Bachelor's thesis

Information and CommunicationsTechnology

2020

Nam Pham

Thao Ho

VOICE ANALYSIS WITH PYTHON AND REACT NATIVE

TURKU AMK
TURKU UNIVERSITY OF
APPLIED SCIENCES

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information and Communications Technology

April 2020 | **54**

Nam Pham; Thao Ho

VOICE ANALYSIS WITH PYTHON AND REACT NATIVE

ABSTRACT

Machine Learning and Mobile Development have recently drawn great attention amongst scientists and developers. In 2018, Finland alone had 81% of men and 79% of women over the population owning a smartphone for own use. The two areas from Artificial Intelligence – Natural Language Processing and Computer Vision, where a machine can understand human language and see images respectively, were considered. Hence, the objectives of the thesis were to let the machine predict the personal preference of user depending on the provided input. On top of that, developing a good model, which means the accuracy of machine prediction must be relatively above the average for classification, and deciding on which smartphone operating system was discussed based on personal skills and interests. Different methods such as conducting a survey, performing an interview, programming in React Native – an iOS-friendly developing system for Apple's product were carried out. As a result, the application – Funny Dino – can predict the characteristics given the prompted input from users. Furthermore, the authors held responsibilities for each side of the development – Thao Ho for the front-end, where the user and machine will communicate, and Nam Pham for the back-end, where all data were processed.

KEYWORDS:

Natural language processing, computer vision, machine learning, model, React Native, Python, Flask.

CONTENTS

APPENDICES

FIGURES

TABLES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
App	Application
BOW	Bag of words
CV	Computer Vision
KNN	K-nearest neighbours clustering
ML	Machine Learning
NB	Naive Bayes
NLP	Natural Language Processing
SoMe	Social Media
TF-IDF	Term frequency – inverse documents frequency
UI	User interface

1 INTRODUCTION

Machine Learning and Mobile Development have become much discussed topics recently. In the United States of America, a substantial majority of the Americans (81%) is now smartphone owners, according to the research by Pew Research Center (Pew Research Center, n.d.). The spiking number demonstrates a great demand for mobile developers in the near future. That means that understanding a huge amount of data generated by users in real-time as well as taking advantage of it is presently a fruitful market.

As a result, the project was carried out to understand more about data handling in two interesting areas of AI – NLP and CV. The goal of this is to provide user with information regarding personality preference and entertaining experiences while experimenting the app. Specifically, the application focuses on the use of React Native (front-end), one of the most popular platforms to build a mobile app, and Python (back-end), a modern programming language on which most of the modern ML libraries have been built. The development of both front-end and back-end will be discussed thoroughly to help readers follow the whole process of building this app.

Even though the study of formality of speech has been conducted earlier by Heylighen et al. in 1999 (Heylighen & Dewaele, 1999), it is believed that there exists a correlation between a set of characterstics and the tendency of language used when speaking. Therefore, the MBTI test (The Myers & Briggs Foundation, n.d.) – a well-known psychology and characteristic test – has been consulted in this case.

The thesis has two authors, who wrote most parts of the thesis jointly. However since Thao Ho was responsible for the front-end, and Nam Pham for the back-end, each author wrote individually the respective implementation section

2 LITERATURE

2.1 Big Data

2.1.1 History of Big Data

The concept of big data dates back to the 1960s and 1970s when the first data centres were built and people began to pay attention to the development of the relational database.

Around 2005, with the explosion of online services such as Youtube, Facebook, Google, people started to realise the massive amount of data users generated . Hadoop was also developed in the same year, as well as the popularity consideration for NoSQL.

Thanks to Hadoop and recently Spark – the two most famous frameworks that are run based on the concept of distributed system, the work of data handling has been facilitated. As a result, the volume itself has grown exponentially. Surprisingly, it is not only humans who are generating thse massive data but also the machines with the advent of Internet of Things (IoT).

IoT is the term defined how objects and devices – connected to the internet – perform assignments based on the streamed and processed data. At the same time, every command users make will be sent back to the cloud for the sake of development.

The benefits of big data to data analytics are:

- Big data makes it possible to yield more complete answers due to a large sample of information.
- Big data offers analysts more confidence in experimenting, or trying out different approaches to tackle problems.

Most of the data collected are flawed and unstructured. Therefore, the thorough study of big data alongside machine learning experience is mandatory to interprete the patterns and symptoms of such large dataset.

2.1.2 Important features of Big Data

As data contributes to the learning process of a machine as the base knowledge, it is of importance for the machine to obtain a relatively sufficient amount of worthable data such that it can analyse and grasp the starting knowledge to perform the tasks as desired. Relying on one source of information can produce bias outcome, which may lead to false anticipation and unsatisfied user-experience.

According to Oracle official wesite (Oracle., n.d.), the main features that add up to a quality dataset appear to be: volume, velocity, and variety. In the context of this thesis, learning the definitions will help to enrich the database as well as to practice other skills such as: determining which type and source of data would be beneficial to analysing, curating raw data, and applying ML techniques.

Volume

The amount of data matters. However, most of data available on in internet are useless, low-density and unstructured at first.

Velocity

This refers to the rate at which data is received and acted on. Nowadays, smart products, such as Alexa, require real-time or near real-time data evaluation and operation in order to respond to users at the same time.

Variety

It consists of many types of data that are available: traditional and non-traditional. Traditional data are those structured and fit neatly in a relational database. The other one is either unstructured or semistructured due to the rise of big data. Examples of non-traditional data can be found easily as text, audio, or video. Thus, they require additional processing, or pre-processing, to derive meaningful interpretation.

2.1.3 Usage of Big Data

Big Data in this thesis (information from pictures and from text mining) is mostly used for ML which anticipates the characteristics of users. Given the base knowledge, the probability will be calculated so that users' inputs will be computed and compared to predict their communication styles. If the case is negative, the users can decide to correct it. Such behaviour allows the machine to improve the accuracy of the statitics itself through the continuous data accumulation.

As mentioned, another purpose of Big Data is to enhance customer experience. By letting users be able to correct the result themselves should the calculation is incorrect, on one hand, it increases the human-computer interaction. On the other hand, it collects a valuable amount of data for developers to improve and develop the application in the future. The data collected can be used for further analytical goals such as: marketing, market-demand analytics, predictive maintenance, and product suggestion by ML.

2.2 Natural Language Processing

Natural Language Processing (NLP) is a subfield of AI and linguistic studies, where the machine is able to manipulate automatically the human language, for example speech, text. These arealso the main sources where data are collected for the thesis work.

2.2.1 Natural Language Processing in modern definition

NLP is viewed as linguistic science:

> “The aim of a linguistic science is to be able to characterize and explain the multitude of linguistic observations circling around us, in conversa-tions, writing, and other media. Part of that has to do with the cognitive side of how humans acquire, produce and understand language, part of it has to do with understanding the relationship between linguistic utterances and the world, and part of it has to do with understanding the linguistic structures by which language comminucates.” (Manning & Schutze, 1999).

As discussed that mathematics has been applied in languages as well as the dominance of statistical methodologies, language is now processed by being encoded and regarded as a large distrubition. Statistics will then be executed in accordance with some unknown probability distribution to find out the likelihood in which a sample will be classified.

Goldberg in Neural Network Methods in Natural Language Processing (2017) (Goldberg, 2017) also succinctly stated that NLP indicates the automation of computational processing of human languages. That considers languages which are processed with advanced algorithms, and then outputs the natural looking text(s).

The Python NLTK library for NLP is a a good example of computers manipulating natural languages. On one hand, one can simply view the problem as counting the frequency of words to distinguish writing styles. On the other hand, NLP is applied to understand human sayings, or at least to generate meaningful, useful responses.

2.2.2 Applications of Natural Language Processing

Some applications of NLP used in the thesis are: Automatic Translation, Sentiment Analysis, Text Classification.

Automatic Translation

When collecting the data for the work, sources of different news have been taken into consideration: news in Norwegian, news in Finnish, and news in Vietnamese. All these texts will be translated into English for the vast majority of readers. Instead of translating each sentence, they will be put in a csv file (Excel file). Once the work is done, they will be all converted by using Python with pandas library and Google Traslation API.

Sentiment Analysis

In order to avoid the extremism (too negative or too positive about a topic) which will bias greatly the final prediction, the dataset will be selected based on its sentiment. There are many APIs available in the market; however, the one used in the thesis is a Python-friendly: VADER (Valence Aware Dictionary sEntiment Reasoner). Hence, only those whose compound score in the range from -0.05 to 0.05 will be kept to remain the neutrality (cjhutto, n.d.).

Text Classification

This is the most latent application. The usage of this can be found easily in Email services, where all the texts will be processed and classified before Text Classificationdelivers an appropriate section (mailbox, spam, junk, etc.). For this purpose, tn the thesis context, each of the data will be randomised classified as 0 and 1, meaning whether it is useful or not.

2.3 Computer Vision

Computer Vision (CV) is the science that trains a machine to see objects by developing computational algorithms to extract useful multi-dimensional data from images and videos (Techpedia, 2019). Specifically, CV is a discipline under the area of AI that deploys the statistical technique from ML to build up, or compacted in devices such as: Google Lens, autonomous vehicles, face recognition.

From the neuroscience point of view, the final goal of CV is to construct computational models that is equivalent to the human visual systems. From the engineering point of view, CV is the study that demands scientists to come up with complex algorithms that can train a machine to stimulate some of the tasks the human visual systems perform, even to surpass the human ability in quite a few cases.

2.3.1 Applications of Computer Vision

CV is applied in the thesis with such applications as: Object Detection, Image Segmentation, and Image Retrieval. To grasp an easy understanding of those definitions, examples are written below.

Object Detection:

Object Detection is a more advanced algorithm than Image Classification. If the previous one only considers one object at a time, object detection requires the computer not only to draw a box to not one but as many objects detected as possible, but also to label every each of them.

Image Segmentation:

A further step to Object Detection is Image Segmentation whose algorithm will provide an exact outline to the object compared to a bounding box. This means that an image will be segmented into several groups of pixels, each of which will be classified and labelled appropriately.

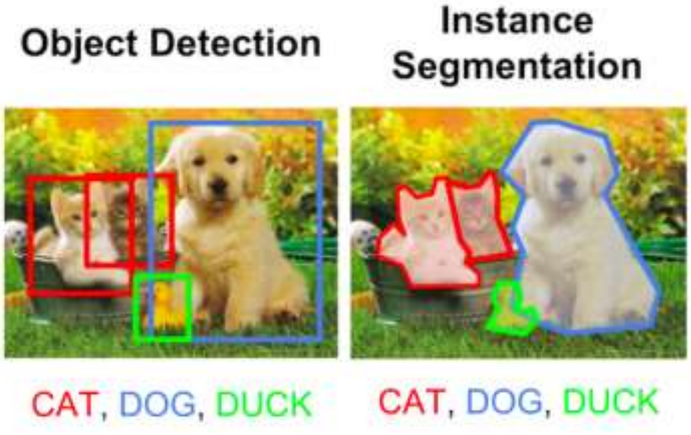


Figure 1. Object Detection and Instance Segmentation. (Source: clarifai.com)

Image Retrieval:

A machine will be prepared with a dataset from which it can learn all the important features in order to repair the corrupted ones.

2.4 React Native, ReactJS Framework:

2.4.1 Brief History of ReactJS

ReactJS, or React.js, is an open-source library built based on the JavaScript platform to perform data displaying with the assistance of HTML. The framework has been favoured exponentially by front-end communities around the world thanks to its user-friendly interface, feasibility in cross-platform programming. Facebook, of no surprise, is the author of this native platform.

With the emergence of big data, developers at Facebook, therefore, could not keep up with coding maintenance due to an increasing number of features in Facebook Ads app. These cascading updates urged Jordan Walke (Facebook Developer, 2015), a software engineer at Facebook, to construct a FaxJS prototype so that the maintenance could be done more efficiently.

In 2012, Facebook acquired another trending SoMe app – Instagram. The acquisition pressured on the company itself to turn React into an open-source platform so that both apps could be managed simultaneously. A year later, the open source was introduced in the JavaScript Conference US. Regardless of the scepticism in the first place, Facebook kept on presenting and integrating the platform to dominant programming languages: JSFiddle, Ruby, and Python (Ferenc, 2018) .

In 2014, ReactJS gained more attention from the public with the React Tour (#reactjsorldtour). Its ecosystem was contributed with: React Developer Tools as an extention of Chrome Developer Tools (Jan 2), ReactiveX.io in Apri, React Hot Leader plug-in in Jul 13, PlanOut – a language for online experiments in which React is included.

Since then, React has become one of the most liked platforms to build a native app. Statistics during the period 2015 – 2016 on Github indicated that: with 1002 contributors commited 7971 times in 45 branches with 124 releases, ReactJS already become the 14th most starred repository.

2.4.2 React Native, Pros and Cons:

a. React Native and React JS:

React Native was released in 2017, two years after ReactJS 2015 was released. While ReactJS focuses more on developing web interfaces, React Native is built as a hybrid app-development framework for iOS and Android. In React Native, the code can be shared, or reused, between iOS and Android, leaving the rest to designing platform-specific interfaces.

b. Pros of React Native:

Since React Native is built upon Java Script, it allows developers to learn and shift to this language easily, quickly. In addition to that, React Native inherits the advantages from ReactJS, which are: isolation ability – developers can update components separately without affecting others in the DOM (document object model) trees, reuse code components, and an open-source library. In addition, thanks to the prevalence of JS code, the task may be completed faster and more efficiently because all the updates can simply be seen by refreshing the viewing page. The aim is to maximise UI performance.

Compared to other frameworks such as PhoneGap, which reduces the performance of the application because the code is rendered via Webview, React Native performs better (altexsoft, 2018). It “communicates with targeted components for iOS or Android and renders code to native APIs directly and independently” (altexsoft, 2018).

Another component is the native modules. The modules, however, are written in Objective-C and Java out-of-the-box, which are at a higher computational level mostly used for meta data analysing. For these reasons, these modules cannot unfortunately be reused. The only solution is either applying the existing modules or writing customising modules in the aforementioned languages.

c. Cons of React Native:

As having been released recently, React Native is greatly lacking documentation, which may be hard for new users to experience the language. Furthermore, as the native modules were written in Objective-C – a high computational level programming language – the number of those are proficient in the above mentioned language is yet fairly small. In addition to that, the ecosystem – or the third-party components – is somehow limited.

According to the website Altexsoft, some of the shortlisted problems that have been reported by the engineers when working with React Native are: “hot reloading failures, incompatibilities between community libraries and different versions of React Native, emulator issues, problems with react-navigation, the need to frequently reinstall packages” (altexsoft, 2018). In short, React Native’s main disadvantages can be described as the instability and compatibility issues.

2.5 RESTful API services

2.5.1 Scope of RESTful API services

An API is an Application Programming Interface that uses HTTP requests to GET, PUT, POST and DELETE data (Rouse, 2019). It suggests a course of rules that let programs to talk to each other. Therefore, a developer builds the API on the server and permits the client to talk to it.

Additionally, REST is known as Representational State Transfer which determines what the API looks like. This includes a collection of guidelines developers must obey when generating their APIs. Some of these guidelines mention that users should be able to receive a piece of data (called a resource) when connecting to a particular URL. Every URL is described a request while the data return to the requester is called a response (Liew, 2018).

2.5.2 Standardisation approach

“In the REST architecture, clients send requests to retrieve or modify resources, and servers send responses to these requests” (Codecademy, n.d.). There are four basic HTTP methods to interact with resources in a REST system.

a. HTTP GET

This request is used to retrieve resource representation or an information from a server so the server looks for the data an user requested and sends it back to him. Moreover, a GET method is a default request method, which is also known as a READ operation. Assuming that the resource is found on the server, for any given HTTP GET API, a HTTP response code 200 (OK) must be returned, along with the response body, which is usually either XML or JSON content (Restfulapi.net, n.d.).

b. HTTP POST

It is used to generate a new subordinate resource so the server will create a new entry in the database and tell the requester whether the creation is successful. “If a resource has been created on the original server, the response should be HTTP code 201 response (Created) and contain an entity which describes the status of the request and refers to the new resource” (Restfulapi.net, n.d.).

c. HTTP PUT

This request is used to update existing resource on a server so the server updates an entry in the database and tells users whether the update is successful. If an existing resource is modified, the HTTP responded code should be 200 (OK) or 204 (No Content) (Restpfulapi.net, n.d.).

d. HTTP DELETE

This request is implemented to remove a resource from a server so the server deletes an entry in the collection of data and tells the user whether the removal is effective. “A successful response of DELETE requests SHOULD be HTTP response code 200 (OK) if the response includes an entity describing the status, 202 (Accepted) if the action has been queued, or 204 (No Content) if the action has been performed but the response does not include an entity” (Restfulapi.net, n.d.).

3 RESEARCH METHODS AND QUESTIONS

Questions arise as how all the knowledge that the authors have acquired, especially those during the exchange studies, can be applied to the application.

- Nam Pham – exchange at Ontario Tech University (former name: University of Ontario Institute of Technology – UOIT, Canada) – Data Science with courses in Big Data Analytics, Computer Vision, AI plus the introductory experience in NLP from the final internship at Valuer.ai in Copenhagen, Denmark.
- Thao Ho – exchange at Amsterdam University of Applied Sciences – Mobile Development with experience in building front-end with React Native alongside former programmings as full-stack engineer.

As a result, the project is divided into two parts: back-end (Python) with Pham and front-end (React Native) with Ho. The application is desired to check the communication style of users by applying ML with NLP and CV. The description of the app can be simply detailed as:

> In order to log in, the user will need to register with email and facial picture. Once submitted, the picture will be sent to the database and perform analysis to return the appropriate style (CV). If the result is negative, the user is able to conduct a questionnaire and generate a better result for the machine to learn. The User can also record his/her voice. Based on this, the analysis will be more accurate (NLP). The app should be able to run on both iOS and Android devices.

