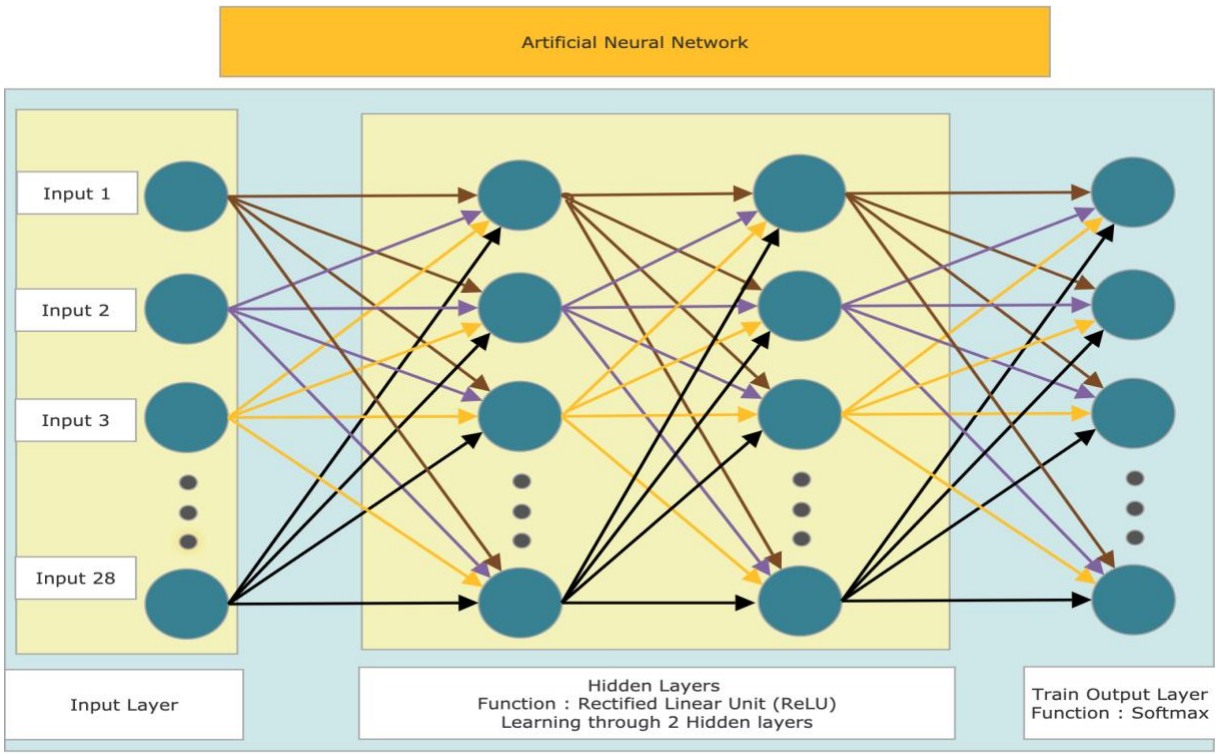


Figure 24: Smartphone Recommendation System Architecture

Figure 24 is the architecture of the smartphone recommendation system.

It consists of 1 Input layer, 2 Hidden Layers, and 1 Output Layer.

13.1. Input Layer

Let us understand the Input Layer of our system first. The input layer has 28 nodes. Let's understand the reason behind the 28 nodes. To understand the reason behind total input nodes, let us see the input form of the questionnaire from our website.

The **first one** is the personal information section:

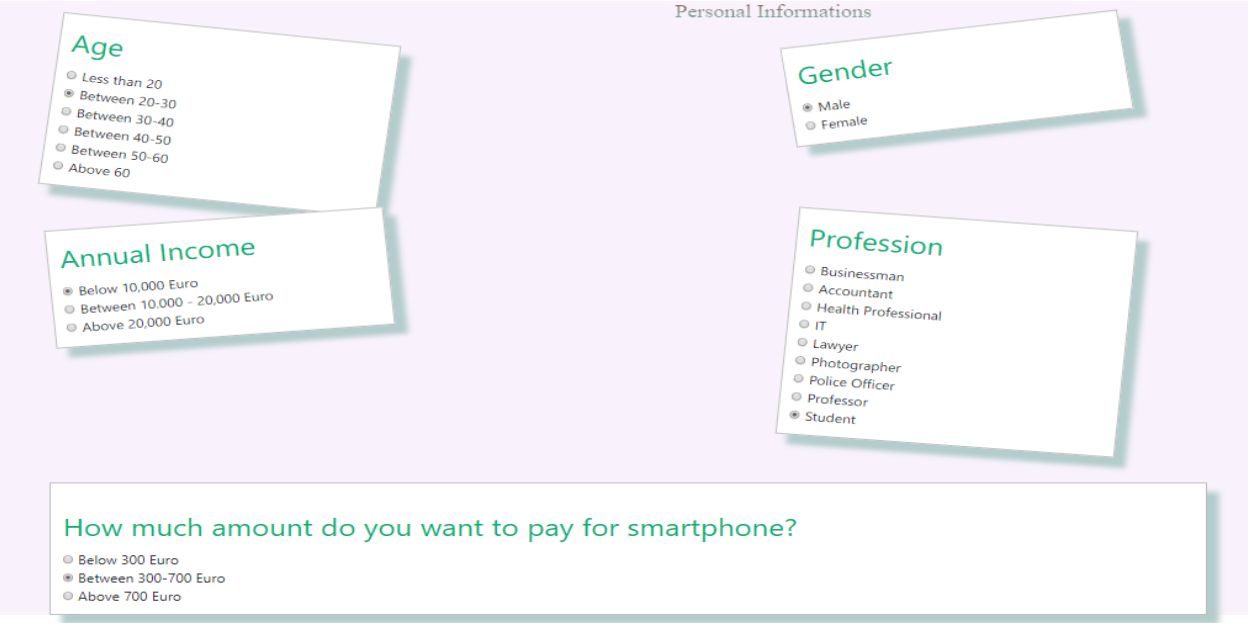


Figure 25: personal information input

There are age, gender, annual income, profession, and willing amount in the personal information input section, as seen in figure 25.

- In **Gender**, there are two possible outcomes. We need to encode data using a 2-1 = 1 binary number for the required combination, as explained in the data encoding section.
- In the **Annual Income,** there are three possible outcomes, so two binary numbers combinations are needed.
- In the **Professions,** there are nine possible outcomes, so eight binary numbers combinations are needed.
- In the **Want to Pay**, there are three possible outcomes, so two binary numbers combinations are needed.
- **Age** section is one exception here. There are six outcomes, but 2 of the 50-60 and above 60 data are redundant as they were both made equal to 40-50. So, that makes four non-redundant inputs, so three binary numbers for combinations. The reason is in the limitation section.
- The total input from this section is **1+2+8+2+3 = 16.**

The **second one** is personal preference input section:

Figure 26: personal preference input

- In the **Operating System,** there are three possible outcomes, so we need two binary number for combinations.
- In the **Smartphone Preference,** there are four possible outcomes, so we need three binary number for combinations.
- In the **Graphics or Game Preference,** there are two possible outcomes, so we need one binary number for combinations.
- In the **Photography Preference,** there are two possible outcomes, so we need one binary number for combinations.
- The total number of inputs from this section are: **2+3+1+1 = 7.**

The third and the last section is features input section:

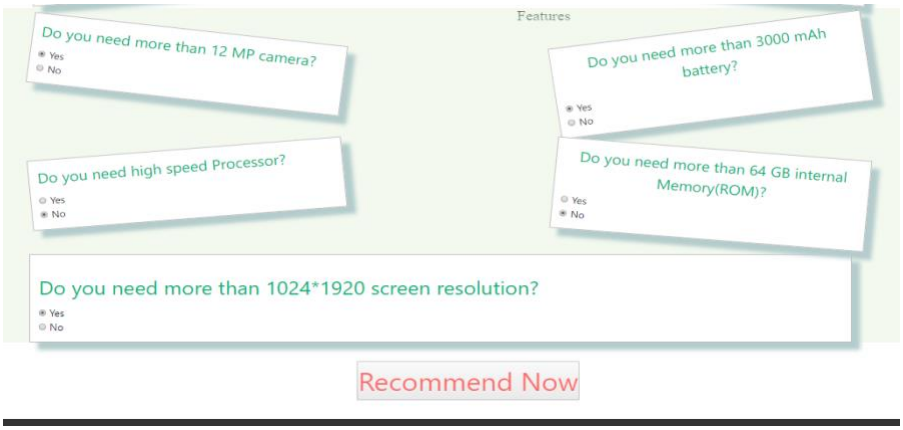


Figure 27: features input

In the feature input section, all of the questions regarding camera, battery, high-speed processor, ROM (Memory) and screen resolution have two possible outcomes, so, there is 1 binary number for all required combination.

That makes **1+1+1+1+1 = 5** of total inputs from this section.

The total required nodes= **16 + 7 + 5 = 28.**

That's the story behind the requirement of 28 nodes in the input section.

13.2. HIDDEN LAYER

- There are two hidden layers.
- Based on the defined criteria in the defining number of hidden layers section, the input and output are high in number. There was no linear relation and required complexity in learning was high, so two hidden layers are required for the required learning.
- There are 28 input nodes and 63 output nodes,

  The nodes in each hidden layer The nodes in each hidden layer

  = (number of input nodes + number of output nodes)/2

  = (28 + 63)/2

  = 91 / 2

  = 45.5

  Which makes 45 hidden nodes each layer.
- The hidden layer's node has a Rectified Linear Unit (ReLU) activation function. (The project team tested with other activation functions as well, but the accuracy was high with ReLU function)

13.2.1. Rectified Linear Unit (ReLU) Function:

The diagram and formula representing ReLU function are shown below in figure 28. It is being represented by R(z) = max (0, z)

This function works as:

- if the input(z) is negative then R(z) = 0.
- if the input(z) is a positive then R(z) = z.

