



Expertise  
and insight  
for the future

Marco Paulo Castro

# Machine Learning in Depth

How People Perceive their Future with Thinking Computer Systems

Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Bachelor's Thesis

24 September 2019

Author Title	Marco Paulo Castro Machine Learning in Depth
Number of Pages Date	20 April 2019
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Professional Major	Software Engineering
Instructors	Janne Salonen, Head of School
<p>Machine Learning and Artificial Intelligence have undoubtedly been the buzzwords in the field of technology and science. Though they have been around for some for approximately 70 years, it has been gaining more widespread attention and popularity in recent years because of its limitless possibilities, applications, and implementations.</p> <p>The objective of the study was to gain insights from people whether they are optimistic or pessimistic about their future with Machine Learning / Artificial Intelligence. Achieving this objective meant building a simple web application to showcase some machine learning implementations in the browser and enquiring people with and without an IT background whether they were excited or worried about this hotly debated subject. The data was then analysed and divided into three parts.</p> <p>The outcome was a working single page web application using JavaScript and its framework in gathering data through a questionnaire type of survey. Data visualization of the results is also available right after participating to the said survey.</p> <p>The primary objective of the study was met. This thesis provides a clear perception that people are optimistic as to their future with machine learning / artificial intelligence. An additional outcome was having a web application proved to be successful in gathering more data as it can reach a wider audience.</p>	
Keywords	Machine Learning, Artificial Intelligence, JavaScript

# Contents

## List of Abbreviations

1	Introduction	1
2	Theoretical Background of Machine Learning and JavaScript	2
2.1	Machine Learning	3
2.2	JavaScript	4
2.3	TensorFlow.js and ML5.js	7
3	Research Methodology	8
3.1	Research Design	8
3.2	Data Gathering	8
3.3	Research Instrumentation and Sampling Design	10
3.4	Statistical Treatment	10
4	Web Application Development Technologies and Tools	10
4.1	Front-end Development	11
4.2	Back-end Development	11
4.3	Database	11
4.4	Text Editor and Command Line	12
4.5	DevOps	12
4.6	Version Control	12
5	Project Implementation	13
5.1	Web Application Architecture	13
5.2	Client-side Implementation	14
5.3	Server-side Implementation	20
6	Project Outcome	23
6.1	Analysis of All Participants	23
6.2	Analysis of Participants within IT Industry	25
6.3	Analysis of Participants outside IT Industry	26
6.4	Implementation Experience	27
7	Conclusion	28

Appendices

Appendix 1. Server-side script – server.js

Appendix 2. Client-side script – Home.js

## List of Abbreviations

AI	Artificial Intelligence
ALICE	Artificial Linguistic Internet Computer Entity
ML	Machine Learning
HTML	Hypertext Markup Language
CSS	Cascading Style Sheet
REST	Representational State Transfer
HTTP	Hypertext Transfer Protocol
JS	Javascript
JSON	JavaScript Object Notation
XML	Extensible Markup Language
AJAX	Asynchronous JavaScript and XML
MERN	MongoDB, Express JS, React, Node
API	Application Programming Interface
NLP	Natural Language Processing
URL	Uniform Resource Locator
IP	Internet Protocol

## 1 Introduction

Machine learning is a very hot topic nowadays. Especially people with technological background are always talking about it in social platforms, messaging applications and even in barber shops. The prospect of a self-driving car, the idea of machines replacing healthcare workers in the future are two of the most discussed topics within machine learning. However, machine learning is still a young concept and the while hype around it has yet to be perceived as successful, how do people really view machine learning?

The objective of this study was to create a web/mobile application that showcases some machine learning capabilities wherein people are able to go to a webpage and after logging in can choose between a variety of machine learning applications such as face recognition, picture/photograph recognition, voice recognition and after experiencing it first-hand will be asked some questions about their experience and their overall perception. The data that collected is analysed and compared in different ways using tables and other methodologies.

The thesis aims to give a clearer picture of how people perceive machine learning. Is it something that they look forward to or does it make them feel unsure about the future? The data is used to draw conclusions as to whether there are relationships between a person's background to their attitude towards machine learning.

When Google introduced TensorFlow.js to the world, it opened a lot of possibilities and now it is possible to run machine learning algorithms directly on the user's device without the need to communicate to a server. This means that accessing these features can be done regardless of the user's location and the availability of internet connection. This thesis is a perfect use case scenario for a machine learning that runs in the browser as the end users can just go to the website that was built by the researcher and see some examples of machine learning without the need of a sophisticated personal computer and expensive graphic cards and other peripherals. All they need is a computer that has access to the internet and a browser and they can already participate in the study.