Below is the first sketch-up of the app:

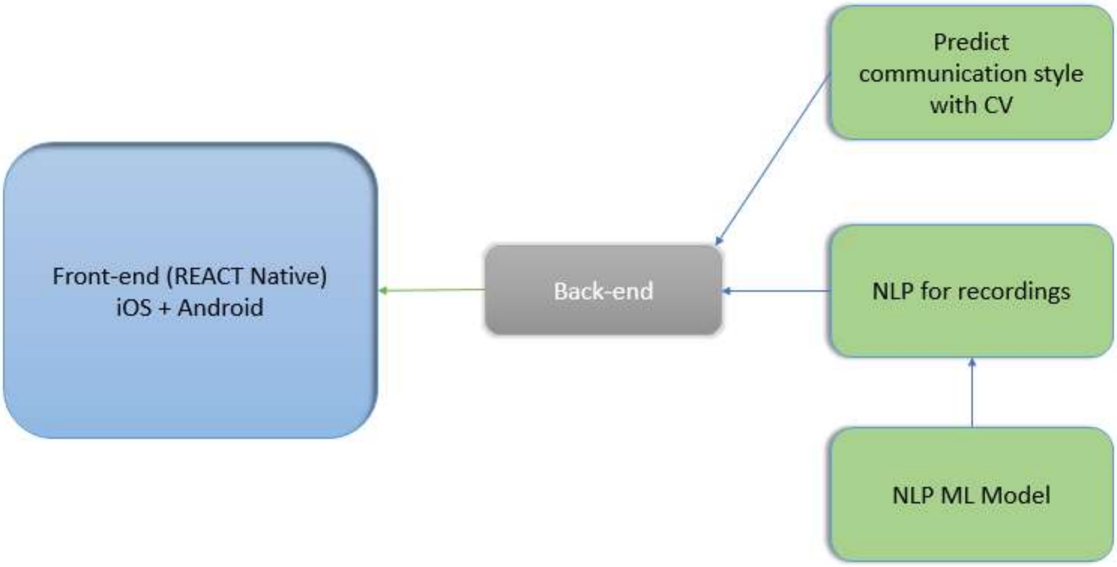

Predict communication style with CV
Front-end (REACT Native) iOS + Android
Back-end
NLP for recordings
NLP ML Model

Figure 2. Sketch-up of the process of building the application.

In Figure 2, the first task is to build a ML model based on the data collected through surveys and interviews. From that, because the app is intended to work with voice recording, assumptions were made in accordance with the results obtained from the data collection. Furthermore, as it is desired to be user-friendly as well as building up the data storage, CV was agreed to be a part of the back-end. All of them will be processed step by step before generating the outcome for front-end.

The research questions this thesis aims to answer are:

- What are the elements to determine the communication style?
- How can metadata (recording) be accessed?
- What data are required and how to build a ML model?
- How to classify and predict a picture?
- How can Python connect with REACT Native?
- How can visualisation be displayed in front-end?

3.1 Elements contributing to the communication style

Communication style is first believed to be whether it is of formal or informal style. In this regard, let us consult with the publication “Formality of Language: definition, measurement and behavioral determinants" by Helyghen and Dewaele, in 1999 (Helyghen & Dewaele, 1999).

3.1.1 Definition of Formality Speech

Due to the ambiguity characteristic of linguistic, there is almost no definition to a formal speech. However, a typical prototype of one is the language alongside the manners used in a trial, or in a presidential speech. In this regard, the opposite – the language used in a relaxed context such as chatting with friends – is considered informal.

According the Longman Dictionary of Applied Linguistics (Richards, Platt & Weber, 1987), formal speech is "the type of speech used in situations when the speaker is very careful about pronunciation and choice of words and sentence structure. This type of speech may be used, for example, at official functions, and in debates and ceremonies" (1987: 109). Unfortunately, the definition is yet vague because it focuses more on the formal contexts, correspondingly appropriate manners but not the choice of words itself. Another study by Labov (1972) and Tarone (1988) also noted that it was the voice and the modulation that makes the speech formal.

As a result, the term formality splits itself into two definition: *surface formality* with focus relies on convention and *deep formality*. The former one concentrates conventionally on creating a "formal situation" where a speaker is paid sufficient attention to through adequate manners and modulations. It also corresponds to the explanation from Oxford Leaner's Dictionary (Oxford Learner's Dictionary, n.d) as "a thing that you must do as part of an official process, but that has little meaning and will not affect what happens". This, however, does not carry any analytical meaning because of the linguistical ambiguity. Whereas the latter, *deep formality*, is believed to be more insightful for analytic since its attention is established by provoking unequivocal understanding of the usual meaning of an expression. Additionally, deep fomality is less dependent on the cultural background of a specific language itself.

3.1.2 Context-dependence

In spite of the logical terms and concepts (for example "if"-"then"-"else", "and", "or", etc.), natural languages are yet close enough to the logic of Calculus. Unless there exists a shared context, the vagueness will be minimised to the most. That is the finding published in the classic paper from Grice (1975) – “Logic and Conversion”.

The context dependency is clearly expressed in the saying: “I will talk to you soon”. If it can be rephrased formally to “Kathy will talk to John the police at 5 o’clock today”, the context will be disregarded and the full comprehension will be reached. More detailed, if there is a person witnessing the conversation between Kathy and John, s/he will apparently grasps the idea of to whom the pronouns “I”, “you” are addressed. However, for those who read the information, the first saying will be less comprehensive than the second one though they both convey the same idea.

The shared framework functions in the similar mechanism. Implicature – a term used when language is expressed explicitly to avoid ambiguity – is implicitly referring to the context when the language does not. An example of that would be when a person is in the middle of a conference, but then suddenly thinks “Perhaps I can get out and enjoy my coffee”. Albeit not literally stating, it is easily understood that the person feel it is boring to be in the conference so that s/he should go out, stop listening to the speech and start to enjoy a coffee.

3.1.3 Formality measurement

The methodology posed in the study is the combination of Part of Speech and Bag of Words. That says, for example, a corpus (sentence in this instance) will be broken down into tokens (separate words), each token will be assigned with an corresponding type (noun, verb, etc.), and then the frequency will be counted. By doing this, the average degree of deixis for categories of word will be calculated. This will reduce the anaphoric comprehension for both a computer and a linguistic, which means less context-dependent.

An example suggested by Heylighen and Dewaele (Heylighen & Dewaele, 1999) would be a sentence “Colonialists devastated the city”. A passive form will be obtained by dropping person deixis (“Colonialists”): “The city was devastated”. Normalisation is conducted by removing the time deixis, which means a noun must be put in use instead of the verb: “The devastation of the city”.

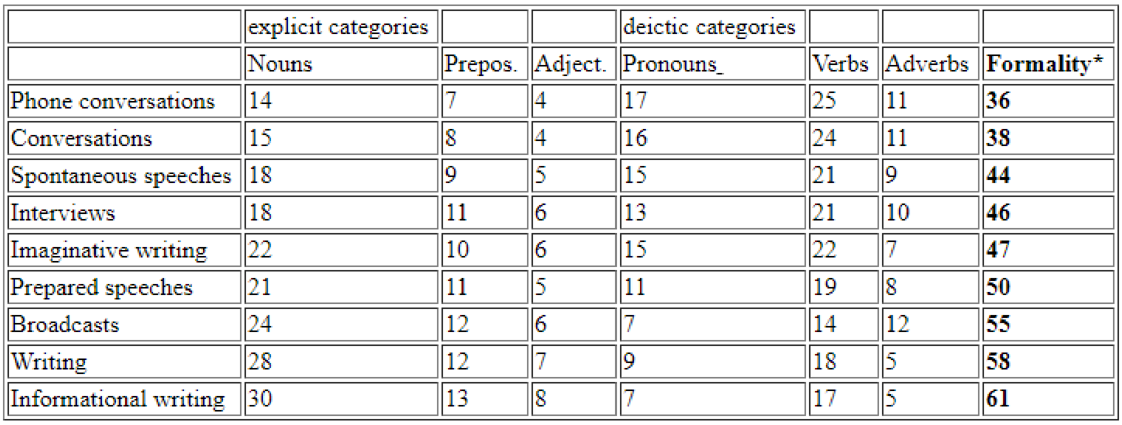
In conclusion, the formula for formality F suggested by Heylighen and Dewaele in 1999 is:

F = (noun frequency + *adjective frequency* + *preposition frequency* + *article frequency – pronoun frequency – verb frequency – adverb frequency – interjection frequency* + *100) / 2*

As a result, F will be within the range of 100 (from 0) percentage. The higher of the value F, the more formal the language is exercised.

Figure 3 and Table 1 below is the result of work done by the scientist Hudson in 1994 (Hudson, 1994) concerning the *formality** (the asterisk denotes the absence of article and interjection categories) of calculation in English language, which is of our interest.

Table 1. Formality* of English language. (Source: Heylighen & Dewaele, 1999)



	explicit categories			deictic categories			
	Nouns	Prepos.	Adject.	Pronouns_	Verbs	Adverbs	**Formality***
Phone conversations	14	7	4	17	25	11	**36**
Conversations	15	8	4	16	24	11	**38**
Spontaneous speeches	18	9	5	15	21	9	**44**
Interviews	18	11	6	13	21	10	**46**
Imaginative writing	22	10	6	15	22	7	**47**
Prepared speeches	21	11	5	11	19	8	**50**
Broadcasts	24	12	6	7	14	12	**55**
Writing	28	12	7	9	18	5	**58**
Informational writing	30	13	8	7	17	5	**61**

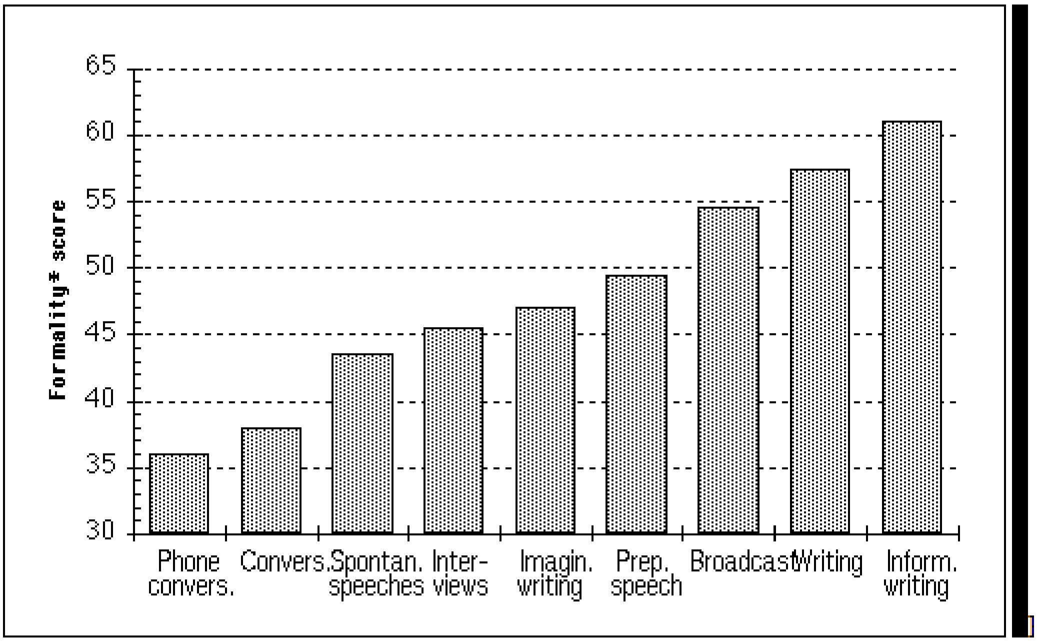
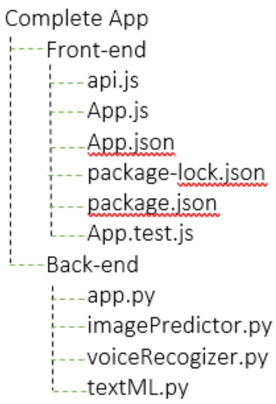


Figure 3: Formality* of English language in different fields.

The result of this research will be consulted later in the implementation.

4 IMPLEMENTATION

Below is the final project structure:



```
Complete App
|---Front-end
|    |---api.js
|    |---App.js
|    |---App.json
|    |---package-lock.json
|    |---package.json
|    └---App.test.js
└---Back-end
     |---app.py
     |---imagePredictor.py
     |---voiceRecogizer.py
     └---textML.py
```

Figure 4: Project final structure.

4.1 Front-end

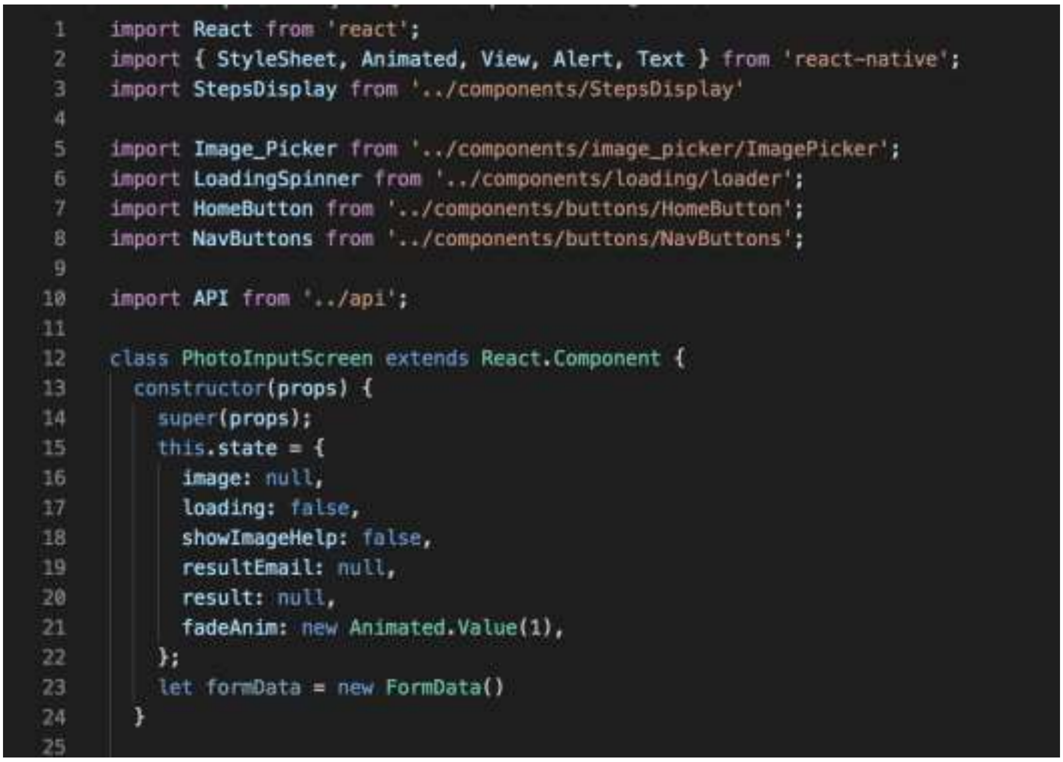
In the Front-end part, we have decided to create small components and put them in component folder because the smaller components could be reused elsewhere and is easier to understand. We also have stored our mocked data to data folder and different view such as Home Screen, Questionaire Screen and other screens to views folder. App.js file will call all components and connect them by using navigation. Moreover, to connect the back end and front end, we will use api url link in api.js file.

4.1.1 Code Review

Photo Input Screen, Recording Screen and DominanceQuestion Screen are three main important screens to get data from user to analyse exactly each user communication styles.

a. PhotoInput Screen

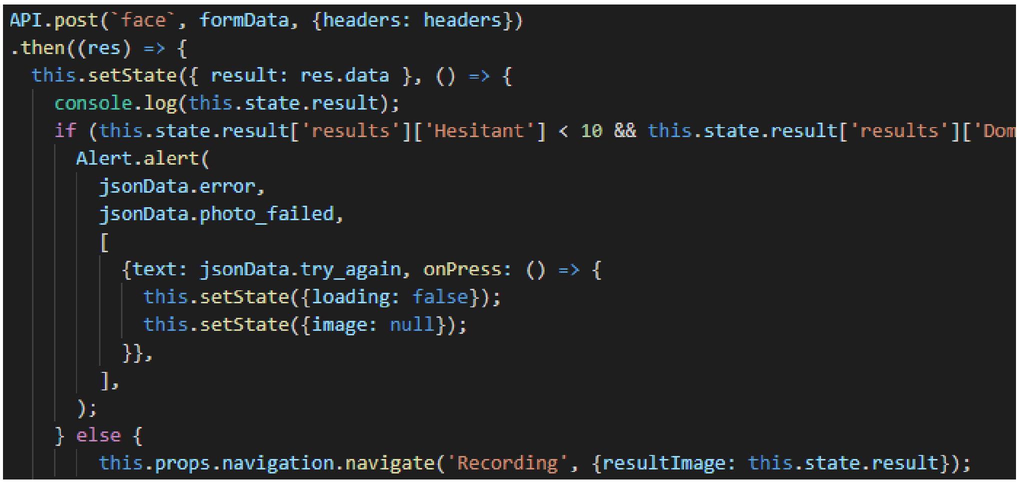
As Figure 5 indicates, there are many different components in Photo Input Screen such as Step Display, Image Picker, Loading Spinner, Home Button, and Navigation Button that was imported and rendered from the component folder in order to display them in the this screen.



```
import React from 'react';
import { StyleSheet, Animated, View, Alert, Text } from 'react-native';
import StepsDisplay from '../components/StepsDisplay'

import Image_Picker from '../components/image_picker/ImagePicker';
import LoadingSpinner from '../components/loading/loader';
import HomeButton from '../components/buttons/HomeButton';
import NavButtons from '../components/buttons/NavButtons';

import API from '../api';

class PhotoInputScreen extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      image: null,
      loading: false,
      showImageHelp: false,
      resultEmail: null,
      result: null,
      fadeAnim: new Animated.Value(1),
    };
    let formData = new FormData()
  }

```

Figure 5: PhotoInputScreen imported libraries code snippet.