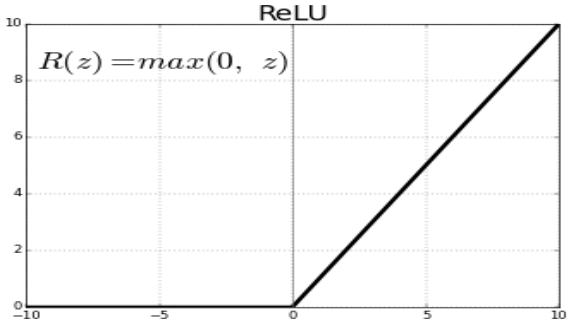


Figure 28: Rectified Linear Unit Function

("Relu: Not A Differentiable Function: Why Used in Gradient Based Optimization? And Other Generalizations of Relu.")

13.3. Output Layer

The output layers consist of 63 output nodes. The number of different smartphone models from the survey was 63, and that is the reason behind 63 output nodes. The activation function used for the hidden layer nodes was the **SoftMax function**.

13.3.1. SoftMax function

The SoftMax function is the activation function represented by the formula in figure 29 as $S(y_i)$:

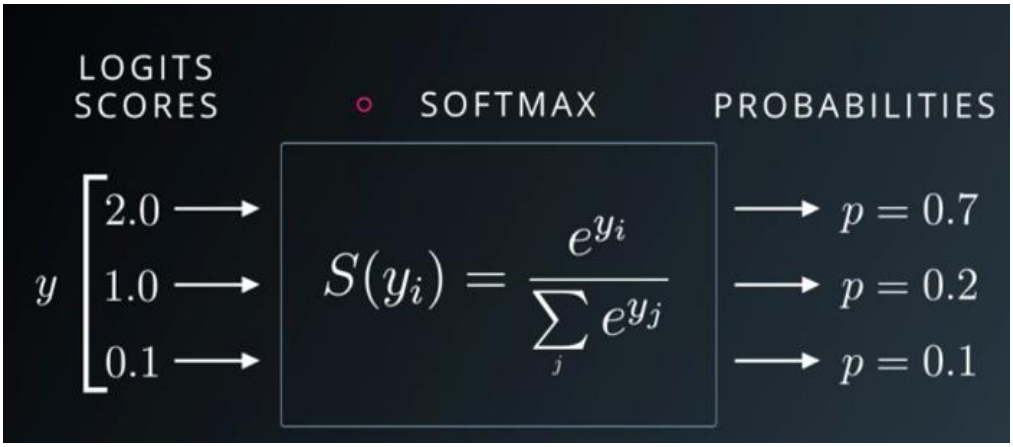


Figure 29: SoftMax Function

("Beginner'S Guide to Building Neural Networks in Tensorflow")

The SoftMax function takes input, and based on that it calculates the probability for each outcome. The sum of all the probabilities adds to 1.

Like in the above image, there are three different probabilities with three different cases of y.

Why SoftMax function?

The output we are expecting is the model name of the smartphone, and there are multiple outputs as we have 63 different models. One of the reasons is that the SoftMax function is to do with SoftMax function being one of the best functions to deal with multiple output situations. The most important reason for using this function is that after getting inputs from the hidden layer, it can calculate the probability of each phone being recommended based on the given input. Finally, the one with the highest probability is the one to be recommended.

14. RECOMMENDATION SYSTEM WORKFLOW

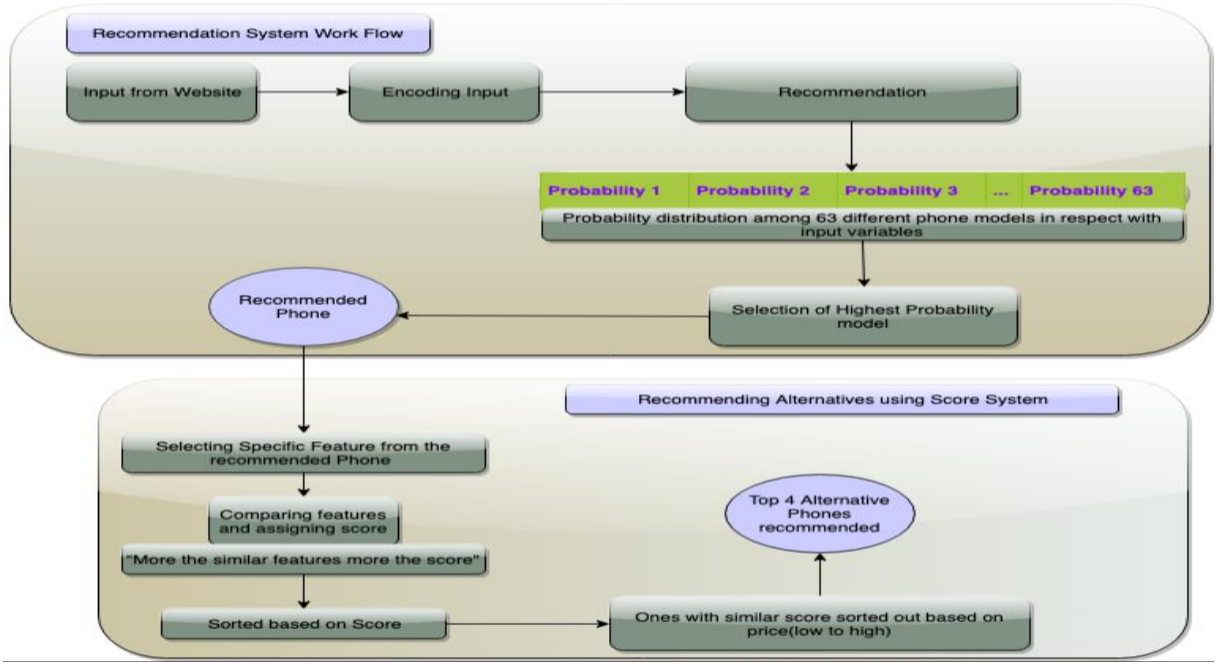


Figure 30: workflow of recommendation system

First, the recommendation page of the website provides the inputs. The input is in Categorical form and hence encoded before passing it to the recommendation system. Then the recommendation system calculates the probability of each phone being selected based on the given input, and finally, the system recommends the one with the highest probability.

Score System based on matching features was built to recommend similar featured alternatives. The algorithm for alternative recommendation:

- First, comparing the specific features, the score is assigned to each matching feature.
- The score increases with the increase in the number of matching features.
- Then sorting is done based on the score from highest to lowest.
- For similar scores sorting is from cheapest to expensive.
- Finally, the system recommends the first four.

15. Result

The Webpage displays the recommended smartphone and four more common featured alternatives based on the inputs of age, gender, annual income, willing amount, profession, feature usage, and purpose of using a smartphone.

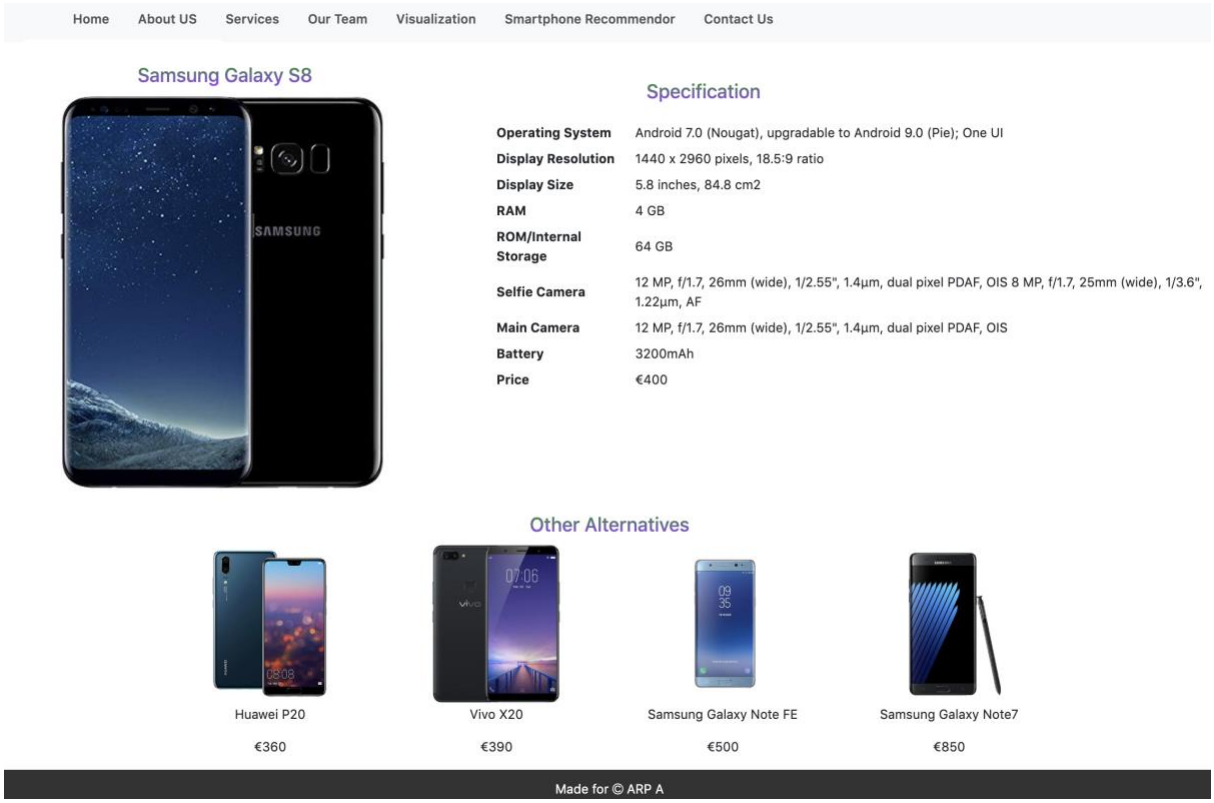


Figure 31: output page

16. Limitations

The survey data was not enough, and because of which system faces some limitations:

- Age factor: The survey data did not consist of the ages 50-60 and above 60. But the project team decided to keep those options on the website because any age group might be interested in using the smartphone. The age 50-60 and above 60 are considered similar to age 40-50 because of limited data. The oldest available group was 40-50, and that was the main reason behind the consideration of 50-60 and above 60.
- A Limited number of outputs: The total number of distinct smartphones received was just 63, which limited the possible outputs for significant recommendations.
- Less Accuracy: As the model was trained using limited data, the possibility of getting expected output is less.
- Limited phones in database: Currently, there are only (249) of smartphones in the database compared to that of the total number of existing phones. It is very less, and because of this, the possibilities of the alternative recommendations are limited.