## 2 Theoretical Background of Machine Learning and JavaScript

This section introduces the topics and articles related to the thesis that discusses the overview, history and conceptions of Machine Learning and Artificial Intelligence.

Artificial Intelligence (referred to as AI from here on) is seeing a dramatic rise in popularity in recent years considering its funding was put on hiatus during the 1970s. People and businesses are grabbing this opportunity and putting a lot of their monies for AI research and development. AIs potential are perceived endless as it can solve complex problems and make use of the huge amount of information available in this age of digitization and innovation.

AI is a system that has the capability to learn using the method of trial and error that increases its chance to accomplish certain objectives accurately. AI then is able to construct its own judgment in doing things based on the information that was fed to it. In several cases, neural networks are given data or task to process and instructed what the outcome should be. The engineers then can modify the weights on each neuron to increase its accuracy and effectiveness. Doing this at least several million times, the AI will be ready to assess real world equivalent. The most critical part here is when the AI determines on its own which property are important. A good example is if an AI was trained to differentiate wolves from dogs, it is more likely to say it's a wolf when the background is snowy. The term used to describe this is called bias and it shows that an AI system is basically dependent on the information that was given to it.

Unsupervised learning is the favorable solution when there is an abundance of data and the programmer is uncertain on how it can be handled. In this case, the AI attempts to establish suitable patterns and clusters while scanning millions or probably billions of data. After analyzing the data, the machine provides the programmer with important associations between items. For example, Amazon and other retailers use this method to supply the end-user with applicable recommendations which are determined based on the information they possess.

AI at present is still very limited. Some implementations are self-driving cars, predicting next online purchase, sort apples based on their colors, and identifying the person behind the picture that was taken. They coined the term "Weak AI" to describe this kind of applications since it only handles a specific domain. Nonetheless, they extremely exceeds human when it comes to doing this designated tasks.

In certain cases e.g. warehouse inventory management, it is not a question of if but when will the machines takeover the tasks that is previously done by humans. This is due to the fact that machine cost dwarves those of humans as they can work 24/7 and with higher precision. The responsibility of AI there is not about the machine but the strategy in which those entities collaborate with each other.

## 2.1 Machine Learning

Machine Learning is the logical study of algorithms and arithmetical models that computer systems use to efficiently accomplish a specific task and predict outcomes without being overtly programmed. The basic premise of machine learning is to construct algorithms that can gather input data and use statistical analytics to forecast the output while continuously revising as new data becomes obtainable. Below is a brief history of Machine Learning to better understand the subject matter.

Pre 1950 – Thomas Bayes, Pierre Simon Laplace, Andrey Markov studied and developed theorems that will be essential to modern day machine learning. [3, p. 215]

1950 – Alan Turing published Computing Machinery and Intelligence in which the premise revolves around the question if “Machines can think?” [3, p. 216]

1951 – Marvin Minsky and Dean Edmonds successfully built the first neural network – a set of algorithms modeled to a human brain that are designed to recognize patterns. [4, p. 57]