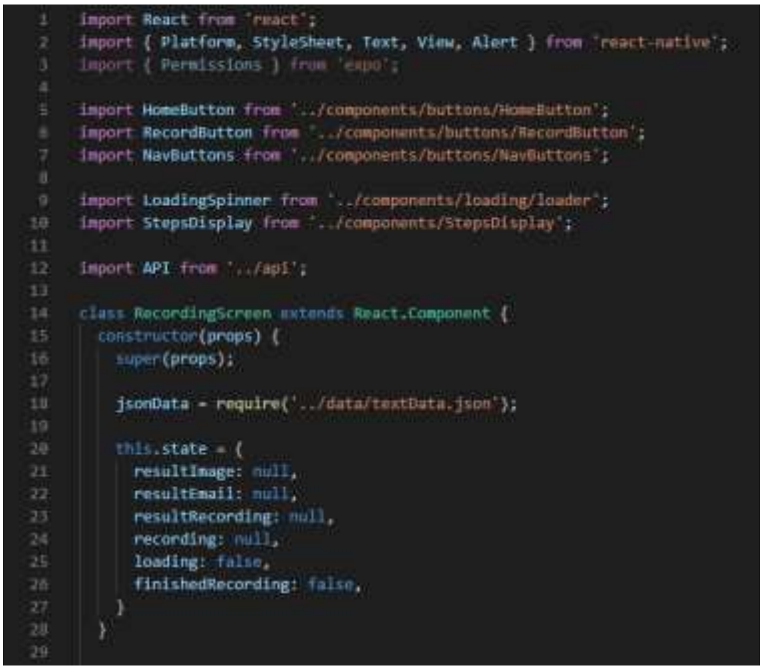
Figure 6 illustrates how to send a photo to the server by using the POST method. In general, PhotoInput component receives the user's image and then sends it to the backend by calling API POST method to examine the user's face from that picture. In fact, it also tests whether or not this is the human image, and the feature warns when the picture is not the human image, so the user has to re-enter the new image. Otherwise, as a Recording step, they will let the process run to the next stage.



```
API.post(`face`, formData, {headers: headers})
.then((res) => {
  this.setState({ result: res.data }, () => {
    console.log(this.state.result);
    if (this.state.result['results']['Hesitant'] < 10 && this.state.result['results']['Dom
      Alert.alert(
        jsonData.error,
        jsonData.photo_failed,
        [
          {text: jsonData.try_again, onPress: () => {
            this.setState({loading: false});
            this.setState({image: null});
          }},
        ],
      );
    } else {
        this.props.navigation.navigate('Recording', {resultImage: this.state.result});
```

Figure 6: PhotoInputScreen API POST method code snippet.

b. Recording Screen

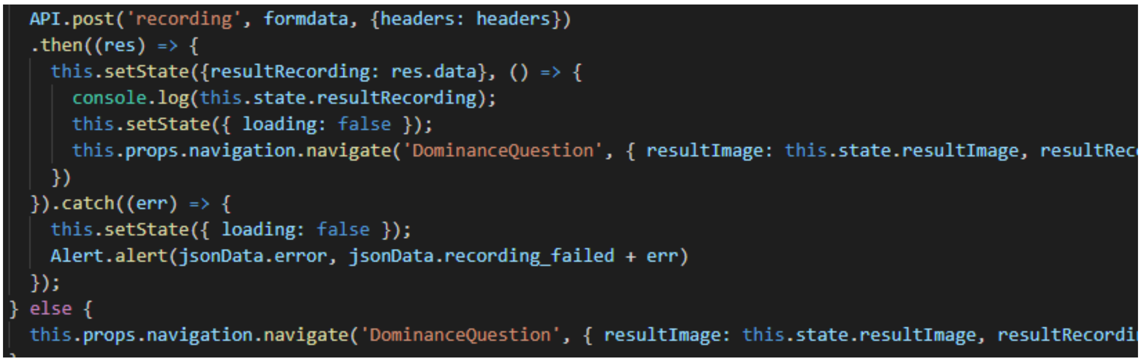


```
import React from 'react';
import { Platform, StyleSheet, Text, View, Alert } from 'react-native';
import { Permissions } from 'expo';

import HomeButton from '../components/buttons/HomeButton';
import RecordButton from '../components/buttons/RecordButton';
import NavButtons from '../components/buttons/NavButtons';

import LoadingSpinner from '../components/loading/loader';
import StepsDisplay from '../components/StepsDisplay';

import API from '../api';

class RecordingScreen extends React.Component {
  constructor(props) {
    super(props);

    jsonData = require('../data/textData.json');

    this.state = {
      resultImage: null,
      resultEmail: null,
      resultRecording: null,
      recording: null,
      loading: false,
      finishedRecording: false,
    }
  }

```

Figure 7: RecordingScreen imported libraries code snippet.

Same as the Photo Input Screen in the Figure 5, different built-up components such as HomeButton, RecordButton, NavButton, LoadingSpinner and StepDiaplay in the Recording Screen are used as seen in the figure 7.

To have the exact tone of their voice, users need to do some recordings for around 10 seconds in Recording Screen. After that the API POST method will send the recording file to the backend. There will the file be evaluated and analysed, user will, however, continue the workflow straight to DominanceQuestion screen.

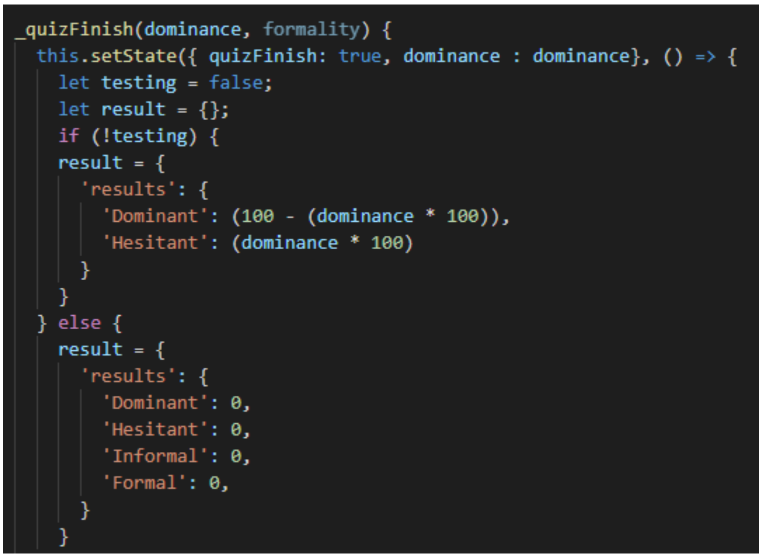


```
  API.post('recording', formdata, {headers: headers})
  .then((res) => {
    this.setState({resultRecording: res.data}, () => {
      console.log(this.state.resultRecording);
      this.setState({ loading: false });
      this.props.navigation.navigate('DominanceQuestion', { resultImage: this.state.resultImage, resultRec
    })
  }).catch((err) => {
    this.setState({ loading: false });
    Alert.alert(jsonData.error, jsonData.recording_failed + err)
  });
} else {
  this.props.navigation.navigate('DominanceQuestion', { resultImage: this.state.resultImage, resultRecordi
```

Figure 8: RecordingScreen API POST method code snippet.

c. DominantQuestion Screen

Figure 9 represents how to calculate percentage of dominant and hesitant (assertive and meticulous as displayed respectively in the front-end) which is one of important methods to determine the user’s dominant style. Firstly, couple of questions stored in questionaireData.json are proposed on this screen. User will have to answer these before the style such as Dominant (Assertive) or Hesitant (Meticulous) will be generated based on the provided answers. Finally, the result will be sent to the Result Screen from the DominantQuestion Screen.

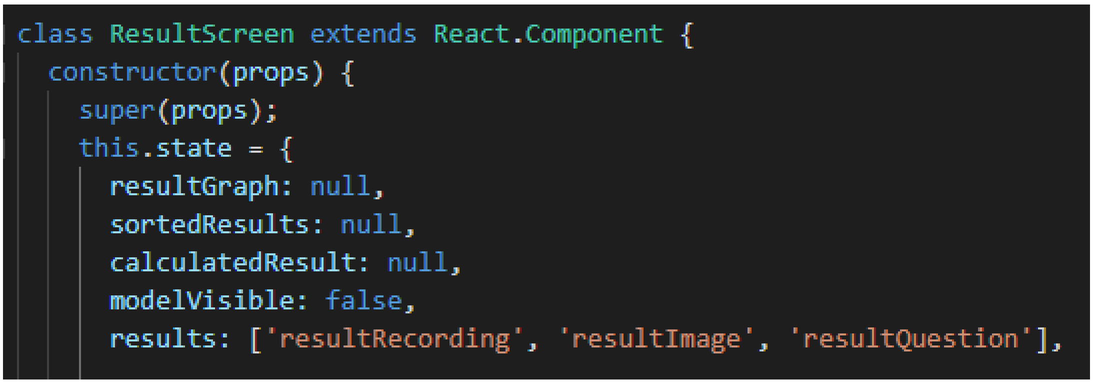


```
_quizFinish(dominance, formality) {
  this.setState({ quizFinish: true, dominance : dominance}, () => {
    let testing = false;
    let result = {};
    if (!testing) {
    result = {
      'results': {
        'Dominant': (100 - (dominance * 100)),
        'Hesitant': (dominance * 100)
      }
    }
  } else {
    result = {
      'results': {
        'Dominant': 0,
        'Hesitant': 0,
        'Informal': 0,
        'Formal': 0,
      }
    }
```

Figure 9: DominantQuestion method code snippet.

d. Result Screen

Result Screen is where all the results will be collected from image analysis, recording analysis and dominance question analysis (Figure 10). After that, the graph will be drawn based on those results and displayed in this Result Screen. Moreover, users can read the definition of their communication style by clicking Info Button.



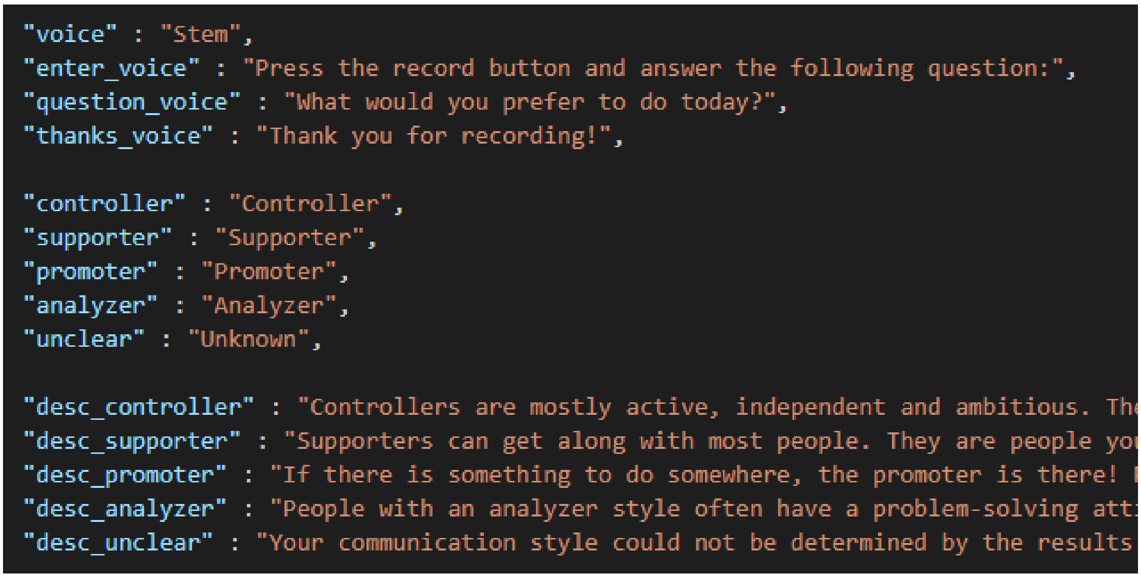
```
class ResultScreen extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      resultGraph: null,
      sortedResults: null,
      calculatedResult: null,
      modelVisible: false,
      results: ['resultRecording', 'resultImage', 'resultQuestion'],
```

Figure 10: ResultScreen code snippet.

4.1.2 Data Storage

To avoid hard-coding and make data easier to be fixed, two different json files were built to store the data and information in the app, which are questionaireData and textData.

Whilst textData.json will store general information, such as app title, description, and communication style definitions, the QuestionnaireData.json will store the various questions, responses and contact styles from the DominanceQuestion screen.



```
"voice" : "Stem",
"enter_voice" : "Press the record button and answer the following question:",
"question_voice" : "What would you prefer to do today?",
"thanks_voice" : "Thank you for recording!",

"controller" : "Controller",
"supporter" : "Supporter",
"promoter" : "Promoter",
"analyzer" : "Analyzer",
"unclear" : "Unknown",

"desc_controller" : "Controllers are mostly active, independent and ambitious. Th
"desc_supporter" : "Supporters can get along with most people. They are people yo
"desc_promoter" : "If there is something to do somewhere, the promoter is there!
"desc_analyzer" : "People with an analyzer style often have a problem-solving att
"desc_unclear" : "Your communication style could not be determined by the results
```

Figure 11: JSON data in the textData json file.

4.1.3 Connection between Front-end and Back-end

To send input from the React Native user interface to the server and vice versa, a connection between them need to be created by using Restful API service. This is because “REST uses standard HTTP requests, our data validation and address verification APIs are easy for developers to understand and implement. In addition, RESTful architectures make it easy to provide outputs in more flexible data formats like JSON” (Serviceobject, n.d.).

In general, our project will use POST and GET methods in order to send and get data. To do this, we have used the localhost endpoint to host our backend server which is loaded inside the axios.create function in the api.js file. ”Axios is a Javascript library used to make HTTP requests from node.js or XMLHttpRequests from the browser” (Kollegger, 2018). Therefore, using the axios.create function will create HTTP request from our localhost enpoint from the server to the React Native interface.

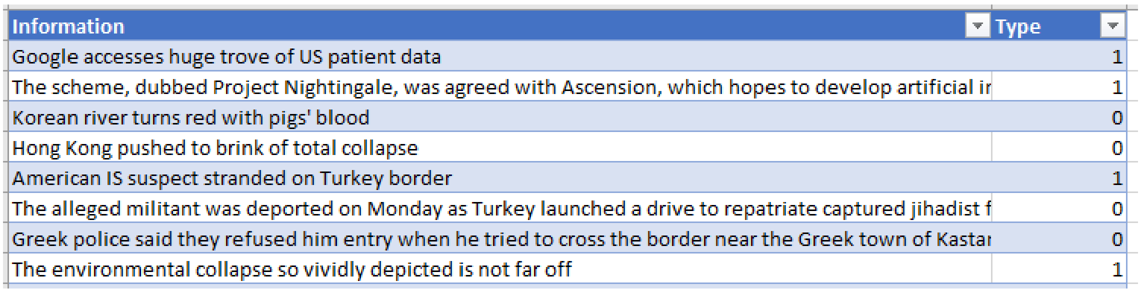
After that, we imported this api.js file into other files with the purpose of querying data or posting user's inputs such as images or recording files. Using the API base link will be the advanced option because we just need to fix the link from the api.js file if there is an error.

4.2 Back-end

4.2.1 Natural Language Processing

a. Machine Learning Model

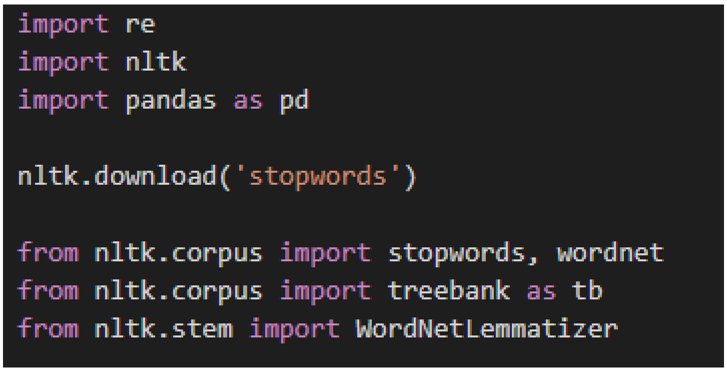
The most challenging problem appears to be having dataset for training. To achieve this, we have decided to collect the data from various resources – note that all the data were randomly picked or self made-up: from newspapers such as Yle (in Finnish), NRK (in Norwegian), BBC, CNN, Financial Times (in English), comments on SoMe such as Facebook, Instagram. For every piece of information, it is stored in a separate row along with a label: one (1) for positive classified and vice versa.



Information	Type
Google accesses huge trove of US patient data	1
The scheme, dubbed Project Nightingale, was agreed with Ascension, which hopes to develop artificial ir	1
Korean river turns red with pigs' blood	0
Hong Kong pushed to brink of total collapse	0
American IS suspect stranded on Turkey border	1
The alleged militant was deported on Monday as Turkey launched a drive to repatriate captured jihadist f	0
Greek police said they refused him entry when he tried to cross the border near the Greek town of Kastar	0
The environmental collapse so vividly depicted is not far off	1

Figure 12: Piece of labelled text data.

Libraries required for importing csv file, regular expression (regex), and performing natural language processing are: pandas, re, nltk.



```
import re
import nltk
import pandas as pd

nltk.download('stopwords')

from nltk.corpus import stopwords, wordnet
from nltk.corpus import treebank as tb
from nltk.stem import WordNetLemmatizer
```

Figure 13: Libraries required for NLP.

Steps to pre-process the text are:

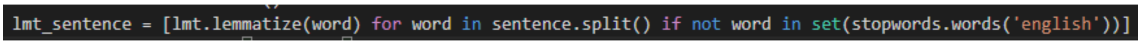
1. Import the csv file (pandas),
2. Remove punctuations, stop-words,
3. Lower all the tokens,
4. Lemmatise or Stem tokens (WordNet).

An extra step is to find the synonyms, antonyms of each meaningful token.

In this regard, stop-words are considred word tokens that are of no use. For example, the absence of the article “the” in this sentence : “I like the tree!” will remain the original meaning of the sentence (“I like tree!”). Furthermore, since machine is not capable of interpreting human emotion, the presence of exclamation mark (!) does not increase any understanding. Henceforth, unnecessary tokens and such should be discarded to improve the speed, reduce memory allocation, as well as increase accuracy of the ML model.

The purpose of lemmatising or stemming is still debatable. Whilst stemming is the technique used to find the stem of a token, whilst lemmatisation will trace a token back to its originiality. For instance, if “nature” or “natures” is stemmed to “natur-”, it will be lemmatised to “nature” instead. Therefore, due to the small size of the dataset (259 x 2), stemming would be less useful since it will reduce the vocabulary to the uttermost compared to lemmatisation.

Thanks to the pre-eminence to Python List Comprehension, the task is syntactically minimised by one line of code:



```
lmt_sentence = [lmt.lemmatize(word) for word in sentence.split() if not word in set(stopwords.words('english'))]
```

Figure 14: Lemmatisation, removing stop-words and getting token in one line.

The next step is to perform ML. Here are two concepts will be exercised: TF-IDF and Naïve Bayes.