17. Outlook

The system can be trained better with more survey data to increase the accuracy and precision of the output. The size of the smartphone's database can be increased and maintained with the precise database of mobile by extracting information of the mobile sets that has a problem like battery explosion and stopped manufacturing, in recent years. Furthermore, it can be hosted on a web server and make it accessible to the public.

18. Conclusion

Finally, implementation of the intelligent system for recommending smartphones using the Artificial neural network was successful with some of the limitations, and the aim to improvise it soon.

References

Holst, Arne. “Number of Smartphones Sold to End Users Worldwide from 2007 to 2018 (in Million Units).” https://Www.statista.com/Statistics/263437/Global-Smartphone-Sales-to-End-Users-since-2007/, Www.Statista.com. Accessed 4th July, 2019.

Prof. P. Govindarajulu, and Mr. Y. Subba Reddy. A Mobile Phone Recommendation System with User Centric Voting Approach, 18, no. 2, ser. 91-102, Feb. 2018. 91-102, paper.ijcsns.org/07_book/201802/20180212.pdf.

"What Is A Front-End Developer?". Frontendmasters.Com, 2019, https://frontendmasters.com/books/front-end-handbook/2018/what-is-a-FD.html.

"Relu: Not A Differentiable Function: Why Used in Gradient Based Optimization? And Other Generalizations of Relu.". *Medium*, 2019, https://medium.com/@kanchansarkar/relu-not-a-differentiable-function-why-used-in-gradient-based-optimization-7fef3a4cecec.

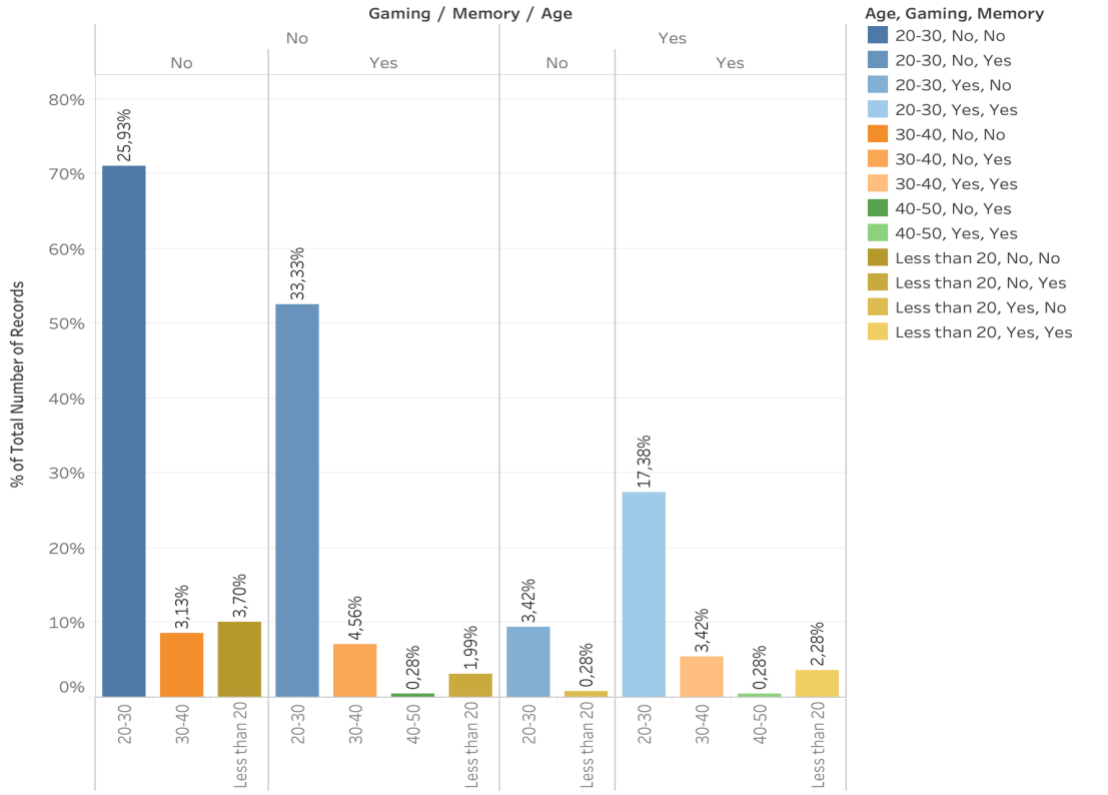
"Beginner’S Guide to Building Neural Networks In Tensorflow". Medium, 2019, https://towardsdatascience.com/beginners-guide-to-building-neural-networks-in-tensorflow-dab7a09b941d.

"How Using Firebase Can Help You Earn More". Google, 2019, https://www.blog.google/products/admob/how-using-firebase-can-help-you-earn-more/.

Appendix

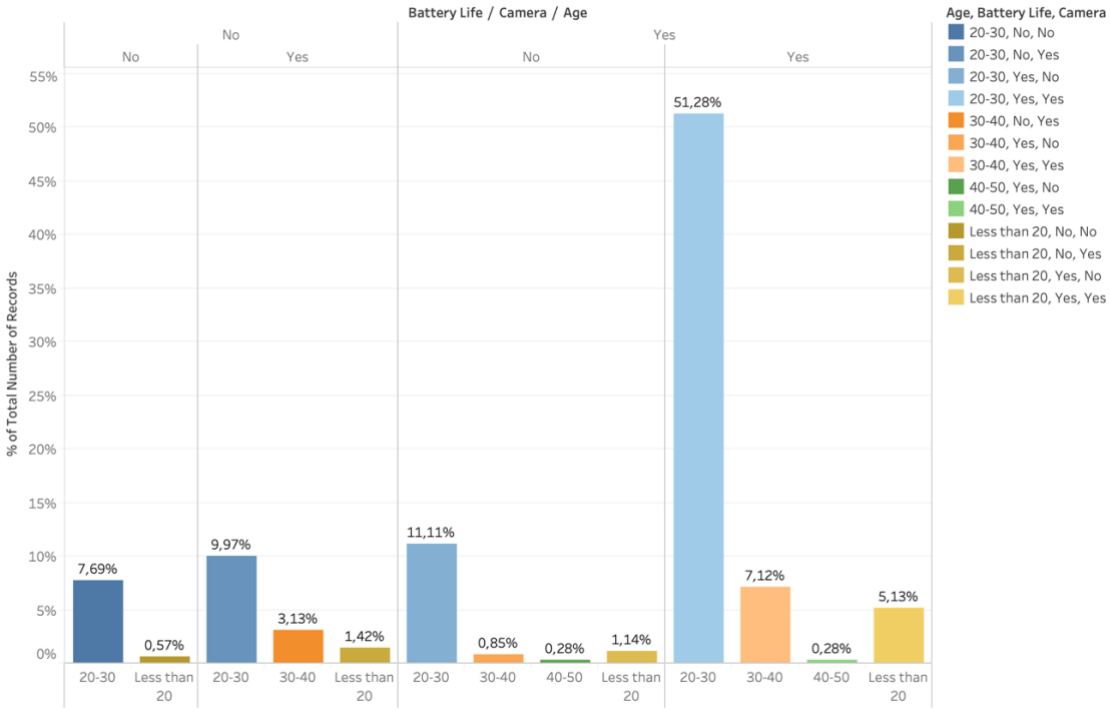
Some more visualization which is not explained in the report on detail are listed below:

Age vs Gaming vs Memory



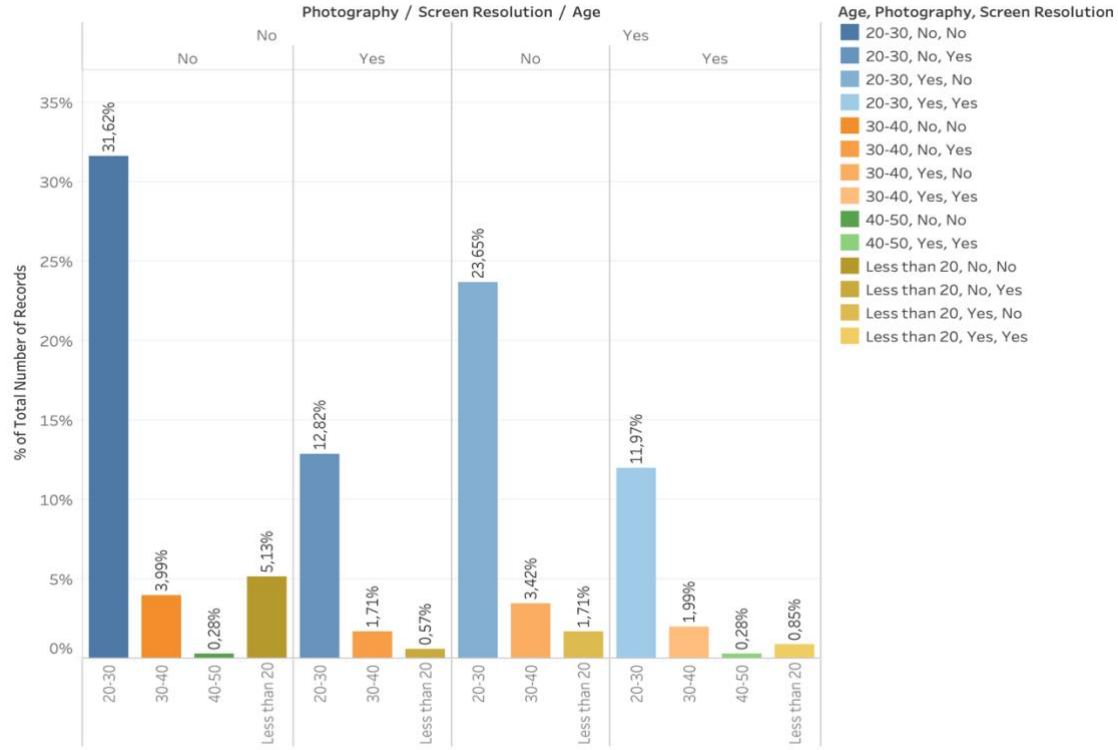
% of Total Number of Records for each Age broken down by Gaming and Memory. Color shows details about Age, Gaming and Memory. The marks are labeled by % of Total Number of Records.