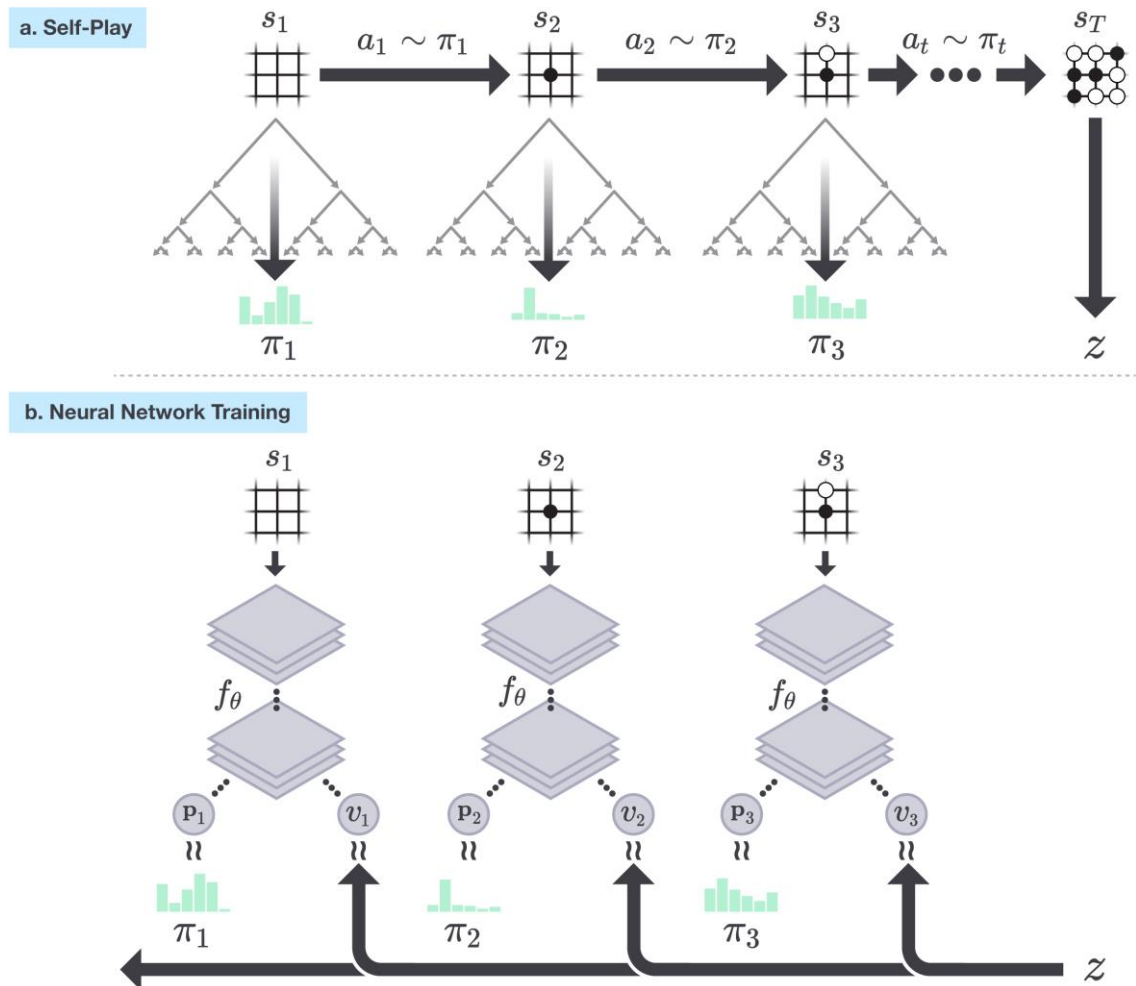
1974 – Due to failure of advancements in Machine Learning, the term AI winter was coined to describe the reduced enthusiasm and funding for developments that are related to Machine Learning and Artificial Intelligence. [7]

1997 – Deep Blue, a chess playing computer developed by IBM defeated 6-time chess world champion Garry Kasparov. [4, p. 54]

2016 – DeepMind researchers developed AlphaGo, a go playing computer who defeated the world’s top players in Go. [5]



Google Deep mind researchers have presented an algorithm built specifically on reinforcement learning without human interaction (see Figure 1). These neural networks were trained by supervised learning from human expert moves, and by reinforcement learning from self-play.



**Figure 1.** Self-play reinforcement learning in AlphaGo Zero.

Figure 1 shows how the program plays a game  $S_1, \dots, S_T$  against itself. Each position  $S_i$ , a Monte-Carlo Tree Search (MCTS) is performed using the latest neural network. The terminal position  $S_T$  is scored according to the rules of the game to calculate the game winner  $Z$ .

## 2.2 JavaScript

JavaScript was unveiled on the 4<sup>th</sup> of December 1995 by Netscape Communications Corporation and Sun Microsystems. It is an open, cross-platform object scripting

language for the creation and customization of applications on enterprise networks and the Internet. [8] It was created by Brendan Eich in 1995 when he was recruited by Netscape Communications. The programming language was inspired by Java, Scheme and Self and was initially called as Netscape's Mocha for the reason that it was originally built for the then popular Netscape Navigator. [9]

JavaScript was first standardized on June of 1997 and it was called ECMA-262 or ECMAScript. Most of the features that was included that time are still used on the current JavaScript today like first-class functions, objects, and prototypal inheritance. [10] JavaScript is a very popular and used language hence it has experienced a lot of improvements over the year and presently JavaScript is on the 9<sup>th</sup> edition or ECMAScript 2018 standardization (see Table 1)