TF (tfidf, n.d.): term frequency – the number of occurrences of a token in a corpus. For example, a document includes two sentences: “I love tree. Tree is green.” The query of interest is “tree”, which appears two times in the document. Other terms are: I, love, is, green (one occurrence each). TF will be equal to number of occurrence the term “tree” appears over the number of terms in the whole document, which is two over five (or 0.4) in this case.

Idf (tfidf, n.d.): inverse document frequency – scaling down the occurence of frequent whilst weighting up the rare ones (“am/is/are” compared to “ML” for instance). This is the most preferred technique alongside the usual vectorisation thanks to the application of normalisation to find out how important a token is in a specific corpus (document). Suppose there exists a corpus of N documents, f_{ij} is the frequency of token *i* in document *j*, n_i indicates token *i* appears in n_i document in the collection. TF-IDF is defined as:

$$TF(i,j) = \frac{f_{ij}}{max_k f_{k,j}}, IDF(i) = \ log_2\left(\frac{N}{n_i}\right) \ \rightarrow \ TF(i,j) * IDF(i)$$

Naïve Bayes (NB): is one of the well-known statistical methodologies for ML. It is also considered as conditional probability technique. The formula is given as:

$$P(A|B) = \frac{P(A) * P(B|A)}{P(B)}$$

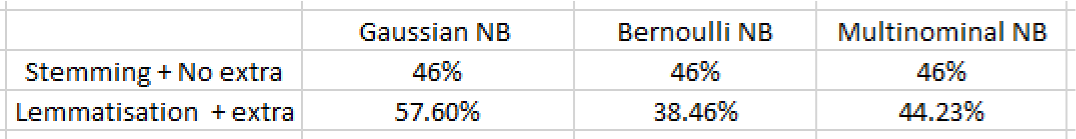
read as: probability of A given B or probability of A when B happened. The algorithm is used to classify and predict where a new data point would fall into based on its reasonings.

There are also three (03) types of NB: Gaussian NB, Bernoulli NB, Multinomial NB. Simply said, Gaussian NB is NB with the assumption of Gaussian (normal distribution), Bernoulli NB is to classify 0s and 1s, and the last one is applied to discrete data.

Another notice is the training – testing proportion. As preferred, the golden fraction goes to either 70 – 30 or 80 – 20 for training and testing respectively. When carrying out the test, the latter ratio (80 – 20) was selected.

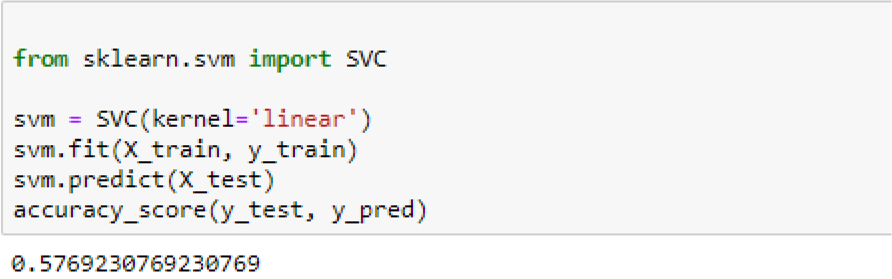
After performing the task without an extra step of increasing the vocabulary list (synonyms and antonyms), the accuracy of three (03) methods are quite the same, ranging around 45 – 46%, which is quite low and not desirable. In the hope of boosting the accuracy, a new approach was picked: increasing the vocabulary with lemmatisation. Table 2 below displays the accuracy of the three methods:

Table 2: Accuracy of three NB methods.



	Gaussian NB	Bernoulli NB	Multinominal NB
Stemming + No extra	46%	46%	46%
Lemmatisation + extra	57.60%	38.46%	44.23%

As a result, Gaussian NB is the one chosen. However, the work is advanced with a null hypothesis (H_o) is that Gaussian NB has not done its best at classifying. Hence, a further step of implementing Support-Vector Machine algorithm (detailed explanation in the Computer Vision section) is performed. The result, nevertheless, remains the same (57.6%), which leads to that of rejecting the H_o.



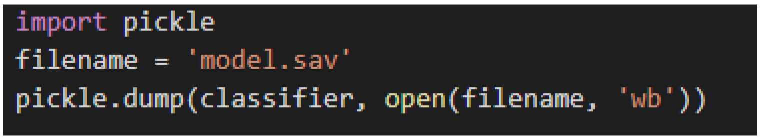
```
from sklearn.svm import SVC

svm = SVC(kernel='linear')
svm.fit(X_train, y_train)
svm.predict(X_test)
accuracy_score(y_test, y_pred)
```

```
0.5769230769230769
```

Figure 15: Snippet code of SVM implementation.

57.6% is the highest percentage that can be reached in this case. The next step is to pickle this model into a binary file so that the ML pipeline can be easily, quickly accessed and yield out prediction fastly. Short words regarding pickle, it is the technique used to convert a file into a binary data so that time and memory allocation is reduced. Once all the work is pickled, the machine does not need to go through all the previous work such as: import libraries, lemmatising, perform ML techniques but to employ the final saved results.

The syntax for this is quite simple, “dump the variable into a file provided the name and its format”.

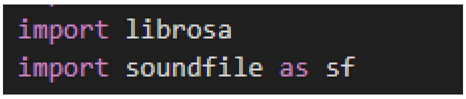


```
import pickle
filename = 'model.sav'
pickle.dump(classifier, open(filename, 'wb'))
```

Figure 16: Snippet code of pickling the model.

b. Voice recording and Formality of language used:

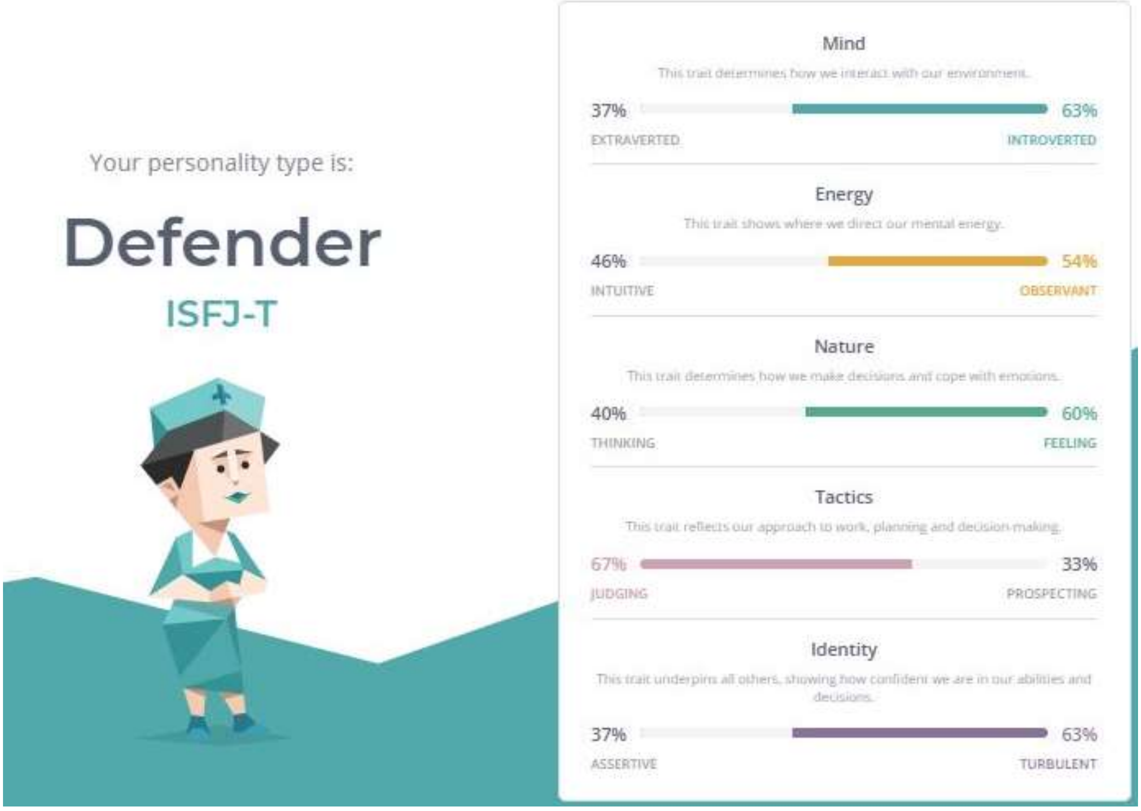
The libraries used to analyse metadata in the section are: librosa, soundfile



```
import librosa
import soundfile as sf
```

Figure 17: Snippet code of importing libraries librosa and soundfile.

Regardless of the statistics numbers introduced in the section (3.1.4) as the benchmark, a small survey was conducted. The goal is to understand whether personality type (from MBTI test) correlates with the language one uses. To achieve this, we invite the participants to take the MBTI test available at https://www.16personalities.com/ and fill in the results in the Google form.



Your personality type is:

Defender

ISFJ-T

Mind

This trait determines how we interact with our environment.

37% EXTRAVERTED — 63% INTROVERTED

Energy

This trait shows where we direct our mental energy.

46% INTUITIVE — 54% OBSERVANT

Nature

This trait determines how we make decisions and cope with emotions.

40% THINKING — 60% FEELING

Tactics

This trait reflects our approach to work, planning and decision-making.

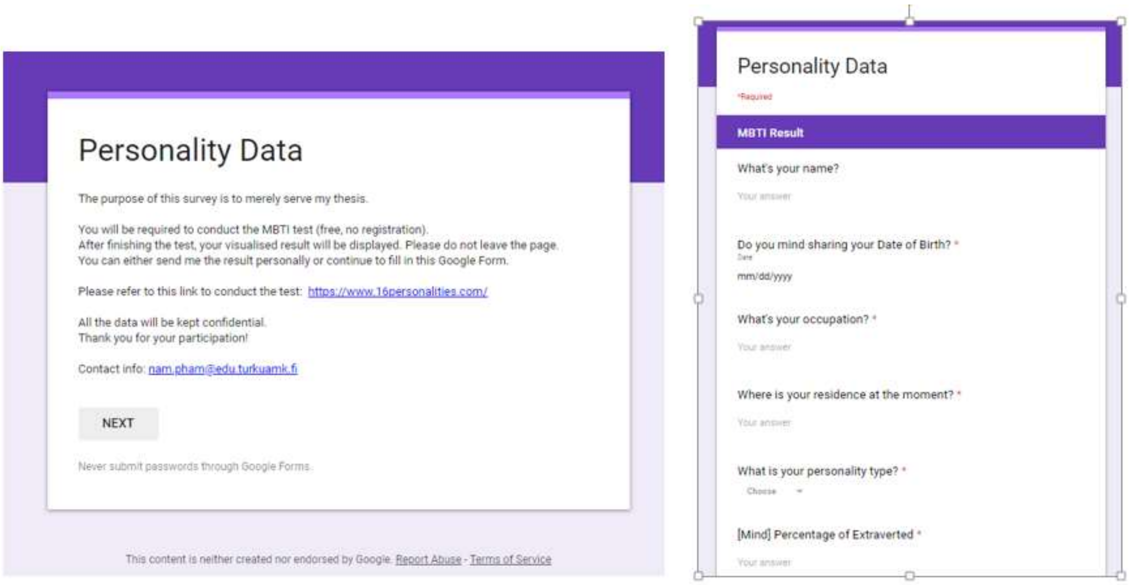
67% JUDGING — 33% PROSPECTING

Identity

This trait underpins all others, showing how confident we are in our abilities and decisions.

37% ASSERTIVE — 63% TURBULENT

Figure 18: Sample Result of the MBTI test.



Personality Data

The purpose of this survey is to merely serve my thesis.

You will be required to conduct the MBTI test (free, no registration).
After finishing the test, your visualised result will be displayed. Please do not leave the page.
You can either send me the result personally or continue to fill in this Google Form.

Please refer to this link to conduct the test: https://www.16personalities.com/

All the data will be kept confidential.
Thank you for your participation!

Contact info: nam.pham@edu.turkuamk.fi

NEXT

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. Report Abuse - Terms of Service

Personality Data

*Required

MBTI Result

What's your name?

Your answer

Do you mind sharing your Date of Birth? *
Date
mm/dd/yyyy

What's your occupation? *

Your answer

Where is your residence at the moment? *

Your answer

What is your personality type? *
Choose

[Mind] Percentage of Extraverted *

Your answer

Figure 19: Snapshot of Google Form.

The interested statistics metrics are: Mind – Nature – Tactics. Intuitively and personally, the three components are believed to contribute a lot to the speaking style one may generate. Assumption is that formal language may be obtained from those who:

- Not too introverted or extraverted (Mind): extraverted 40% – 55% ± 3%,
- Balanced thinking and feeling (Nature): thinking 40% – 55% ± 5%,
- Has the analytical approach (Tactics): judging 50% - 65% ± 5%.

Below is the visualised statistics of our sample:

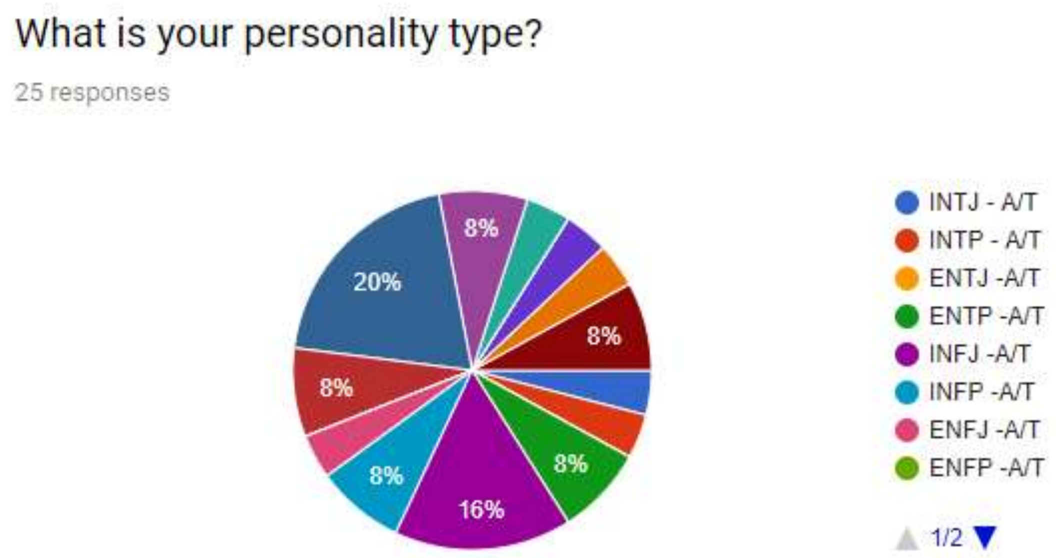


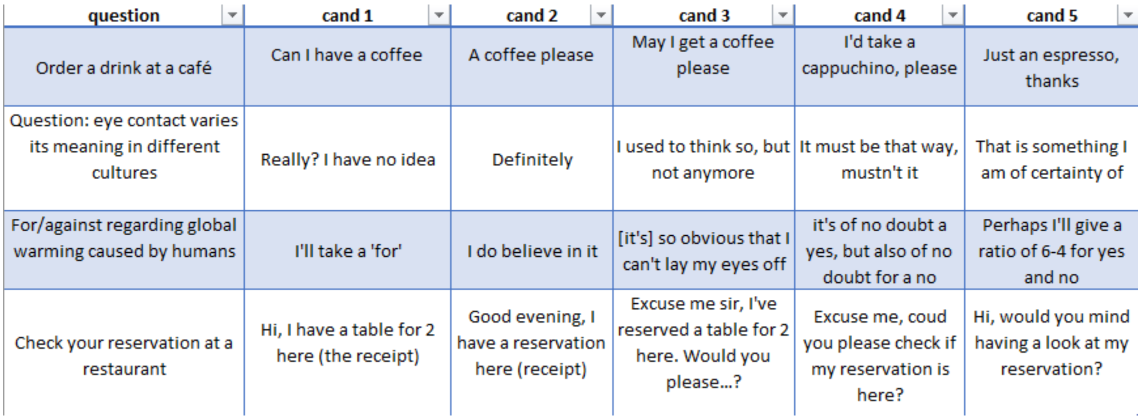
Figure 20: MBTI test visualisation of the population.

The aim of carrying out a statistics test with:

- Null hypothesis (H_o): those who satisfy the requirements mentioned above will offer a more formal conversation than others (or broader range of vocabularies).
- Alternative hypothesis (H_1): the requirements mentioned cannot tell whether a candidate has broader range of vocabularies.

After consideration the metrics, unfortunately, both Z-test and T-test cannot be performed due to external conditions (the sample is not normally distributed and is not large enough for Z-test, or has unknown standard deviation). Therefore, a number of five random candidates from the sample with different personalities were selected. “Semi-structured interview” method (Cohen & Crabtree, 2006) was applied. This means, each person was invited for a talk in a relaxed atmostphere alongside with some fixed questions to measure the length of the answers. The interviewees were asked the prepared questions accompanied with the instruction, for instance “in one sentence, what do you think about”. Examples are shown in the tabe 3 below:

Table 3 : Examples of asked questions and responses.

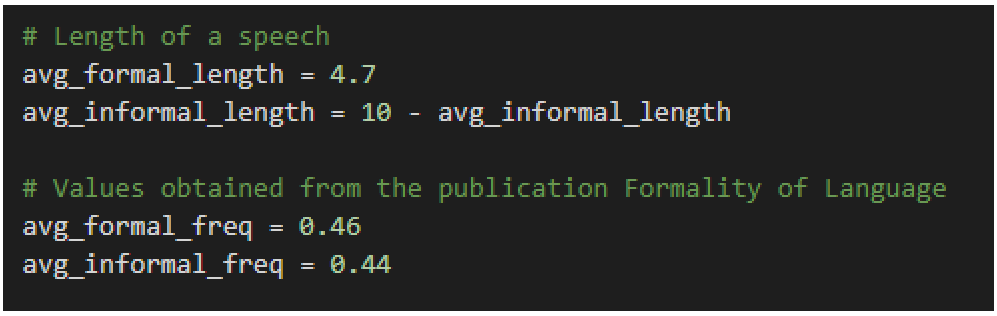


question	cand 1	cand 2	cand 3	cand 4	cand 5
Order a drink at a café	Can I have a coffee	A coffee please	May I get a coffee please	I'd take a cappuchino, please	Just an espresso, thanks
Question: eye contact varies its meaning in different cultures	Really? I have no idea	Definitely	I used to think so, but not anymore	It must be that way, mustn't it	That is something I am of certainty of
For/against regarding global warming caused by humans	I'll take a 'for'	I do believe in it	[it's] so obvious that I can't lay my eyes off	it's of no doubt a yes, but also of no doubt for a no	Perhaps I'll give a ratio of 6-4 for yes and no
Check your reservation at a restaurant	Hi, I have a table for 2 here (the receipt)	Good evening, I have a reservation here (receipt)	Excuse me sir, I've reserved a table for 2 here. Would you please...?	Excuse me, coud you please check if my reservation is here?	Hi, would you mind having a look at my reservation?