Age vs Battery Life vs Camera



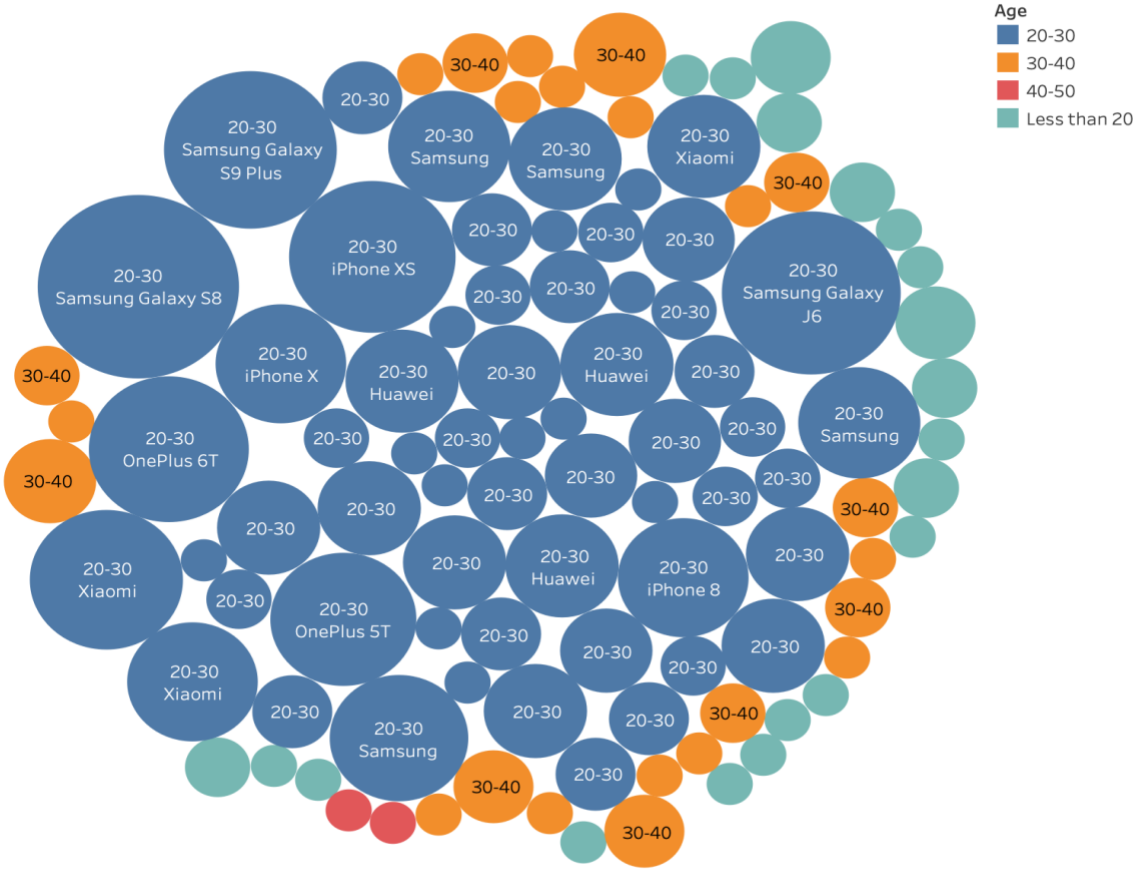
% of Total Number of Records for each Age broken down by Battery Life and Camera. Color shows details about Age, Battery Life and Camera. The marks are labeled by % of Total Number of Records.

Age vs Photography vs Screen Resolution



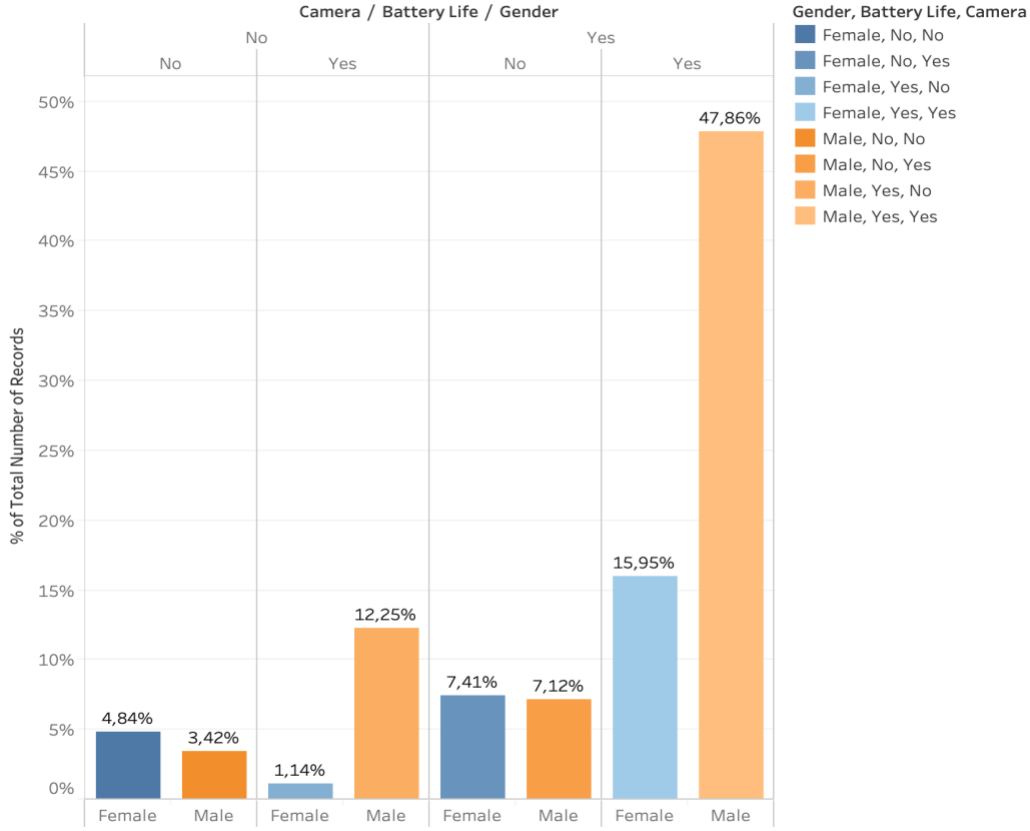
% of Total Number of Records for each Age broken down by Photography and Screen Resolution. Color shows details about Age, Photography and Screen Resolution. The marks are labeled by % of Total Number of Records.

Age vs SmartPhone



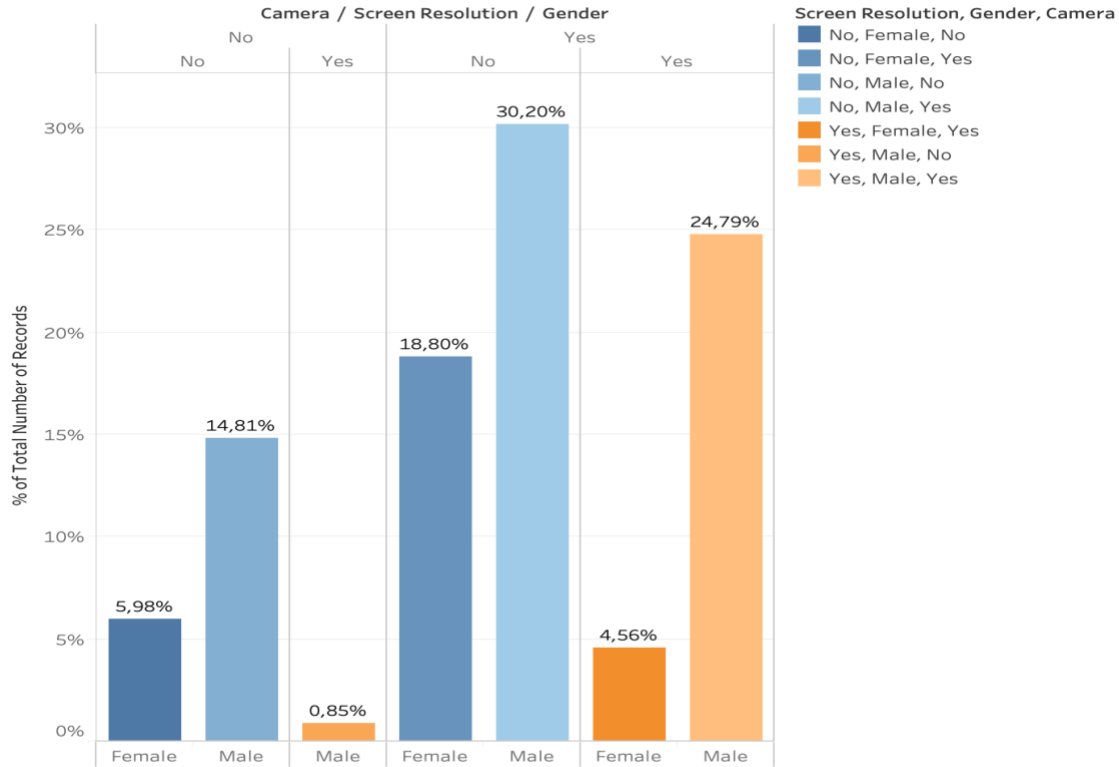
Age and Smartphone. Color shows details about Age. Size shows % of Total Number of Records. The marks are labeled by Age and Smartphone.

Gender vs Camera vs Battery Life



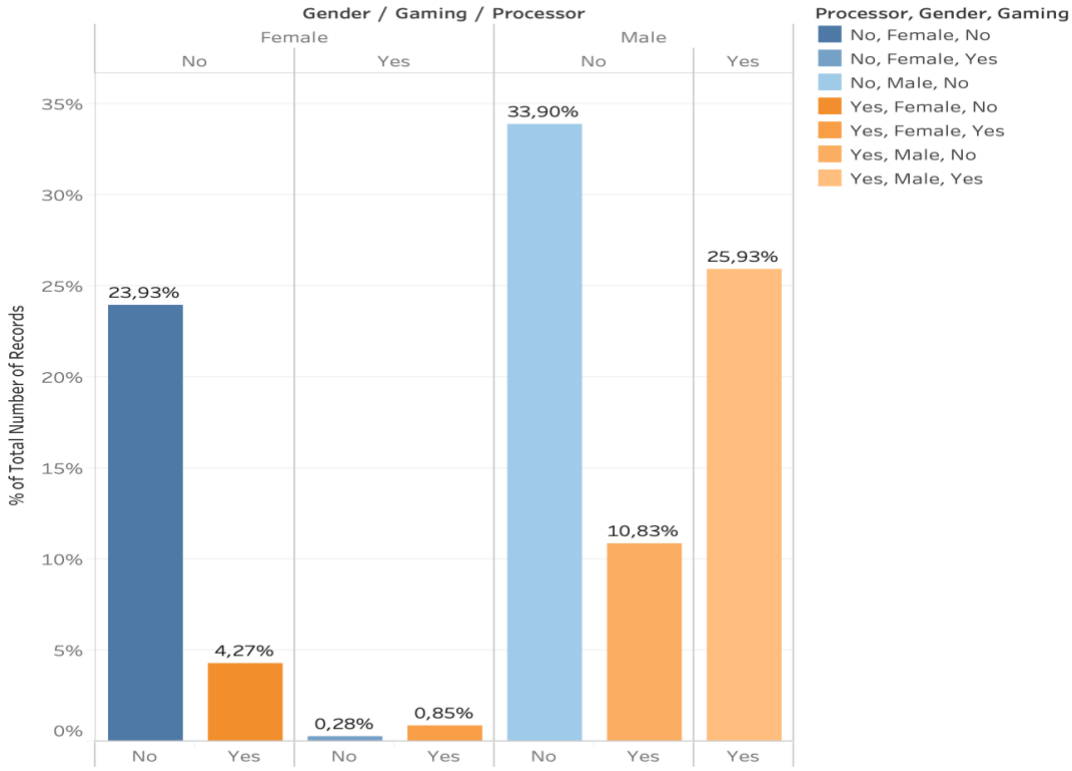
% of Total Number of Records for each Gender broken down by Camera and Battery Life. Color shows details about Gender, Battery Life and Camera. The marks are labeled by % of Total Number of Records.

Gender vs Camera vs Screen Resolution



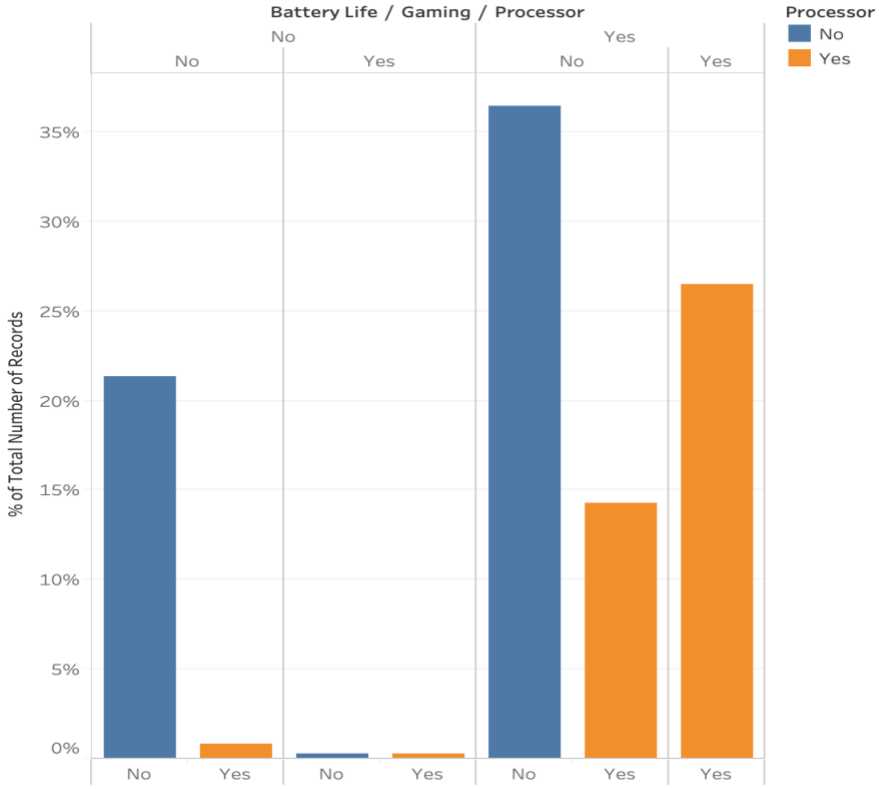
% of Total Number of Records for each Gender broken down by Camera and Screen Resolution. Color shows details about Screen Resolution, Gender and Camera. The marks are labeled by % of Total Number of Records.

Gender vs Gaming vs Processor



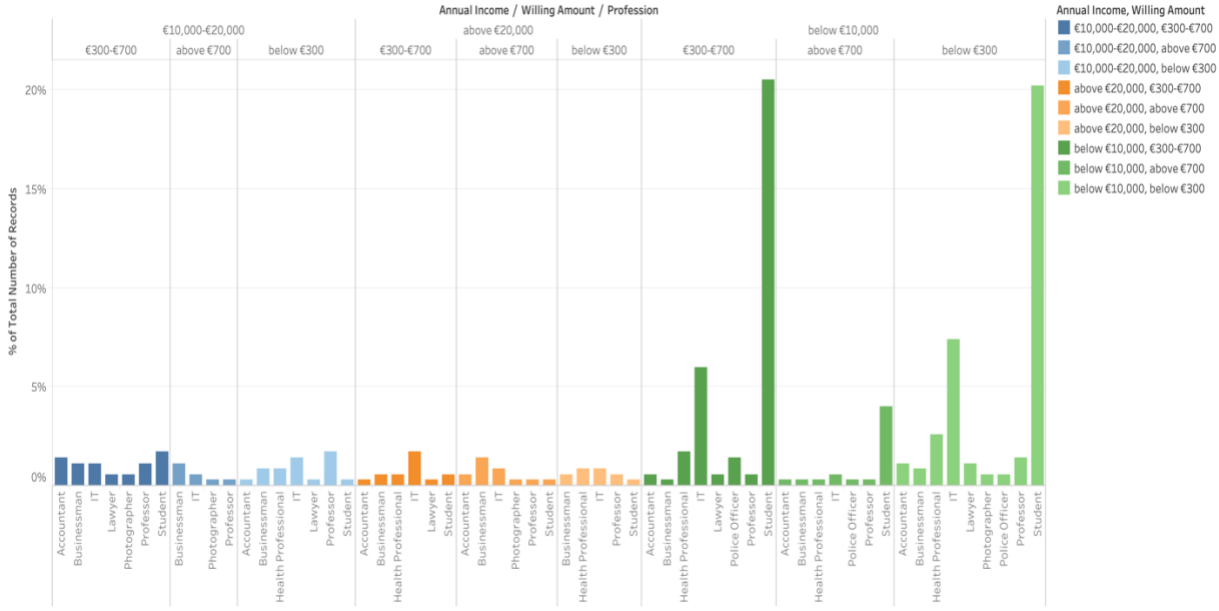
% of Total Number of Records for each Processor broken down by Gender and Gaming. Color shows details about Processor, Gender and Gaming. The marks are labeled by % of Total Number of Records.

Processor vs Bettery Life vs Gaming



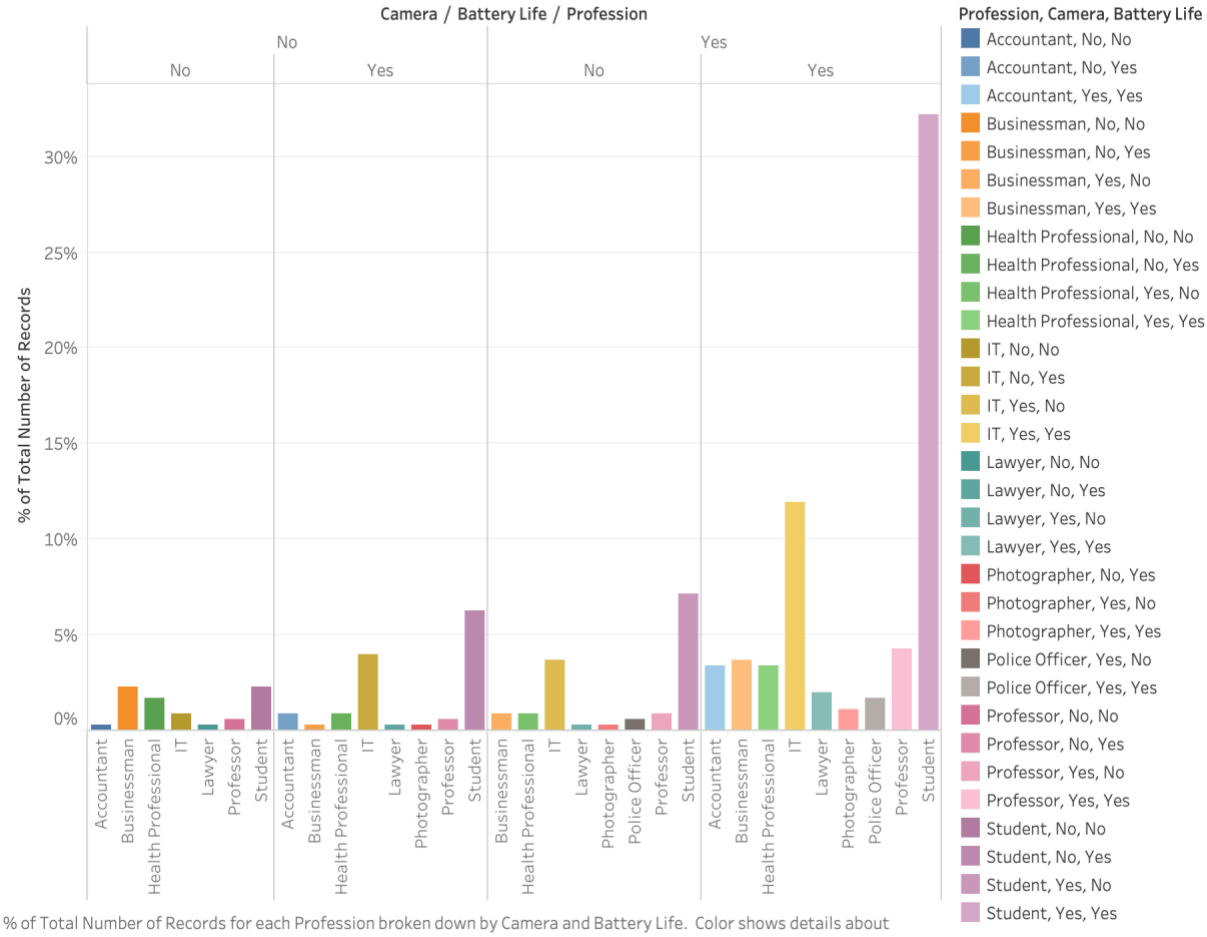
% of Total Number of Records for each Processor broken down by Battery Life and Gaming. Color shows details about Processor.