In general, it can be learned that the formality of the response will be in accordance with the context. For example, when in a starred-restaurant, participants tend to be more polite and formal with their speeches, which is in contrast to that at a café. Furthermore, the point of semi-structured interview is to, as said, reduced and distract the concentration of the candidates. It can be seen clearly at the candidate 5 (cand 5), when lack of awareness, the first response is formal and casual enough compared to the tautological in the second response after having recalled himself/herself of the goal of the interview. Reducing the tautology means reducing the bias of the survey.

In addition to that, the average length of the response ranges from 3.6 to 4.8 seconds (calculated by the built-in recording application in the iPhone). Hence, it is agreed to take the bigger values (4.7) since neither English is native language to some users nor probable technical errors that may occur.

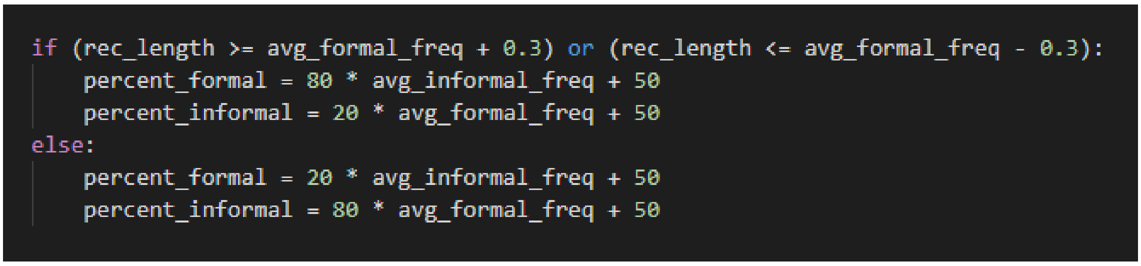
Afterwards, some important answered were mimicked and recorded to measure its length. All the metrics used in this section are listed as shown in the figure 21:



```
# Length of a speech
avg_formal_length = 4.7
avg_informal_length = 10 - avg_informal_length

# Values obtained from the publication Formality of Language
avg_formal_freq = 0.46
avg_informal_freq = 0.44
```

Figure 21: Important metrics.

Given the context, the formality is decided based on the length of the recorded. If it falls into the range of [4.4, 5.0] (the average formal calculated ± 0.3), the formality accounts for 80 points whilst the others takes 20 points. Furthermore, to reduce the bias in case the recorded is long enough to reach the target but is informal, the metric is multipled with the reversed frequency of formality (obtained from the publication Formality of language mentioned in the Literature). That says, the formality metric will be accompanied with the informality frequency and vice versa.



```python
if (rec_length >= avg_formal_freq + 0.3) or (rec_length <= avg_formal_freq - 0.3):
    percent_formal = 80 * avg_informal_freq + 50
    percent_informal = 20 * avg_formal_freq + 50
else:
    percent_formal = 20 * avg_informal_freq + 50
    percent_informal = 80 * avg_formal_freq + 50
```

Figure 22: Python codes for formality calculation.

Notice the value 80-20 is randomly picked in accordance with the golden ratio when splitting train-test set in ML while 50 is the base point. On of that, the use of variables instead of the value itself will facilitate the process of tracking if errors occur or updating the code.

4.2.2 Computer Vision

Initially, CV is thought to be integrated as described:

- User will be required to import a picture (or a selfie).
- The picture will be sent to back-end for preprocessing.
- The machine will make prediction and return corresponding data.

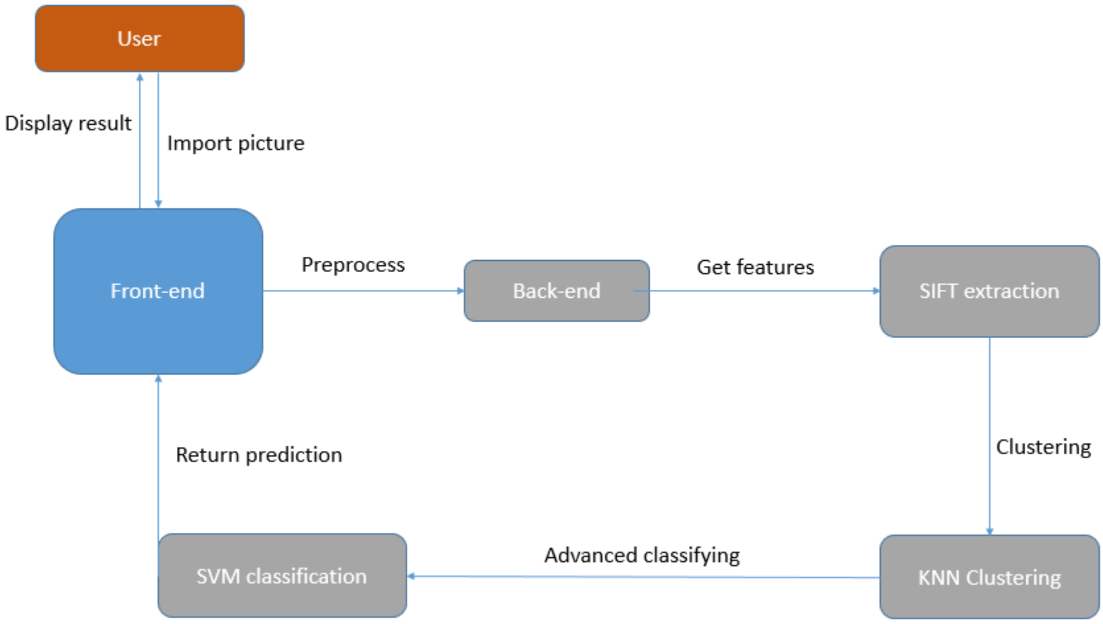


Figure 23: CV for image prediction process.

The trial is done with four (04) images: accordion, soccer ball, motorbike, and dollar bill.

Figure 24: Experiment pictures.

The dataset is divided with the proportion of 70 – 30: 70% of the pictures are used for training, the rest is testing.

a. Bag of Words – SIFT extraction and K-nearest neighbours clustering

Bag of Words is an old statistics method of frequency counting. Unlike other obvious analyses where patterns are visible, for example frequency of words, an image is practiced in a different way. Its pattern is considered the meaningful edges of an image. Let us explain why the edges are more meaningful by a simple example below:

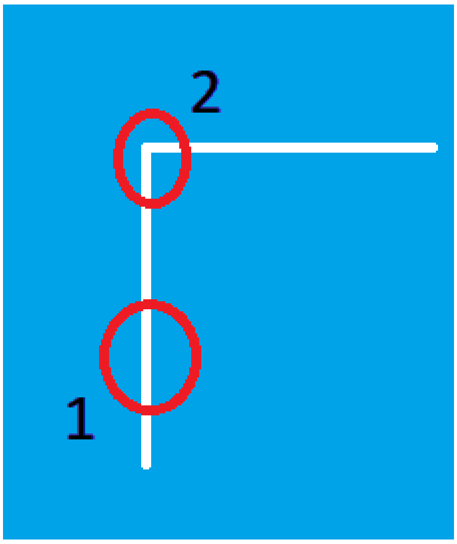


Figure 25: Example of meaningful edges.

At the position 1:

- X-axis: the colours are the same (blue) → one feature learned.
- Y-axis: the colours are the same (white) → one feature learned.

➔ 2 features learned in total

At the position 2:

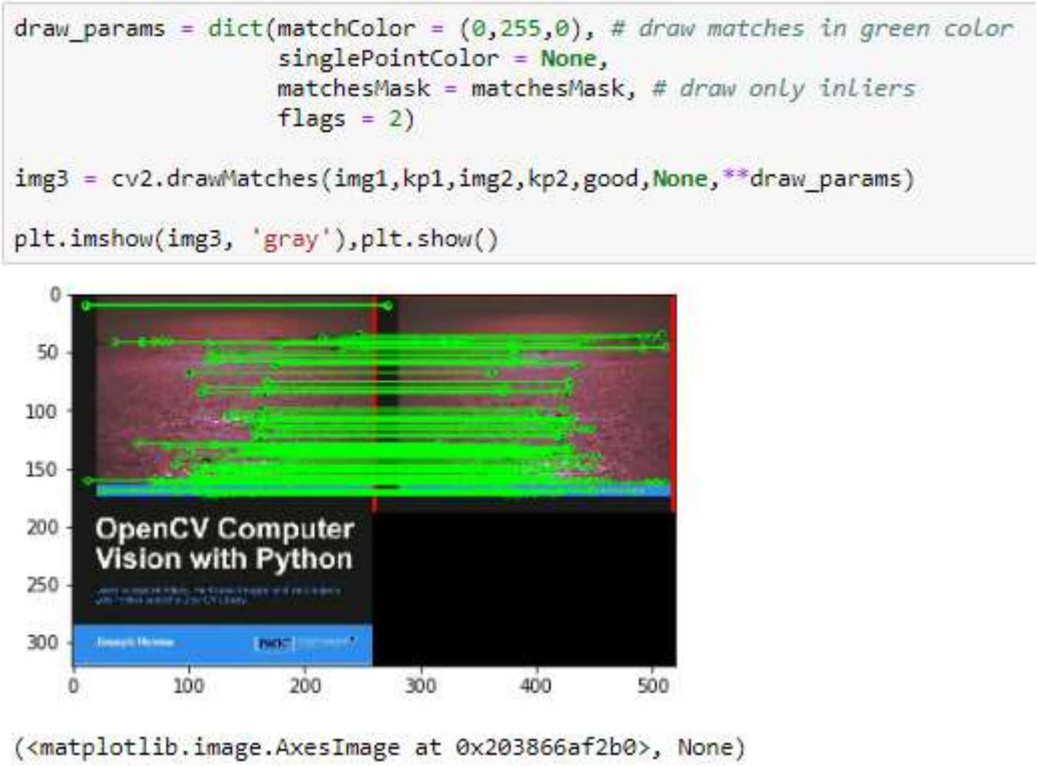
- X-axis: the colours are different (blue vs. white) → 2 features learned.
- Y-axis: the colours are different (blue vs. white) → 2 features learned.

➔ 4 features learned in total.

As a result, the edge (position 2) would bring more knowledge and require less memory allocation for a machine.

However, it is not the edge detection that matters in this case. The reason is that with the same object, the angle can vary leading to another training case for the machine with same object. This, in fact, will cause problems due to machine memory constraints.To avoid this, SIFT – scale-invariant features transform – was introduced by David Lowe (Lowe, 1999) in 1999.

SIFT Transform is a well-known algorithm used in image stitching – matching features of this image with the corresponding one in the other image regardless of size or angles. An example of image stitching is shown in figure 26:



```
draw_params = dict(matchColor = (0,255,0), # draw matches in green color
                   singlePointColor = None,
                   matchesMask = matchesMask, # draw only inliers
                   flags = 2)

img3 = cv2.drawMatches(img1,kp1,img2,kp2,good,None,**draw_params)

plt.imshow(img3, 'gray'),plt.show()
```

```
(<matplotlib.image.AxesImage at 0x203866af2b0>, None)
```

Figure 26: Image stitching with SIFT extractions.

The good thing is SIFT algorithm is available for use in OpenCV.

After figuring out all the meaningful features, the vocabulary list is needed. KNN clustering method will be applied to increase the vocabulary of the machine.

K-nearest neighbours clustering (KNN) is an old, unsupervised ML technique. A simple explanation of KNN will be that after defining a number of k randomly – number of centroids, all the data points will be clustered according to the distance to the centroid.

In this specific circumstance of the test, because each image will contain numerous features, vectorisation technique must be used to reduce the memory requirement and speed up the processing time. Features of all the training images will be stacked into vectors; based on this, they will be re-grouped, and be standardised to reduce the bias for ML later. Eventually a complete vocabulary list have we received. Here is the final histogram of frequency:

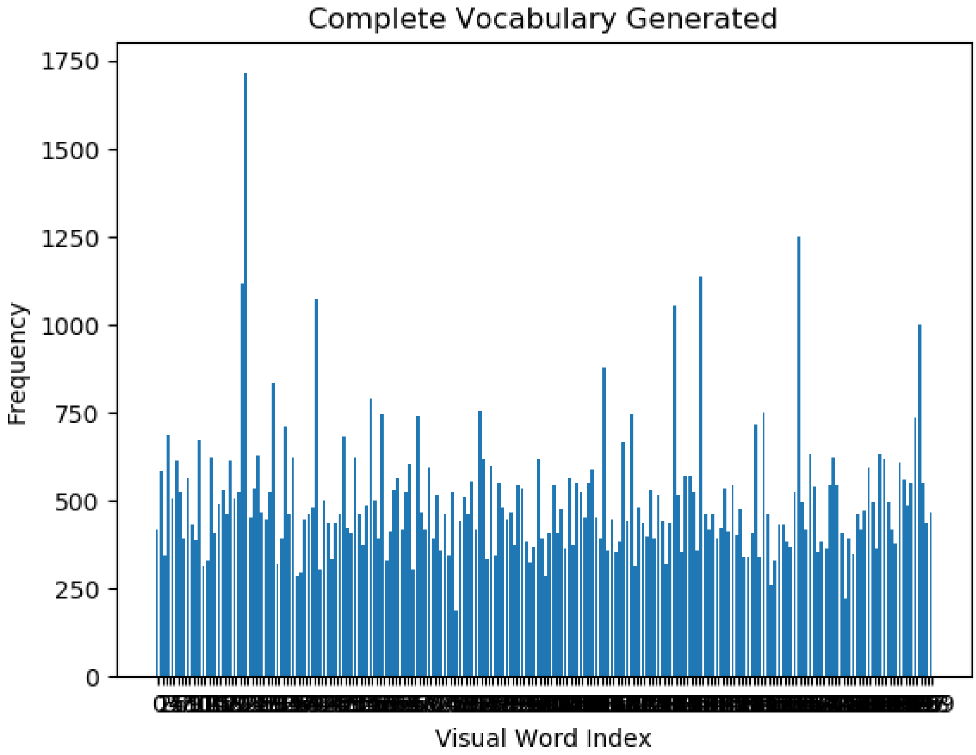


Figure 27: Final Histogram of Frequency.

The ML algorithm used here is Support-Vector Machine (SVM). Let us take an example in two-dimensional (2D): In the Oxy graph, there exists two clusters. What SVM does is to draw an optimal hyperlane – a term used to indicate the decision boundaries in ML (DeepAI, n.d.) – to separate those two clusters so that whenever a new data point comes in, its distance to the surrounding centroids will determine which class it belongs to. With the same concept, however, SVM is advanced in being able to classifying clusters in multi-dimensional space instead of two as in the example.

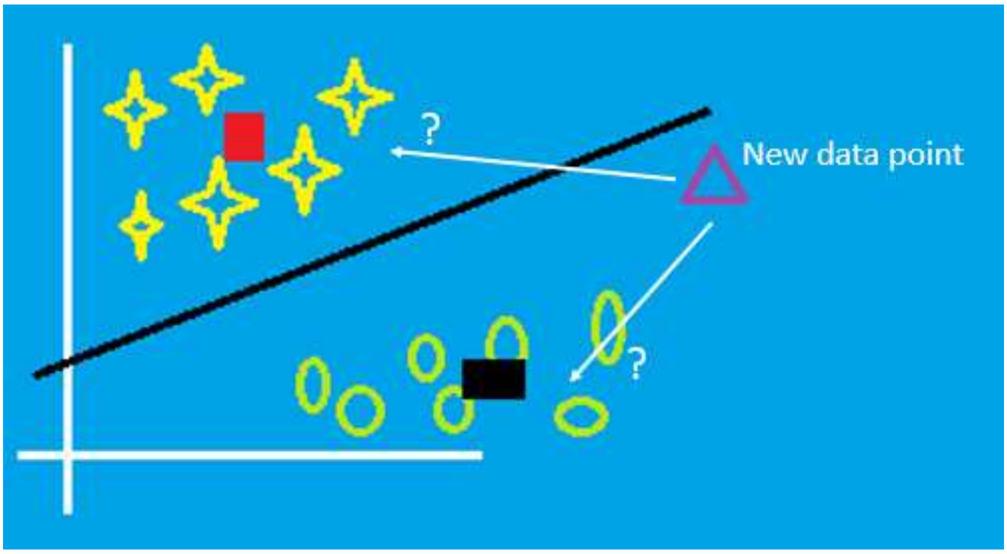
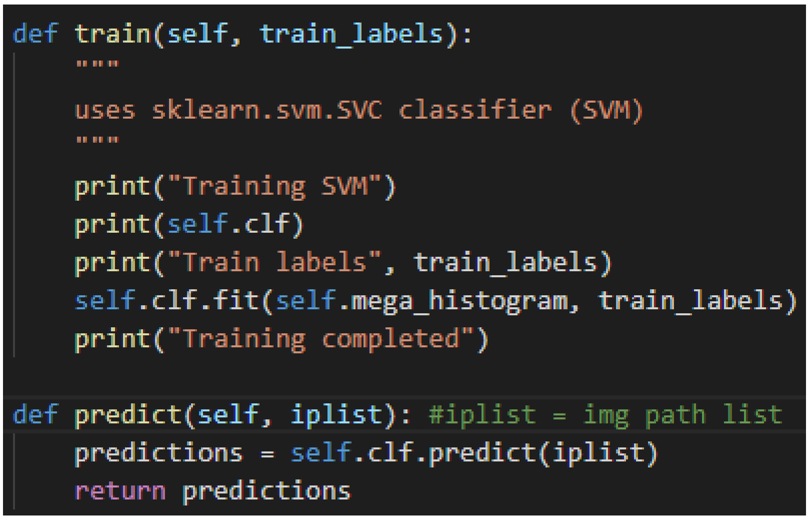


Figure 28: Simple example of SVM.