Profession vs Annual Income vs Willing Amount



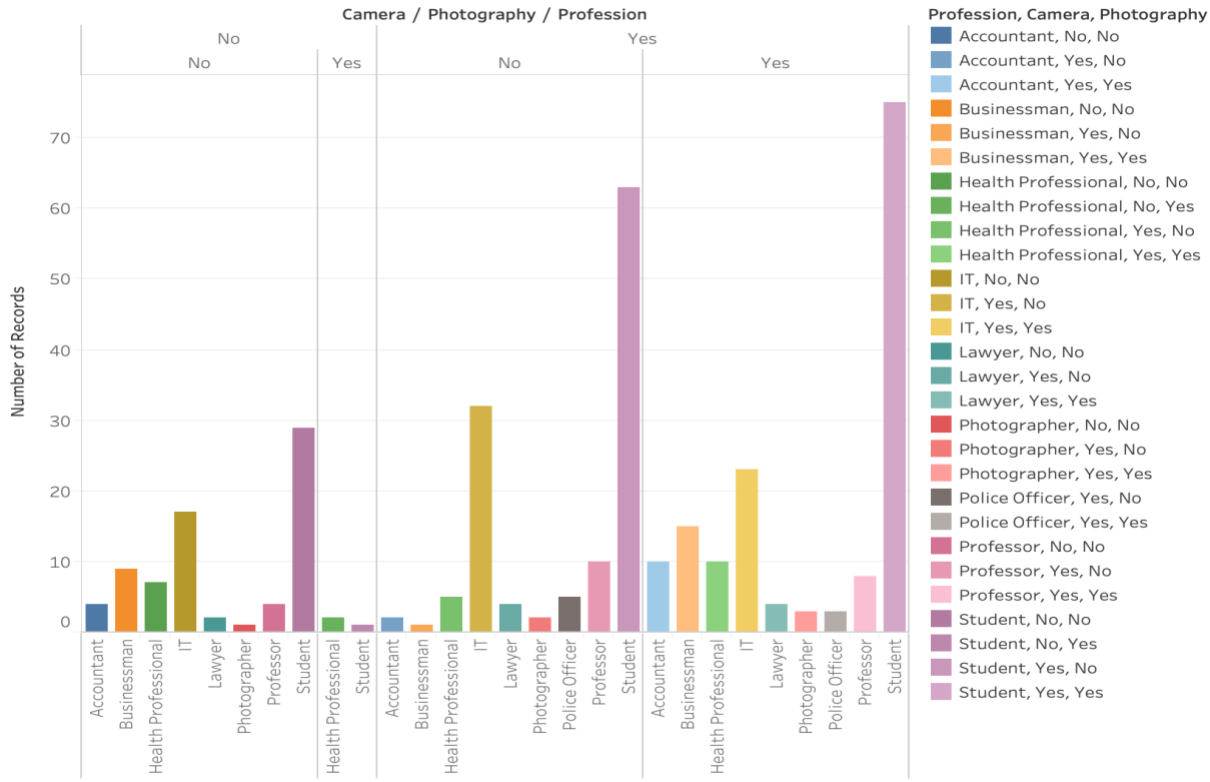
% of Total Number of Records for each Profession broken down by Annual Income and Willing Amount. Color shows details about Annual Income and Willing Amount.

Profession vs Camera vs Battery Life



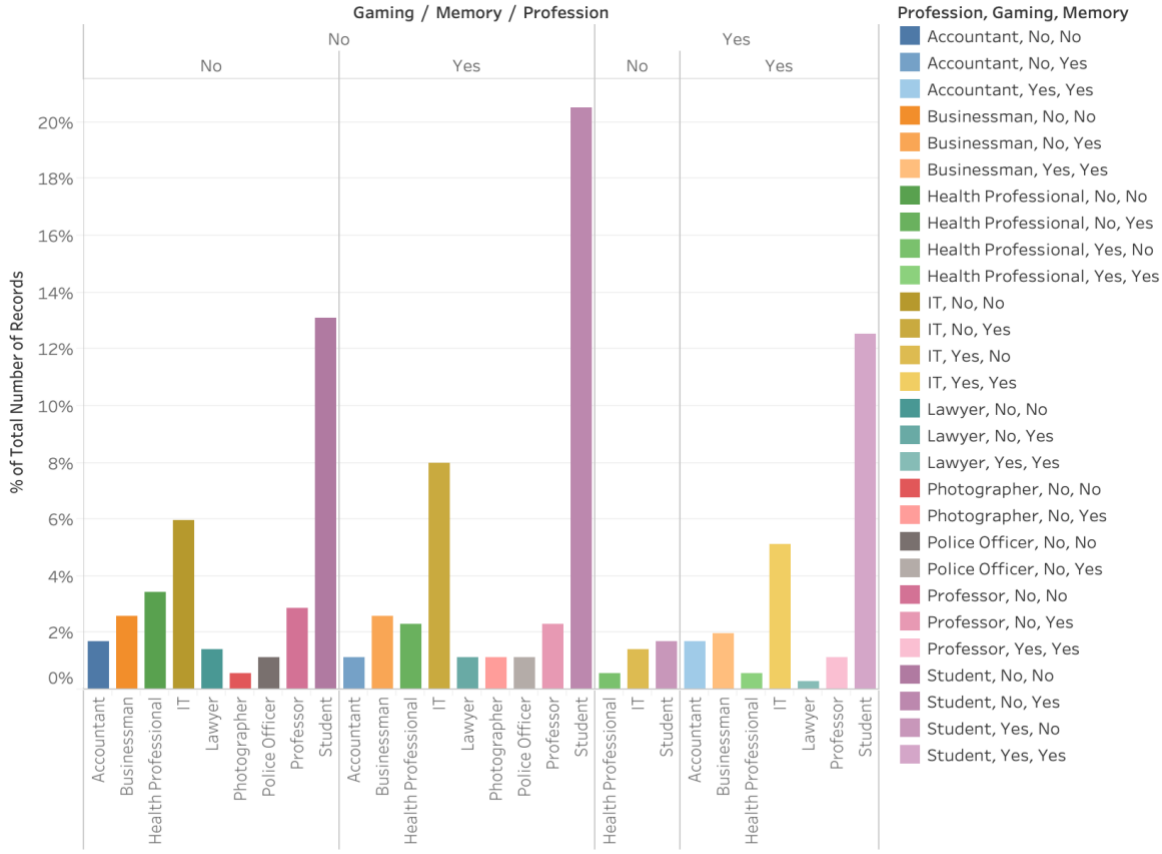
% of Total Number of Records for each Profession broken down by Camera and Battery Life. Color shows details about Profession, Camera and Battery Life.

Profession vs Camera vs Photography



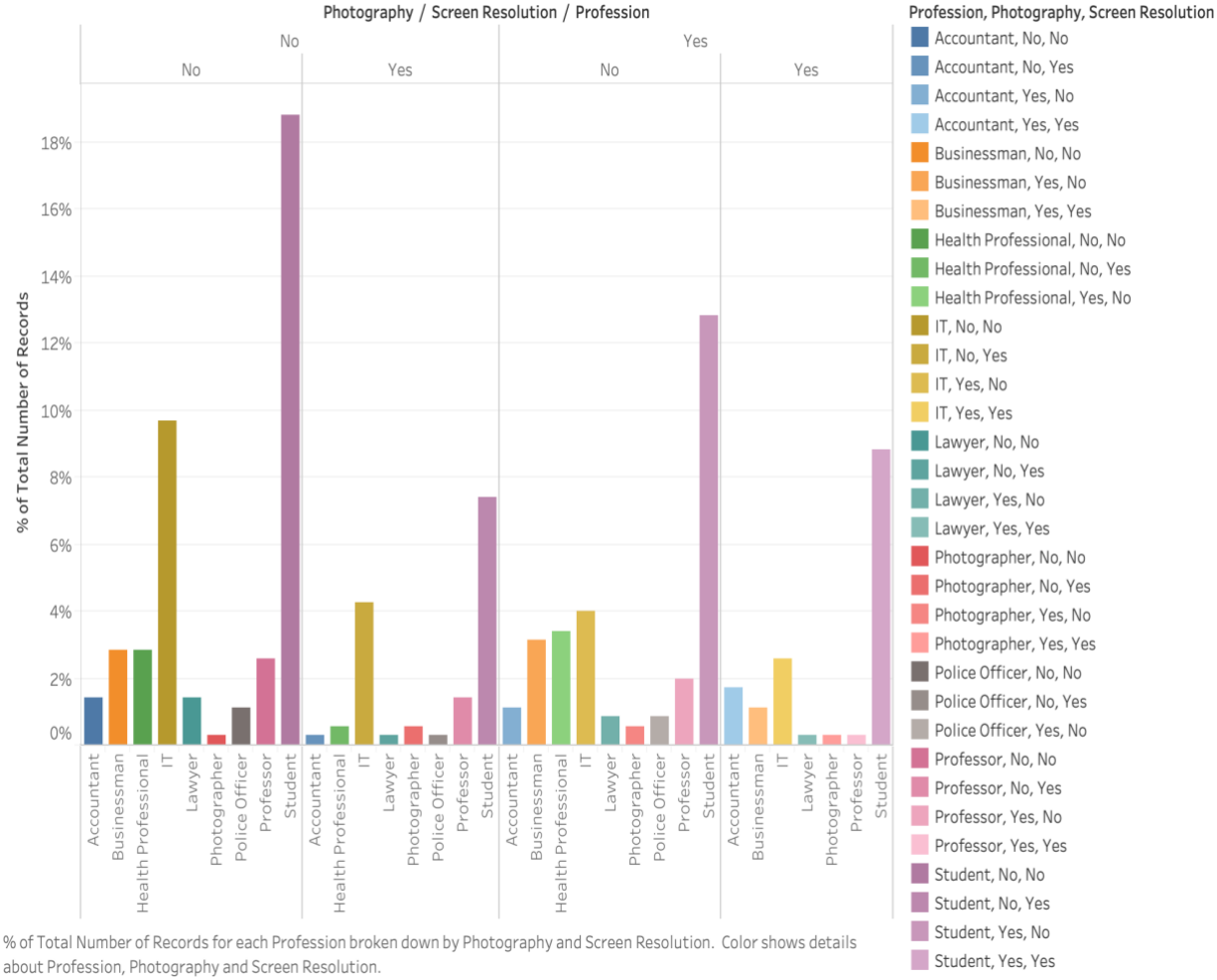
Sum of Number of Records for each Profession broken down by Camera and Photography. Color shows details about Profession, Camera and Photography.