Thankfully, SVM algorithm is supported in Python SKLEARN library. The syntax to provoke it is quite simple:



```
def train(self, train_labels):
    """
    uses sklearn.svm.SVC classifier (SVM)
    """
    print("Training SVM")
    print(self.clf)
    print("Train labels", train_labels)
    self.clf.fit(self.mega_histogram, train_labels)
    print("Training completed")

def predict(self, iplist): #iplist = img path list
    predictions = self.clf.predict(iplist)
    return predictions
```

Figure 29: Syntax to provoke SVM.

Figure 30 below shows some of correctly classified images.

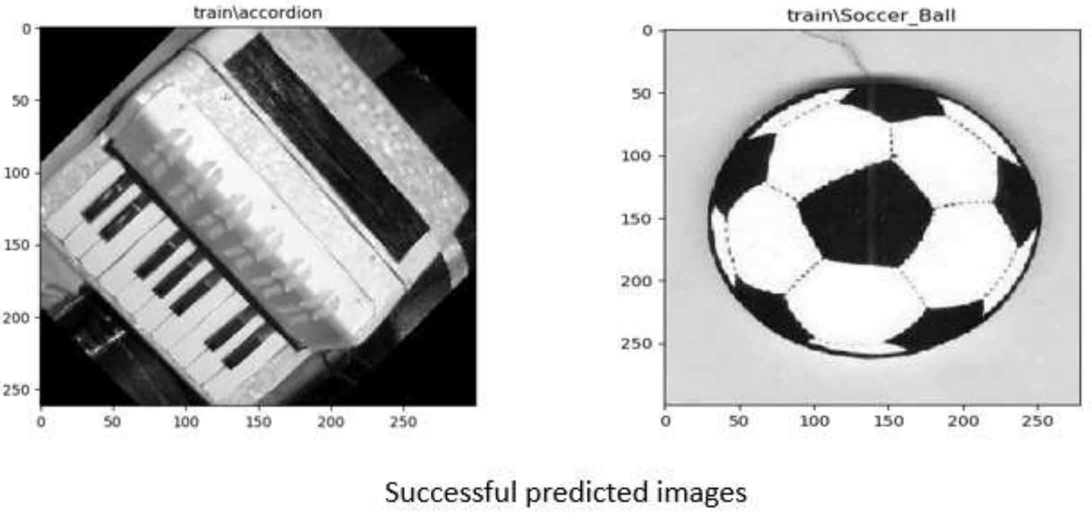


Figure 30: Succesful predicted images of accordion (left) and soccer ball (right).

5 INTERACTIVE EXAMPLE

In this section, images of all functionalities will be introduced. Let us start with the opening screen when launching the app.

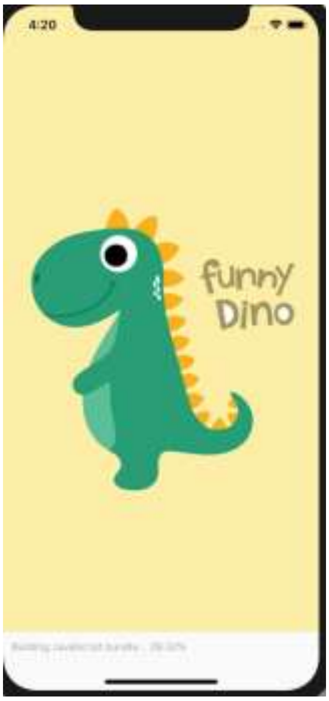
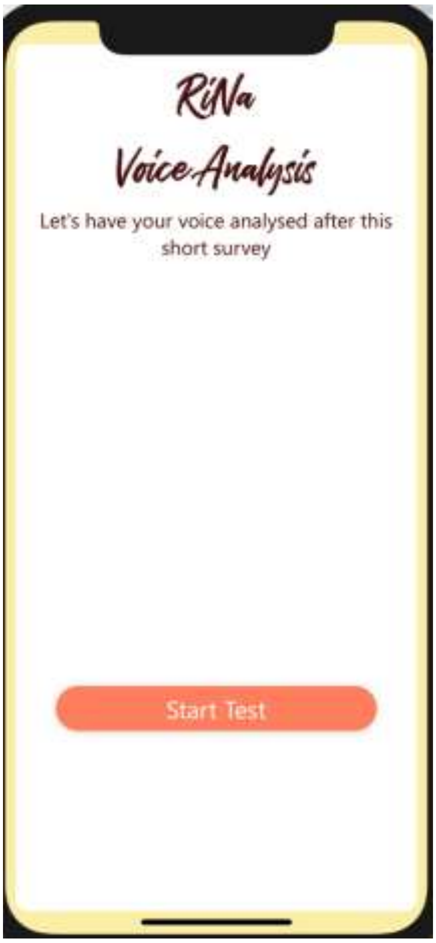


Figure 31: Opening screen of the app.

The app's visuals were decided in accordance with one of the current trends that companies aroud the world, such as BMW, are following: flat characters, transparent colours, and minimalistic in terms of details. "Funny Dino" is, however, not the name of the application. This page is only a flashing opening screen whilst waiting for data to be loaded. Once fully loaded, the users are warmly greeted as in the figure 32.



RiNa

Voice Analysis

Let's have your voice analysed after this short survey

Figure 32: Welcome screen.

RiNa is the name of the app, which stands for Risa (Thao) and Nam. Once again, regarding the font of the title, it is decided to select the old-fashioned calligraphy, which is similar to the trend of the pop-culture at the moment where everything is dating to the 80s. This says, besides focusing on programming, visualisation is another important element of the thesis. Everything needs to be eye-catching, contemporary so that user will not feel bored when interacting with this app. This also shows the social skills and the social understandings of ours.

After clicking “Start Test”, user will be walked to the page where they can upload their image as the first part of the test (figure 33).

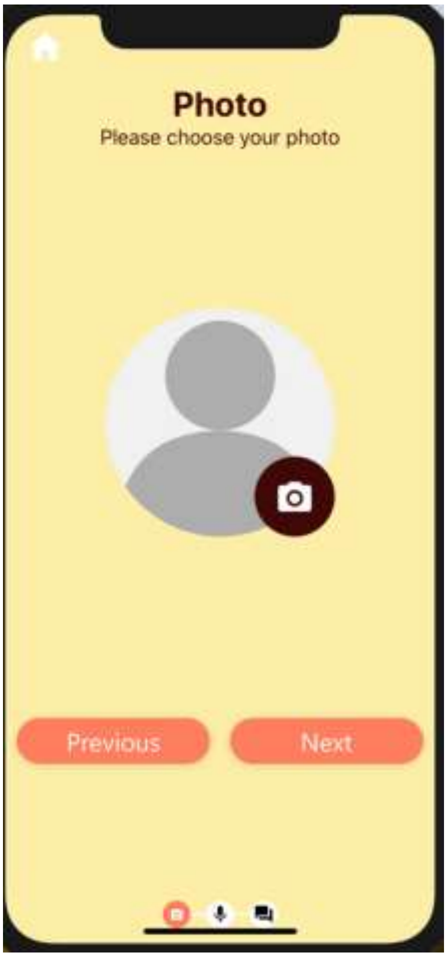
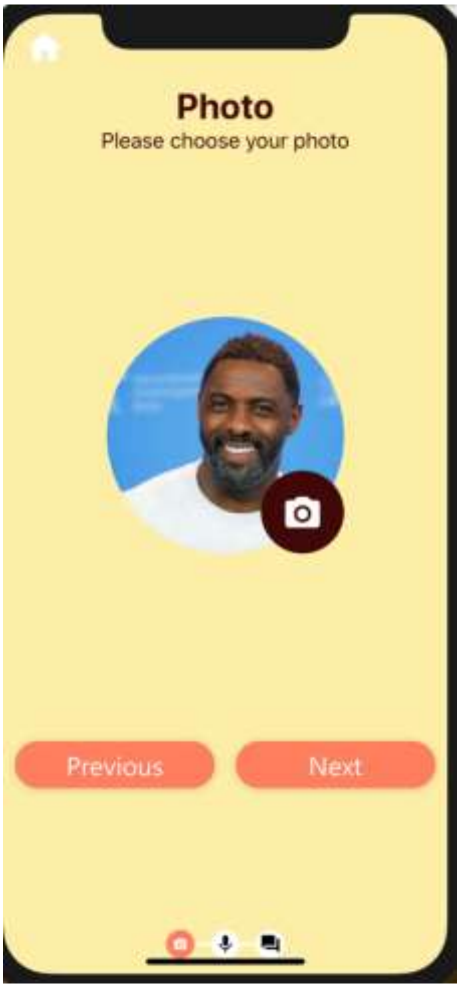


Figure 33: PhotoInput page.

As noticed, the footer of the app also indicates which step it is in the process so that users can keep track on what they have gone through, in case they want to go back and make editions. For the purpose of the test and confidentiality, a random image obtained from the internet is used (figure 34).

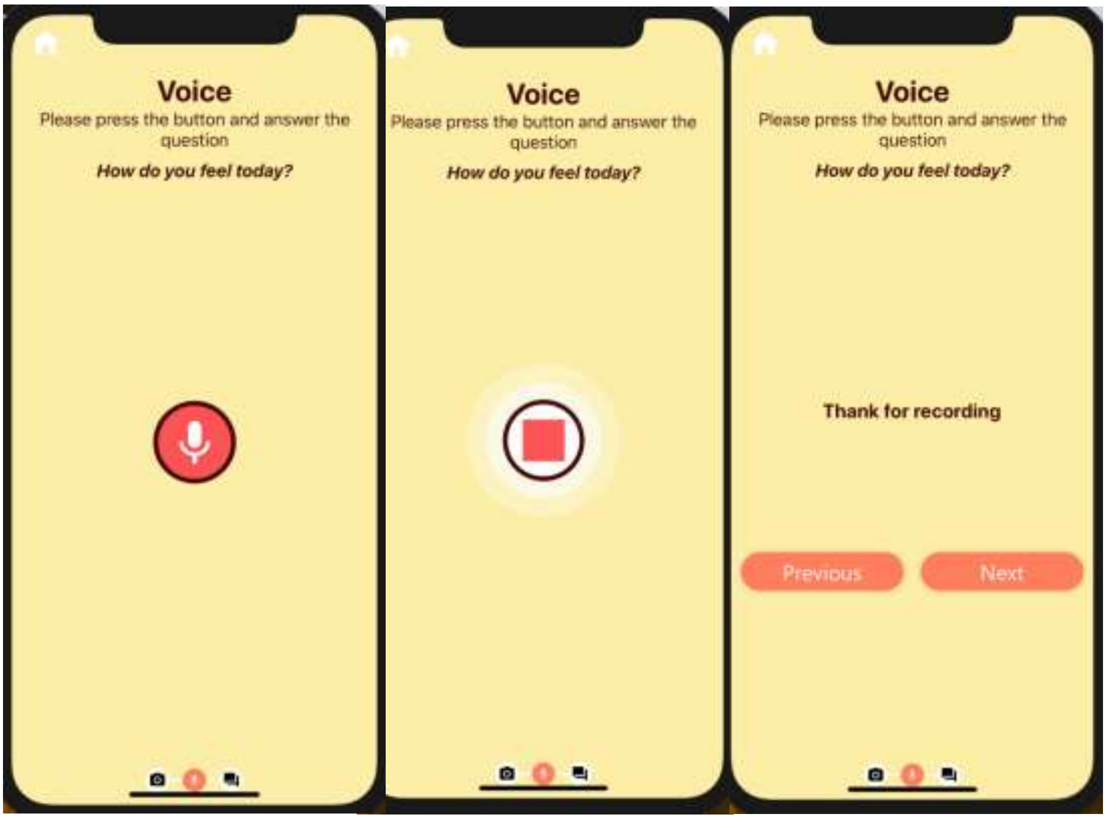


Photo

Please choose your photo

Figure 34: Image inputted.

Once done, the next figure (figure 35) shows the section where users are required to record their answer against a random question, when it is being recorded, and when it finishes recording.



Voice

Please press the button and answer the question

How do you feel today?

Voice

Please press the button and answer the question

How do you feel today?

Voice

Please press the button and answer the question

How do you feel today?

Thank for recording

Figure 35: UI Interactions when recording the answer.

To not make the task boring, the button changes its appearance when pressed, and finally disappeared when unpressed. A *thank you* sentence is in place to thank for the participating in the survey. When all the steps are conducted appropriately, the outcome is displayed as expected (figure 36).

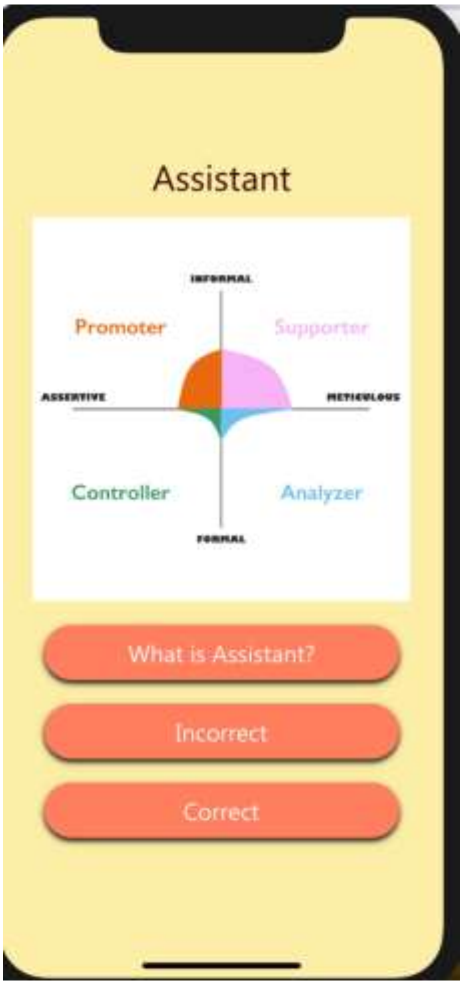
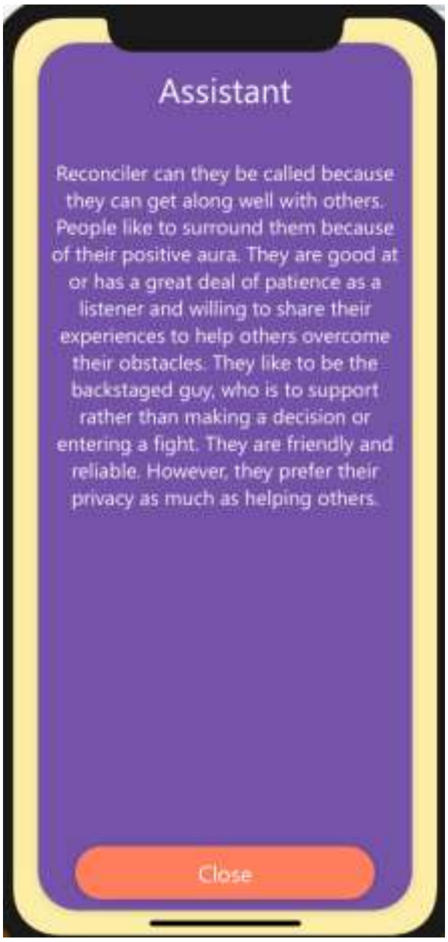


Figure 36: Final outcome.

Instead of pure text, a visualisation is included for participants to see what the machine has predicted their characteristics to be. In this case, it appears to be that a participant is more likely an assistant than other roles. In addition to that, there are three options as seen “What is assistant?”, “Incorrect”, and “Correct”. Pressing the first one will open a tab explaining what the characteristic is in details (figure 37).



Assistant

Reconciler can they be called because they can get along well with others. People like to surround them because of their positive aura. They are good at or has a great deal of patience as a listener and willing to share their experiences to help others overcome their obstacles. They like to be the backstaged guy, who is to support rather than making a decision or entering a fight. They are friendly and reliable. However, they prefer their privacy as much as helping others.

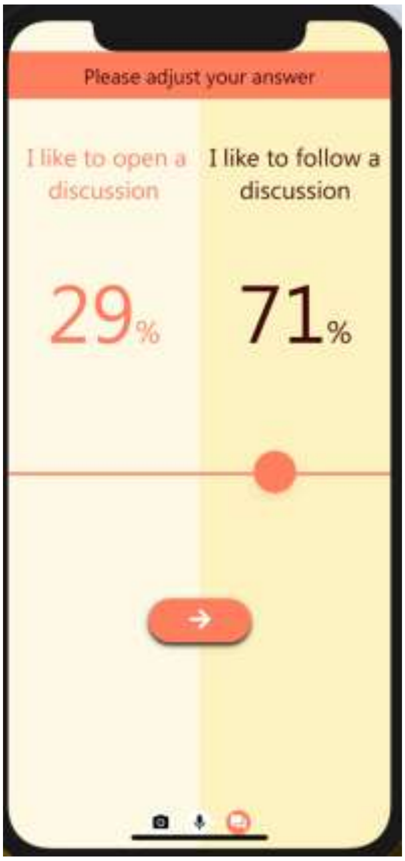
Figure 37: Explanation when pressing “What is Assistant”.

The information here is generated based on the MBTI readings.

If the answer is not of satisfaction, user can apparently select “Incorrect” to adjust the answer again. The questionnaire consists of the seven paired questions, which are:

- I like to open a discussion – I like to follow a discussion,
- Deliberate – Spontaneous,
- Independent – dependent,
- Emotional – Numerical,
- Intuitive – Rational,
- Analytical – Attempting,
- Compromising – Leading.

Figure 38 shows how user can answer the questions in the questionnaire.



Please adjust your answer

I like to open a discussion

I like to follow a discussion

29%

71%

Figure 38: An example from the questionnaire.

If everything now is as desired, the result will once again calculated and another graph will be generated, as shown in the figure 39.

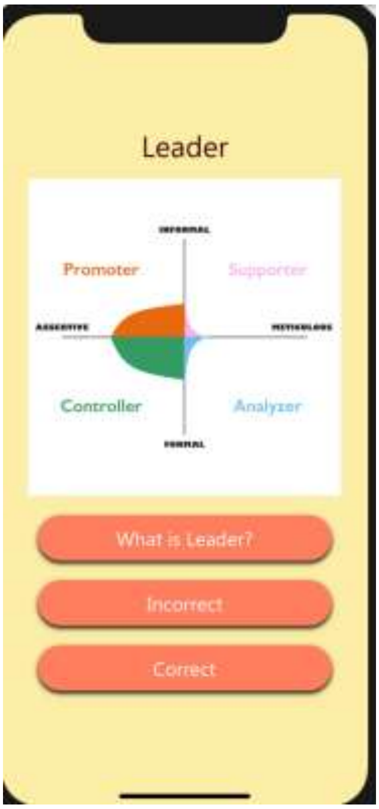


Figure 39: Result after adjusting.