Profession vs Gaming vs Memory



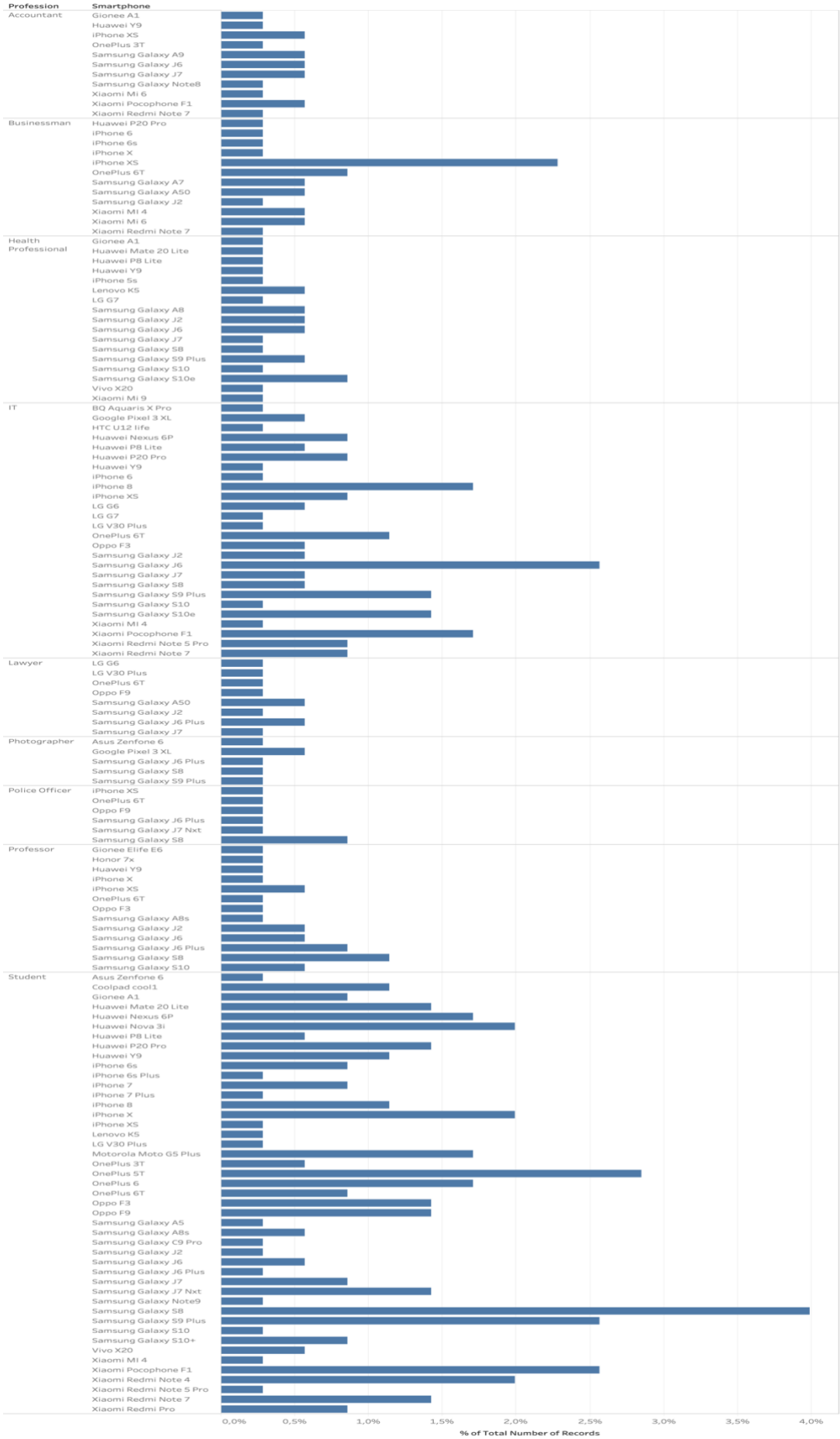
% of Total Number of Records for each Profession broken down by Gaming and Memory. Color shows details about Profession, Gaming and Memory.

Profession vs Photography vs Screen Resolution



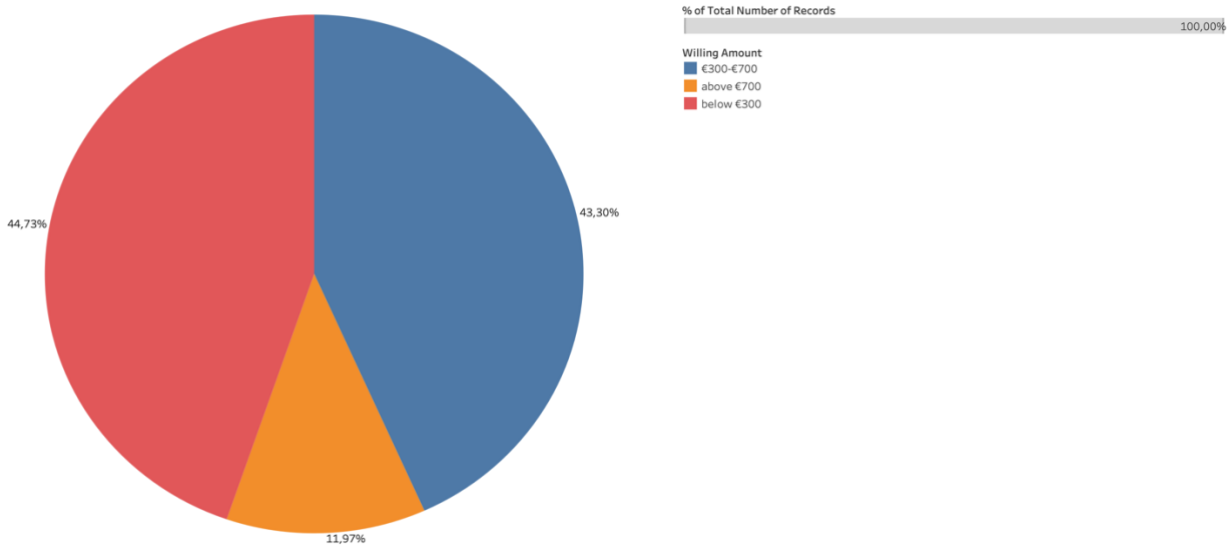
% of Total Number of Records for each Profession broken down by Photography and Screen Resolution. Color shows details about Profession, Photography and Screen Resolution.

Profession vs Smartphone



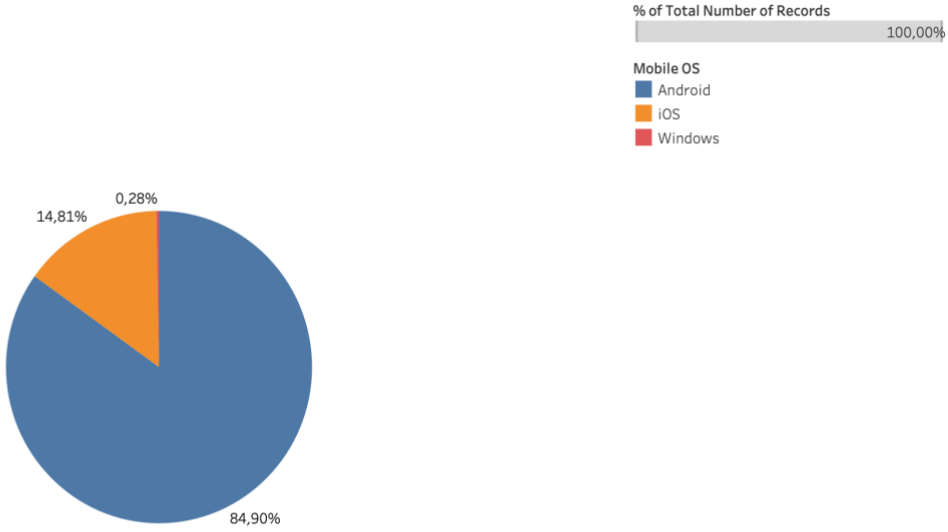
% of Total Number of Records for each Smartphone broken down by Profession.

Willing Amount to pay



% of Total Number of Records. Color shows details about Willing Amount. Size shows % of Total Number of Records. The marks are labeled by % of Total Number of Records.

Mobile OS Pie chart



% of Total Number of Records. Color shows details about Mobile OS. Size shows % of Total Number of Records. The marks are labeled by % of Total Number of Records.