Finally, once “Correct” button is triggered, the app will redirect itself back to the first page, which was introduced in the figure 31.

6 DISCUSSION

One of the problems occurred during testing concerns the CV. Because the work is low-level, the algorithm does not usually work as expected. When tested separately, the outcome resulted in less than 50% correct out of 7 trials, for example a dollar bill picture was predicted as a motorbike. Therefore, the outcome of the app was also affected.

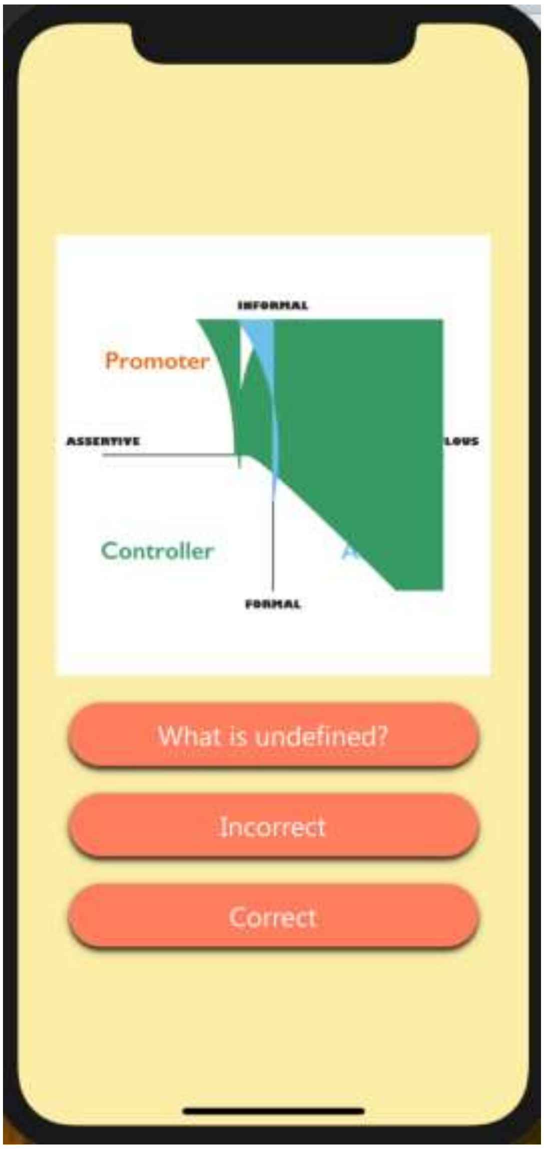
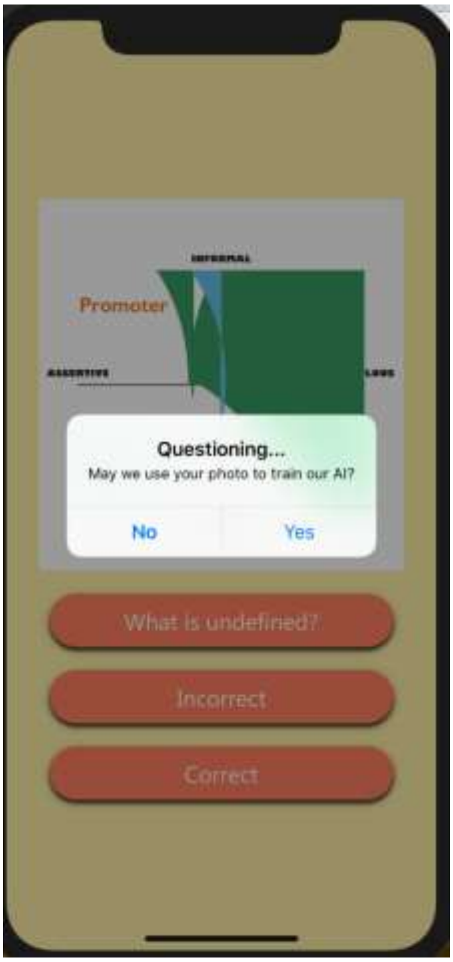


Figure 40: Error occurred while displaying the final result.

As a result, after pressing “Correct”, user is asked if s/he allows to have his/her image stored. By doing this, the database can be enriched for the machine to better its learning.



Questioning...

May we use your photo to train our AI?

Figure 41: Question prompted after pressing “Correct”.

Overall, the work was carried out within the given timeframe   with the result generated as expected.

7 CONCLUSION

The goal of the thesis was to build a cross-platform app with focus on NLP. The project used React Native as the main framework for front-end whilst Python was selected for the back-end. At the end of thesis, the app was able to perform functionalities as expected such as loading images, recording and analysing voices, making predictions and visualising the outcome.

During the developing phase, both sides (front- and back-end) of the app were consecutively revised. Regarding the front-end, the greatest question was to construct a well-designed UI that follows the minimalism requirements that are perfectly fit for small screens as iPhones but not old-fashioned visually. Furthermore, retrieving and displaying the data from back-end was another challenge that has been resolved. On the back-end, the most difficult task lied on the quality of the data as to where and what data that should be collected. In order to do that, multiple methods from conducting surveys to holding interviews, analysing and studying the results have been successfully carried out because in ML, the quality of data matters. Therefore, if this phase fails, the outcome will be consequently disregarded.

The codes in both sides were analysed continuously so that the readability – code is readable and the syntax is not chunky – must be followed. This allows programmers to read and fix the code should an issue emerges or changes to be implemented. As a result, the work was divided into multiple functions to keep track of easily.

There was always a small test when a new function was created to remain the integrity and correctness as desired. Eventually, a final test was carried out to evaluate the whole functionality of the app. The results initially were as expected and as a good starting point for further developments. In case of advacements, studies in psychology, human behaviour as well as linguistics are worth taking into consideration. In addition, the amount of qualified data plays an important role for the predictions to be the most accurate that can eventually benefit users.

REFERENCES

Altexsoft. 2018. The Good and the Bad of ReactJS and React Native. [online] Available at: <https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-reactjs-and-react-native/> [Accessed 1 October 2019]

Brownlee, J., 2019a. A Gentle Introduction to Computer Vision. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/what-is-computer-vision/> [Accessed 1 October 2019]

Brownlee, J., 2019b. 9 Applications of Deep Learning for Computer Vision. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/applications-of-deep-learning-for-computer-vision/> [Accessed 3 October 2019]

Codecademy. n.d. What is REST? | Codecademy. [online] Available at: https://www.codecademy.com/articles/what-is-rest> [Accessed 10 January 2020]

Cohen, D. and Crabtree, B., 2006. Qualitative Research Guidelines Project. [online] Qualres. Available at: <http://www.qualres.org/HomeSemi-3629.html> [Accessed 5 January 2020]

Cjhutto. n.d. VADER-Sentiment-Analysis. [online] Github. Available at: <https://github.com/cjhutto/vaderSentiment> [Accessed 5 November 2019]

DeepAI. n.d. What is a Hyperplane? [online] DeepAI. Available at: <https://deepai.org/machine-learning-glossary-and-terms/hyperplane> [Accessed 20 February 2020]

Restfulapi.net n.d. HTTP Methods – REST API Verbs – REST API Tutorial. [online] Restfulapi. Available at: <https://restfulapi.net/http-methods/> [Accessed 10 January 2020]

Facebook Developer. 2015. React.js Conf 2015 Keynote – Introducing React Native. [online] Youtube. Available at: <https://youtu.be/KVZ-P-ZI6W4> [Accessed 15 November 2019]

Fletcher, N., 2019. Classification vs Detection vs Segmentation Models: The Differences Between Them and When to Use Each. [online] Clarifai. Available at:

<https://www.clarifai.com/blog/classification-vs-detection-vs-segmentation-models-the-differences-between-them-and-how-each-impact-your-results> [Accessed 10 October 2019]

Goldberg, Y., 2017. Neural Network Methods for Natural Language Processing. [online] Morganclaypool. Available at: <https://www.morganclaypool.com/doi/abs/10.2200/S00762ED1V01Y201703HLT037> [Accessed 10 October 2019]

Heylighen, F., Dewaele, J.-M., 1999. Formality of Language: definition, measurement and behavioral determinants. [online] Available at: <http://pespmc1.vub.ac.be/Papers/Formality.pdf> [Accessed 5 September 2019]

Harmoni, F., 2018. The History of React.js on a Timeline. [online] RisingStack. Available at: <https://blog.risingstack.com/the-history-of-react-js-on-a-timeline/> [Accessed 8 September 2019]

Hudson, R., 1994. About 37% of Word-Tokens are Nouns [online]. UVM Edu. Available at: <http://www.uvm.edu/pdodds/files/papers/others/1994/hudson1994a.pdf> [Accessed 22 October 2019]

Kollegger, E., 2018. What is Axios.js and why should I care? [online] Medium. Available at: <https://medium.com/@MinimalGhost/what-is-axios-js-and-why-should-i-care-7eb72b111dc0> [Accessed 10 October 2019]

Liew, Z., 2018. Understanding And Using REST APIs — Smashing Magazine. [online] Smashing Magazine. Available at: <https://www.smashingmagazine.com/2018/01/understanding-using-rest-api/> [Accessed 15 February 2020]

Lowe, D., 2004. Object Recognition from Local Scale-Invariant Features. [online] UBC (University of Brisith Columbia). Available at: <https://www.cs.ubc.ca/~lowe/papers/iccv99.pdf> [Accessed 3 September 2019]

Manning, C., and Schutze H., 1999. Foundations of Statistical Natural Language Processing [online]. Available at: <https://www.cs.vassar.edu/~cs366/docs/Manning_Schuetze_StatisticalNLP.pdf> [Accessed 25 September 2019]

Official Statistics of Finland (OSF): Use of information and communications technology by individuals [e-publication]. [online] Tilastokeskus. Available at: <http://tilastokeskus.fi/til/sutivi/2018/sutivi_2018_2018-12-04_tie_001_en.html?ad=notify> [Accessed 11 May 2020]

Oracle. n.d.. What is Big Data. [online] Oracle. Available at: <https://www.oracle.com/big-data/guide/what-is-big-data.html> [Accessed 25 September 2019]

Oxford's Learner Dictionary. n.d. [online] Oxford Learner Dictionary. Available at: <https://www.oxfordlearnersdictionaries.com/definition/american_english/formality> [Accessed 2 January 2020]

Rouse, M., 2019. What is a RESTful API (REST API) and How Does it Work? [online] SearchAppArchitecture. Available at: <https://searchapparchitecture.techtarget.com/definition/RESTful-API> [Accessed 1 February 2020]

SAS. n.d. Natural Language Processing (NLP) What it is and why it matters. [online] SAS. Available at: <https://www.sas.com/en_us/insights/analytics/what-is-natural-language-processing-nlp.html> [Accessed 18 November 2019]

Serviceobjects. n.d. Why REST is So Popular. [online] Available at: <https://www.serviceobjects.com/resources/articles-whitepapers/why-rest-popular> [Accessed 23 January 2020]

Shoutem. 2016. A brief history of React Native. [online] Medium. Available at: <https://medium.com/react-native-development/a-brief-history-of-react-native-aae11f4ca39> [Accessed 7 October 2019]

Singh, R., n.d. Computer Vision - An introduction. [online] Towards Data Science. Available at: <https://towardsdatascience.com/computer-vision-an-introduction-bbc81743a2f7> [Accessed 15 November 2019]

Tfidf. n.d. Terms Frequency – Inverse Document Frequency. [online] Tfidf. Available at: <http://www.tfidf.com/> [Accessed 15 January 2020]

The Myers & Briggs Foundation, n.d. MBTI Basics [online]. Available at: <https://www.myersbriggs.org/my-mbti-personality-type/mbti-basics/> [Accessed 11 May 2020]

DATA STATISTICS AND DATA CURATION APPENDIX

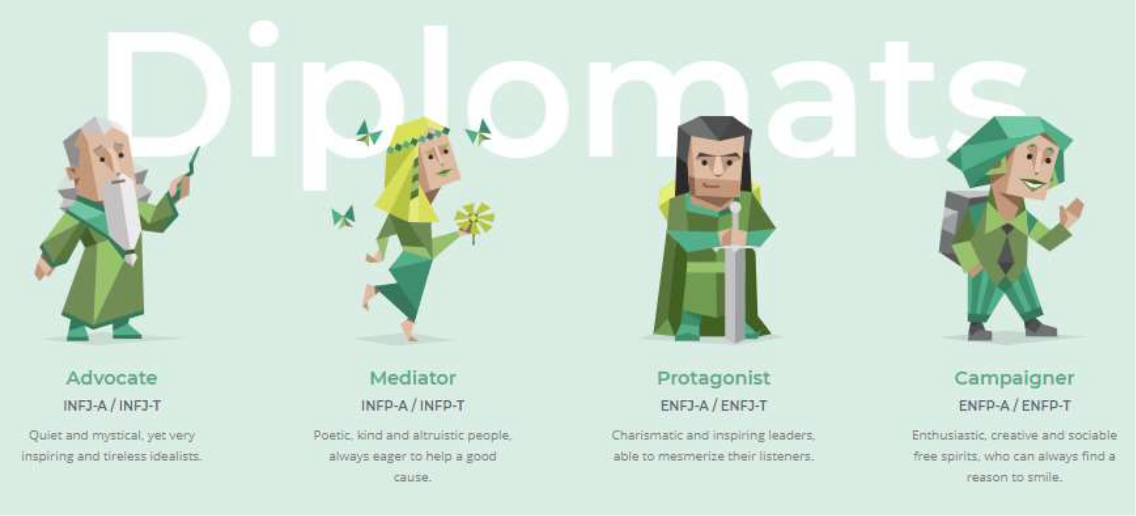
Analysts

Architect
INTJ-A / INTJ-T
Imaginative and strategic thinkers, with a plan for everything.

Logician
INTP-A / INTP-T
Innovative inventors with an unquenchable thirst for knowledge.

Commander
ENTJ-A / ENTJ-T
Bold, imaginative and strong-willed leaders, always finding a way – or making one.

Debater
ENTP-A / ENTP-T
Smart and curious thinkers who cannot resist an intellectual challenge.



Diplomats

Advocate
INFJ-A / INFJ-T
Quiet and mystical, yet very inspiring and tireless idealists.

Mediator
INFP-A / INFP-T
Poetic, kind and altruistic people, always eager to help a good cause.

Protagonist
ENFJ-A / ENFJ-T
Charismatic and inspiring leaders, able to mesmerize their listeners.

Campaigner
ENFP-A / ENFP-T
Enthusiastic, creative and sociable free spirits, who can always find a reason to smile.

Sentinels

Logistician
ISTJ-A / ISTJ-T
Practical and fact-minded individuals, whose reliability cannot be doubted.

Defender
ISFJ-A / ISFJ-T
Very dedicated and warm protectors, always ready to defend their loved ones.

Executive
ESTJ-A / ESTJ-T
Excellent administrators, unsurpassed at managing things – or people.

Consul
ESFJ-A / ESFJ-T
Extraordinarily caring, social and popular people, always eager to help.

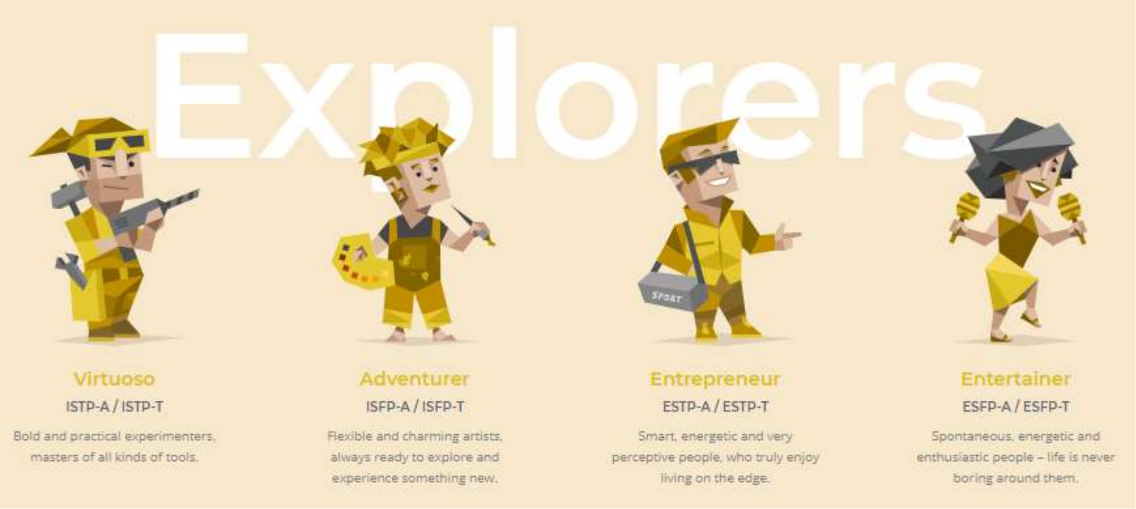
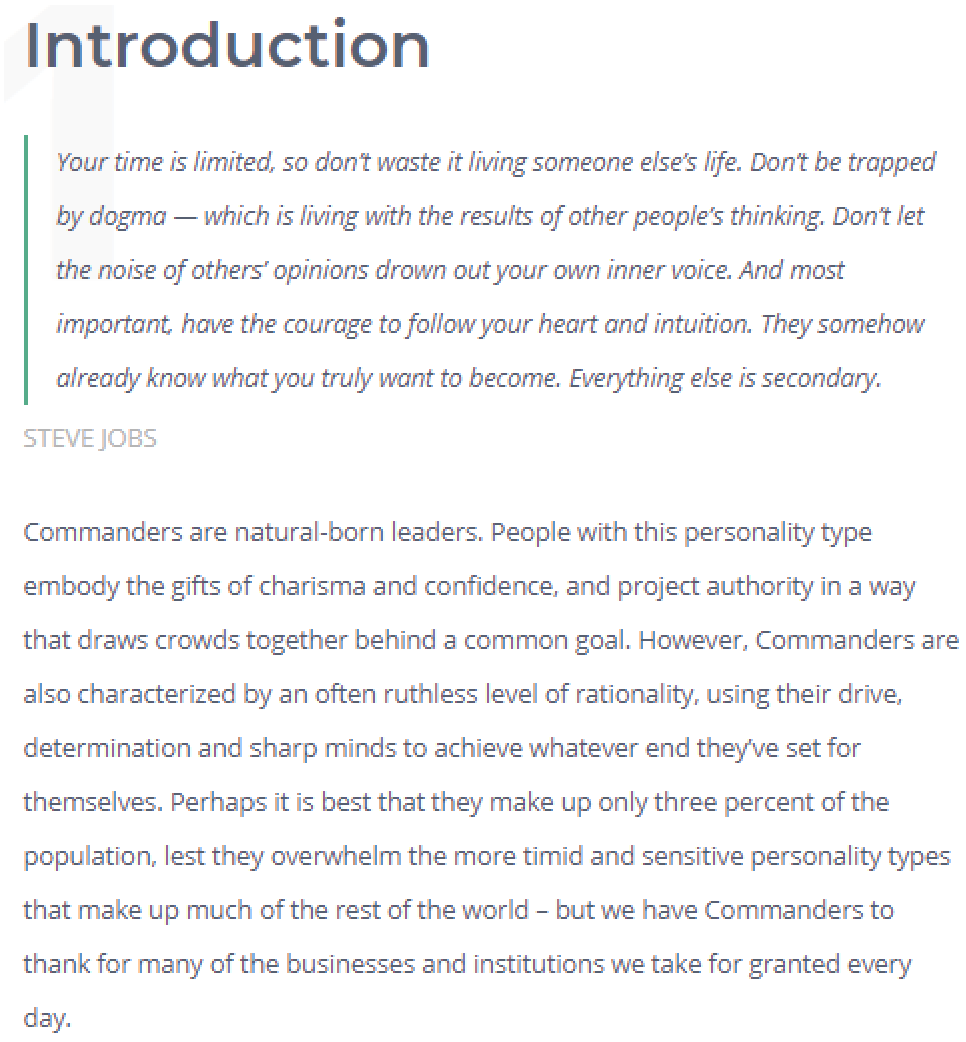

Explorers
SPORT
Virtuoso
ISTP-A / ISTP-T
Bold and practical experimenters, masters of all kinds of tools.
Adventurer
ISFP-A / ISFP-T
Flexible and charming artists, always ready to explore and experience something new.
Entrepreneur
ESTP-A / ESTP-T
Smart, energetic and very perceptive people, who truly enjoy living on the edge.
Entertainer
ESFP-A / ESFP-T
Spontaneous, energetic and enthusiastic people – life is never boring around them.