Smartphone

Smartphone. Size shows % of Total Number of Records. The marks are labeled by Smartphone.

```
#function to train the recommendation system
def TrainRecommendationSystem(self):
    #Adding input layer and ist hidden layer
    self.classifier.add(Dense(output_dim = 45,init = 'uniform', activation= 'relu',input_dim = 28))
    #2nd hidden layer
    self.classifier.add(Dense(output_dim = 45,init = 'uniform', activation= 'relu'))
    #adding the output layer
    self.classifier.add(Dense(output_dim = 63,init = 'uniform', activation= 'softmax'))
    #compiling ANN
    self.classifier.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])
    #training
    self.classifier.fit(self.X_train,self.y_train,batch_size = 10, epochs = 100)
    self.system_trained = True
    print('Training complete')

    #function to recommend smartphones
def RecommendSmartphones(self,recommend_input):
    print(type(recommend_input))
    major_recommendation = self.classifier.predict(recommend_input)
    max_val = major_recommendation.max(axis = 1)
    result = nmp.where(major_recommendation == max_val)
    recommended_phone = self.y.columns[nmp.asscalar(result[1])]
    return recommended_phone


def Recommend(self,recommend_input):
    if(self.system_trained):
     return   self.RecommendSmartphones(recommend_input)

    else:
        self.TrainRecommendationSystem()
        return self.RecommendSmartphones(recommend_input)
```

```
def EncodeInputs(self,gender,age,annual_income,willing_amt,profession,mobile_os,main_reason,gaming,photography,camera,battery,processor,memory,screen_res):
    gender_inp_encoded = self.convert(self.data_encoder.GetGenderEncoded(gender))
    age_inp_encoded = self.convert(self.data_encoder.GetAgeEncoded(age))
    annual_income_inp_encoded = self.convert(self.data_encoder.GetAnnualIncomeEncoded(annual_income))
    willing_amt_inp_encoded = self.convert(self.data_encoder.GetWillAmtEncoded(willing_amt))
    profession_inp_encoded = self.convert(self.data_encoder.GetProfessionsEncoded(profession))
    mobile_os_inp_encoded = self.convert(self.data_encoder.GetMobileOsEncoded(mobile_os))
    main_reason_inp_encoded = self.convert(self.data_encoder.GetMainReasonEncoded(main_reason))
    #purpose inps
    gaming_inp_encoded = self.convert(self.data_encoder.GetGamingEncoded(gaming))
    photgrphy_inp_encoded = self.convert(self.data_encoder.GetPhotographyEncoded(photography))
    #features inputs
    camera_inp_encoded = self.convert(self.data_encoder.GetCameraFeatureEncoded(camera))
    battery_life_inp_encoded = self.convert(self.data_encoder.GetBatteryLifeEncoded(battery))
    processor_inp_encoded = self.convert(self.data_encoder.GetProcessorEncoded(processor))
    memory_inp_encoded = self.convert(self.data_encoder.GetMemoryEncoded(memory))
    screen_res_inp_encoded = self.convert(self.data_encoder.GetScreenResolutionEncoded(screen_res))
    #encoded input pass it to the predictor
    input_encoded = nmp.asmatrix(nmp.concatenate([gender_inp_encoded,age_inp_encoded,annual_income_inp_encoded,willing_amt_inp_encoded,
                                        profession_inp_encoded,mobile_os_inp_encoded,main_reason_inp_encoded,
                                        gaming_inp_encoded,photgrphy_inp_encoded,camera_inp_encoded,
                                        battery_life_inp_encoded,processor_inp_encoded,memory_inp_encoded,
                                        screen_res_inp_encoded]))
    return input_encoded
```

```
def GetSmartPhoneDisplayInfo(self,phone_name):
    return self.GetSmartphoneData(phone_name,self.smartphone_attributes.DISPLAY_INFORMATION)

def GetSmartPhoneOS(self,phone_name):
    return self.GetSmartphoneData(phone_name,self.smartphone_attributes.OS)

def GetSmartPhoneOSTag(self,phone_name):
    return self.GetSmartphoneData(phone_name,self.smartphone_attributes.OS_TAG)

def GetSmartPhoneRAM(self,phone_name):
    return self.GetSmartphoneData(phone_name,self.smartphone_attributes.RAM)

def GetSmartPhoneStorage(self,phone_name):
    return self.GetSmartphoneData(phone_name,self.smartphone_attributes.STORAGE)

def GetSmartPhonePrice(self,phone_name):
    return self.GetSmartphoneData(phone_name,self.smartphone_attributes.PRICE)

def GetSmartPhonePriceTag(self,phone_name):
    return self.GetSmartphoneData(phone_name,self.smartphone_attributes.PRICE_TAG)

def GetSmartPhoneBattery(self,phone_name):
    return self.GetSmartphoneData(phone_name,self.smartphone_attributes.BATTERY)

def GetSmartPhoneBatteryTag(self,phone_name):
    return self.GetSmartphoneData(phone_name,self.smartphone_attributes.BATTERY_TAG)

def GetSmartPhoneSelfieCamera(self,phone_name):
    return self.GetSmartphoneData(phone_name,self.smartphone_attributes.SELFIE_CAMERA)

def GetSmartPhoneMainCamera(self,phone_name):
    return self.GetSmartphoneData(phone_name,self.smartphone_attributes.MAIN_CAMERA)

def GetSmartPhoneMainCameraTag(self,phone_name):
    return self.GetSmartphoneData(phone_name,self.smartphone_attributes.MAIN_CAMERA_TAG)

def GetSmartPhoneImage(self,phone_name):
    return self.GetSmartphoneData(phone_name,self.smartphone_attributes.IMAGE)
```

```
global major_recommendation

major_recommendation = recommendation_system.Recommend(input_encoders.EncodeInputs(gender_inp,age_inp,annual_income_inp,
                                                       willing_amt_inp,profession_inp,mobile_os_inp,
                                                       main_reason_inp,gaming_inp,photgrphy_inp,
                                                       camera_inp,battery_life_inp,processor_inp,memory_inp,
                                                       screen_resolution_inp))

major_recomm_os = smartphone_information.GetSmartPhoneOS(major_recommendation)
major_recomm_screen = smartphone_information.GetSmartPhoneDisplayInfo(major_recommendation)
major_recomm_display_size = smartphone_information.GetSmartPhoneDisplaySize(major_recommendation)
major_recomm_RAM = smartphone_information.GetSmartPhoneRAM(major_recommendation)
major_recomm_memory = smartphone_information.GetSmartPhoneStorage(major_recommendation)
major_recomm_selfie_camera = smartphone_information.GetSmartPhoneSelfieCamera(major_recommendation)
major_recomm_main_camera = smartphone_information.GetSmartPhoneMainCamera(major_recommendation)
major_recomm_battery = smartphone_information.GetSmartPhoneBattery(major_recommendation)
major_recomm_price = smartphone_information.GetSmartPhonePrice(major_recommendation)
major_recomm_url = smartphone_information.GetSmartPhoneImage(major_recommendation)

global recommendation_alternatives
recommendation_alternatives = more_recommendation.GetMoreRecommendations1(major_recommendation)

phone1_name = recommendation_alternatives[0]
phone1_url = smartphone_information.GetSmartPhoneImage(phone1_name)
phone1_price_in_euro = smartphone_information.GetSmartPhonePrice(phone1_name)

phone2_name = recommendation_alternatives[1]
phone2_url = smartphone_information.GetSmartPhoneImage(phone2_name)
phone2_price_in_euro = smartphone_information.GetSmartPhonePrice(phone2_name)

phone3_name = recommendation_alternatives[2]
phone3_url = smartphone_information.GetSmartPhoneImage(phone3_name)
phone3_price_in_euro = smartphone_information.GetSmartPhonePrice(phone3_name)

phone4_name = recommendation_alternatives[3]
phone4_url = smartphone_information.GetSmartPhoneImage(phone4_name)
phone4_price_in_euro = smartphone_information.GetSmartPhonePrice(phone4_name)

print("recommendation complete")
```

```python
@app.route('/Recommendation_System#home')

def home():
    return render_template('index.html#home')

@app.route('/Recommendation_System#about')

def about():
    return render_template('index.html#about')

@app.route('/Recommendation_System#services')

def service():
    return render_template('index.html#services')

@app.route('/Recommendation_System#team')

def team():
    return render_template('index.html#team')

@app.route('/Recommendation_System#contact')

def contact():
    return render_template('index.html#contact')

@app.route('/visualization')

def visualization():
    return render_template('visualization.html')

@app.route('/Other_recomm1')
```

```python
ACCOUNTANT = "Accountant"
BUSINESSMAN = "Businessman"
HEALTH_PROFESSIONAL = "Health Professional"
IT = "IT"
LAWYER = "Lawyer"
PHOTOGRAPHER = "Photographer"
POLICE_OFFICER = "Police Officer"
PROFESSOR = "Professor"
STUDENT = "Student"

ACCOUNTANT_CODE = [0,0,0,0,0,0,0,0]
BUSINESSMAN_CODE = [1,0,0,0,0,0,0,0]
HEALTH_PROFESSIONAL_CODE = [0,1,0,0,0,0,0,0]
IT_CODE = [0,0,1,0,0,0,0,0]
LAWYER_CODE = [0,0,0,1,0,0,0,0]
PHOTOGRAPHER_CODE = [0,0,0,0,1,0,0,0]
POLICE_OFFICER_CODE = [0,0,0,0,0,1,0,0]
PROFESSOR_CODE = [0,0,0,0,0,0,1,0]
STUDENT_CODE = [0,0,0,0,0,0,0,1]
```

Annex

Running the project:

The project was done using the Spyder IDE (Integrated Development Environment) in Anaconda.

Please install anaconda in the system before running this project.

open the anaconda

on the anaconda, navigator go to environments

you will see base(root) and play button click that and click open terminal

using that terminal

install pandas and numpy first

pip install pandas

pip install numpy

then install keras, tensorflow, theano

pip install keras

pip install tensorflow

pip install theano

installing firebase

pip install python-firebase

pip install sklearn

For the project to work correctly, we need to configure firebase files as well.

Go inside the Anaconda installed folder and then follow as Anaconda3\Lib\site-packages\firebase

open the _init.py_ file and change .async to .async_

rename the .async file to .async_

open the firebase file and there also change the .async to .async_

after going through all these steps you can run your project

just make sure that while running the code from spyder you are running the app.py file and all the files are present in your current working directory

Declaration of Authenticity

We, Bhuwan Acharya, Sudarshan Awasthi, Bikash Paudel, Anup Katuwal, Musifique Ahmed Isha and Md. Mansoor Ali, hereby declare that the work presented herein is my own work completed without the use of any aids other than those listed. Any material from other sources or works done by others has been given due acknowledgment and listed in the reference section. Sentences or parts of sentences quoted literally are marked as quotations; identification of other references with regard to the statement and scope of the work is quoted.

The work presented herein has not been published or submitted elsewhere for assessment in the same or a similar form. We will retain a copy of this assignment until after the Board of Examiners has published the results, which we will make available on request.