Figure 1: 16 Personality Types of MBTI.



Introduction

> *Your time is limited, so don't waste it living someone else's life. Don't be trapped by dogma — which is living with the results of other people's thinking. Don't let the noise of others' opinions drown out your own inner voice. And most important, have the courage to follow your heart and intuition. They somehow already know what you truly want to become. Everything else is secondary.*

STEVE JOBS

Commanders are natural-born leaders. People with this personality type embody the gifts of charisma and confidence, and project authority in a way that draws crowds together behind a common goal. However, Commanders are also characterized by an often ruthless level of rationality, using their drive, determination and sharp minds to achieve whatever end they've set for themselves. Perhaps it is best that they make up only three percent of the population, lest they overwhelm the more timid and sensitive personality types that make up much of the rest of the world – but we have Commanders to thank for many of the businesses and institutions we take for granted every day.

Figure 2: Introduction of Commander. (Source: 16personalities.com/entj-personality)

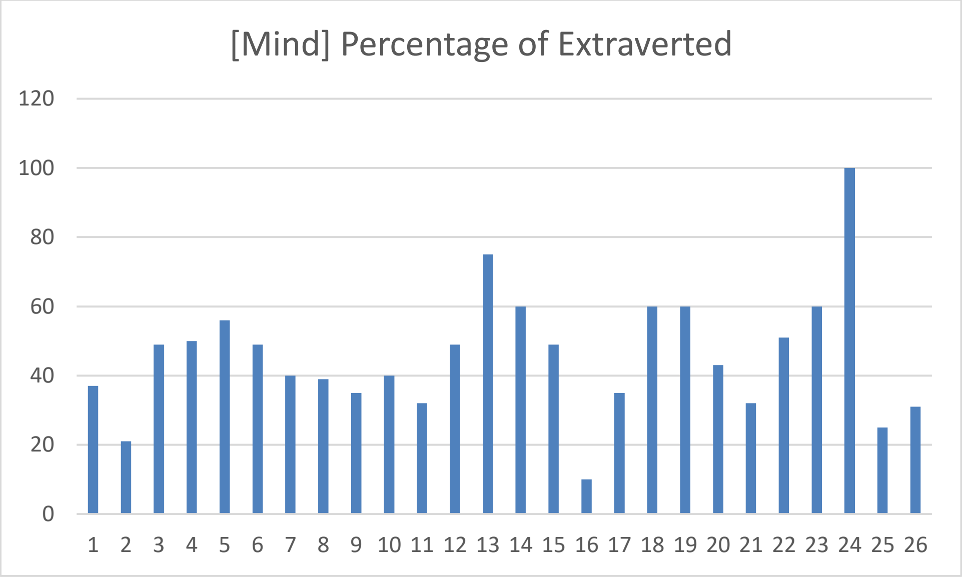


Figure 3: Percentage of Extraverted.

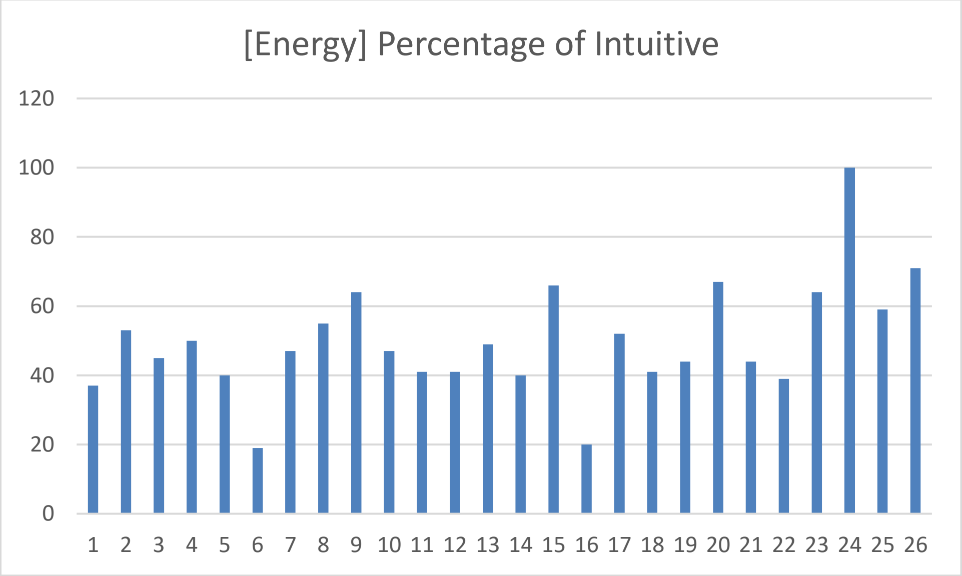


Figure 4: Percentage of Intuititive.

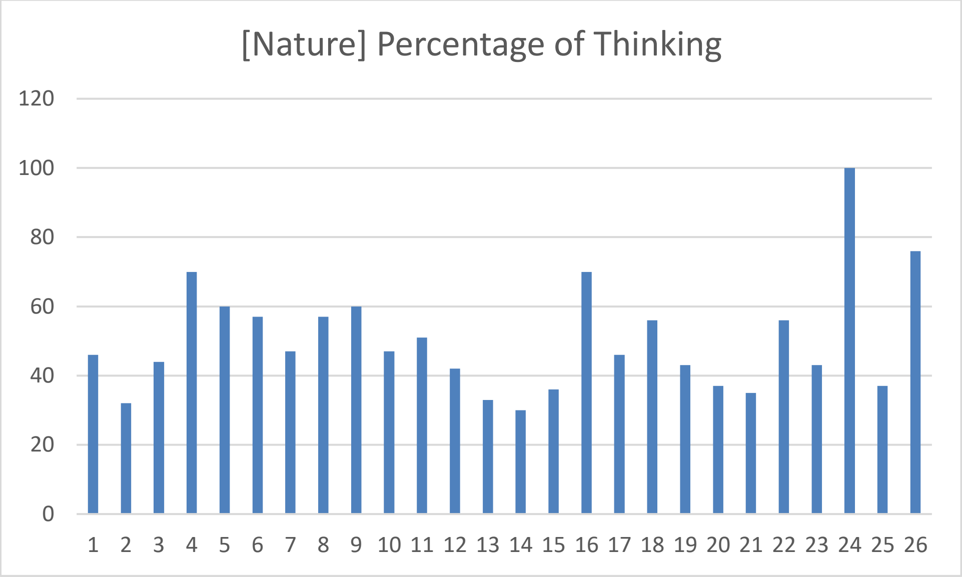


Figure 5: Percentage of Thinking.

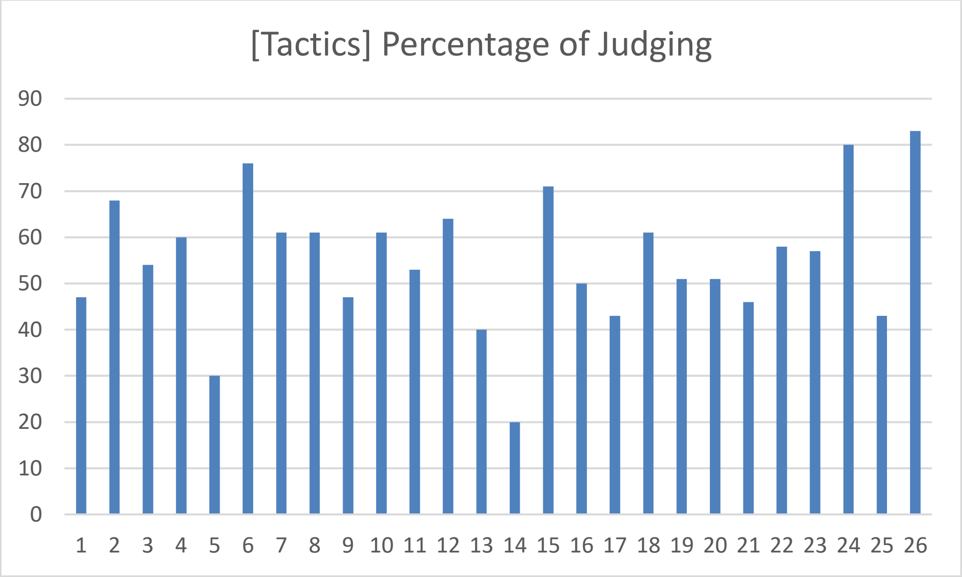


Figure 6: Percentage of Judging.

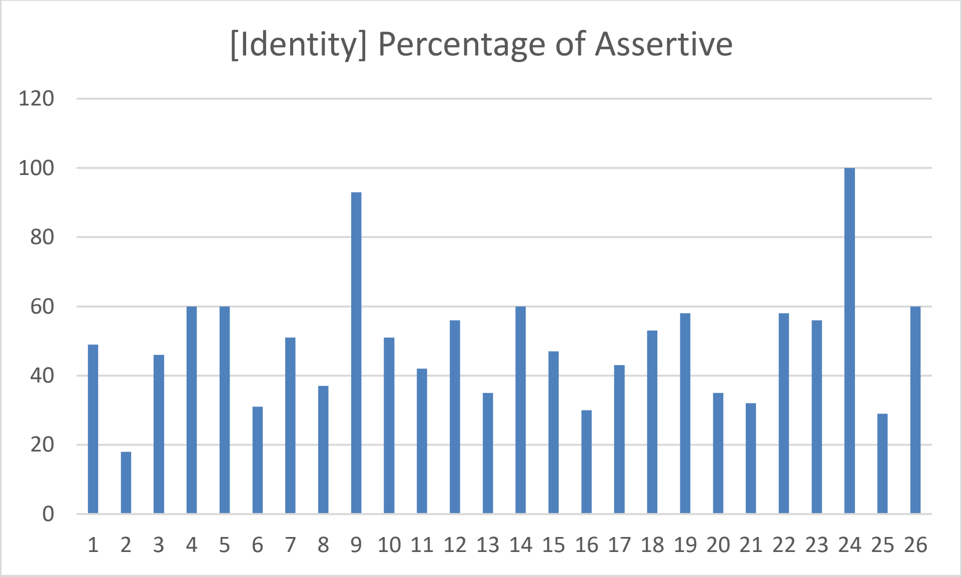


Figure 7: Percentage of Assertive.

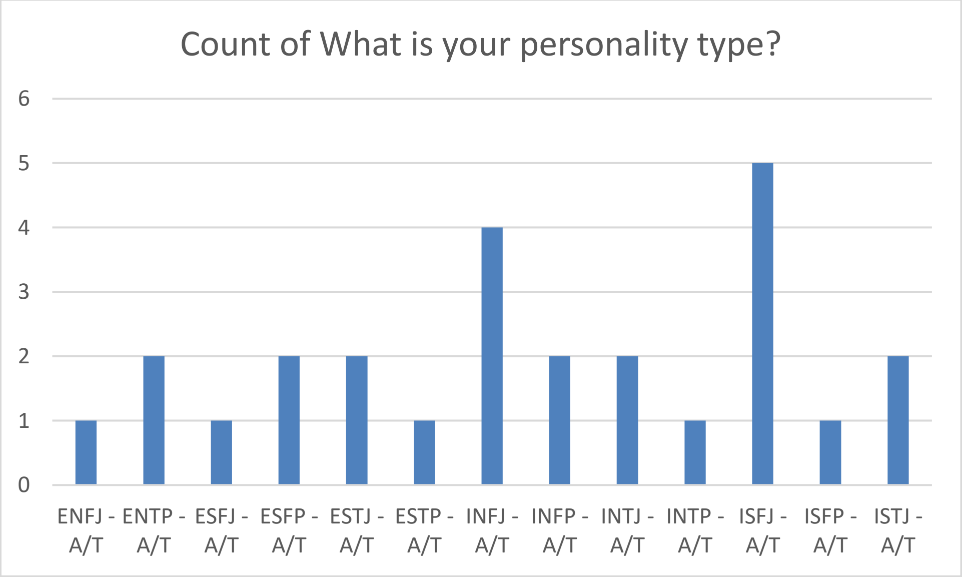
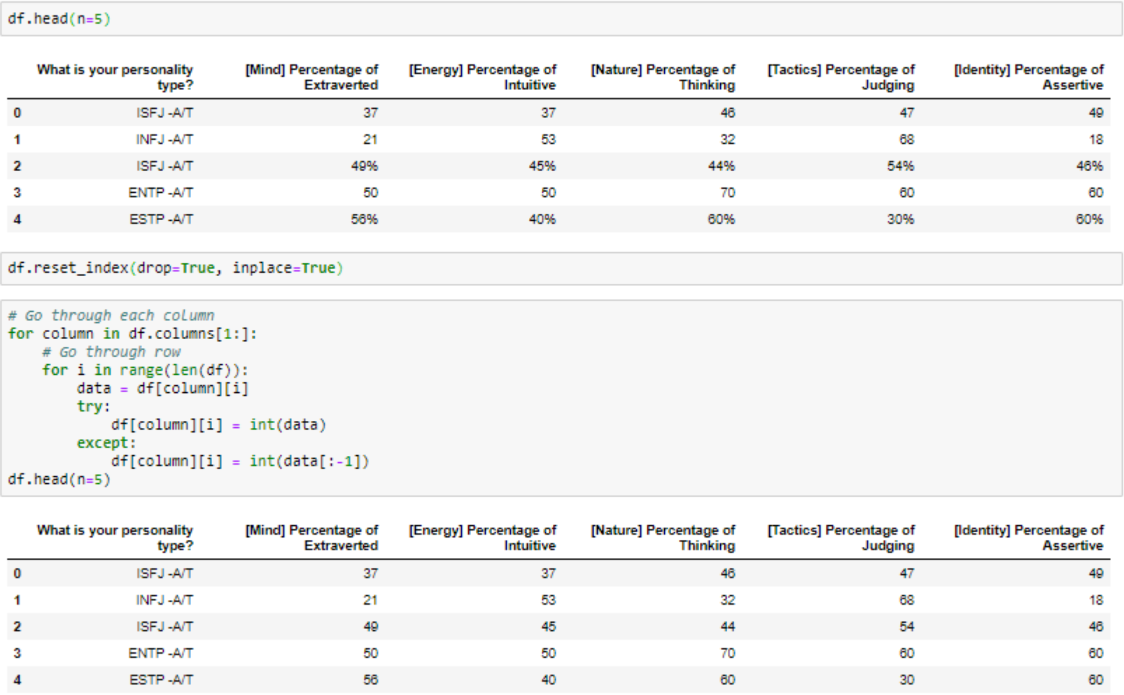


Figure 8: Count of personality type from the sample.



```
df.head(n=5)
```

	What is your personality type?	[Mind] Percentage of Extraverted	[Energy] Percentage of Intuitive	[Nature] Percentage of Thinking	[Tactics] Percentage of Judging	[Identity] Percentage of Assertive
0	ISFJ -A/T	37	37	46	47	49
1	INFJ -A/T	21	53	32	68	18
2	ISFJ -A/T	49%	45%	44%	54%	46%
3	ENTP -A/T	50	50	70	60	60
4	ESTP -A/T	56%	40%	60%	30%	60%

```
df.reset_index(drop=True, inplace=True)
```

```
# Go through each column
for column in df.columns[1:]:
    # Go through row
    for i in range(len(df)):
        data = df[column][i]
        try:
            df[column][i] = int(data)
        except:
            df[column][i] = int(data[:-1])
df.head(n=5)
```

	What is your personality type?	[Mind] Percentage of Extraverted	[Energy] Percentage of Intuitive	[Nature] Percentage of Thinking	[Tactics] Percentage of Judging	[Identity] Percentage of Assertive
0	ISFJ -A/T	37	37	46	47	49
1	INFJ -A/T	21	53	32	68	18
2	ISFJ -A/T	49	45	44	54	46
3	ENTP -A/T	50	50	70	60	60
4	ESTP -A/T	56	40	60	30	60

Figure 9: Data Inspection and Data Curation.