Table 1. Brief history of ECMAScript standardization [10]

Standardization Name	Date	Some Features and Facts
ECMAScript	June 1997	-First Edition  -The Ecma International (a standards organization) has created the ECMA-262 standard which is the official specification of the ECMAScript language.
ECMAScript 4	2008	-Abandoned and was never released.
ECMAScript 6	June 2015	-Arrow function  -Classes

		<ul style="list-style-type: none"> <li>-Enhanced object literals</li> <li>-Template Strings</li> <li>-Destructuring</li> <li>-Let and Const</li> <li>-Modules and Module Loaders</li> <li>-Symbols</li> <li>-Promises</li> <li>-Reflect API</li> </ul>
ECMAScript 7	June 2016	<ul style="list-style-type: none"> <li>-Decided to release a year update</li> <li>-Array.prototype.includes</li> <li>-Exponentiation operator</li> </ul>
ECMAScript 8	June 2017	<ul style="list-style-type: none"> <li>-Async/Await functions</li> <li>-Shared memory and atomics</li> <li>-String padding</li> </ul>
ECMAScript 9	June 2018	<ul style="list-style-type: none"> <li>-Asynchronous Iteration</li> <li>-Rest/Spread properties</li> <li>-Template literal revision</li> </ul>



As shown in Table 1, ECMAScript has been updated 9 times and the current version is ECMAScript 9. In the earlier versions 1,2 and 3, the fundamental features of JavaScript were introduced, some of which are taken for granted nowadays like regular expressions, try/catch exception handling and formatting for numeric output. ECMAScript 4 was plagued with problems as the committee were divided into 2. One camp wanted to push through to a major update that will be named ECMAScript 4. The other camp however did not share the same view and wanted a minor ECMAScript 3.1 update. It took almost 8 years for the committee to land on a common ground and on July of 2008, a compromised was done in Oslo Norway. In the end, some features are added to the next version which is ECMAScript 5 and ECMAScript 4 was completely abandoned. ECMAScript 6 was a huge update from the previous version. It was initially planned to be released on 2013 but had to be moved to June of 2015. It introduced us to the newer and simpler syntaxes like arrow function which is less code than the original function declaration. ECMAScript 7 and 8 are mostly improvements and the addition of the Async/Await functions.

### 2.3 TensorFlow.js and ML5.js

Previously, it was nearly impossible to run machine learning in the browser. Some reasons for this are that it needed to be hosted on cloud services, the developer had to be proficient with programming languages such as R and python, and expensive computer peripherals. This changed when Tensorflow.js and ML5.js were introduced. TensorFlow.js is an open-source library that can be used to define, train, and run machine learning models completely in the browser using JavaScript and a high-level layer API. Running machine learning in the browser unlocks new opportunities. A good example is 2 developers having made a demo of how PACMAN can be played and controlled using the player's eyes through the computer's webcam. [12]

ML5.js is a JavaScript library as well that is built over TensorFlow.js which provides access to machine learning algorithms, task and models in the browser. Basically ML5.js is to TensorFlow.js as to Keras is to TensorFlow.

Machine Learning that works in the browser means that from an end-user's viewpoint, there is no requirement to download any libraries or drivers. Similar to the concept of Plug N Play, users have to access a website and the application is ready to run. In addition, it means that a mobile device - as long as it has a browser - can take advantage of this technology. Lastly, all the data stays on the client making it less affected by latency issues as well as for privacy and security. [12]

### **3 Research Methodology**

This chapter introduces the design of the research, the data gathering, and the research instrumentation and sampling design.

#### **3.1 Research Design**

The descriptive research design method was chosen because it seeks to describe the current status of a variable or a phenomenon and the conclusions will be done after the data has been collected and analyzed. In addition, the research is non-experimental descriptive research design for it deals about the relationship of variables such as the person's professional background and their views about the future of machine learning.

#### **3.2 Data Gathering**

The data were gathered in two approaches. The first approach was through a questionnaire. The researcher met with a group of people and show them some machine learning implementations and were asked about their name, country of residence, and if they considered themselves an IT professional. After getting the basic information of the participant, they were also asked if they were excited or worried about their future with machine learning / artificial intelligence. The second approach was through the web application that was built by the researcher. Upon landing to the homepage, users were asked again the same questions as the paper questionnaire. This time however, if the participant is using a personal computer or a laptop, they were able to see 2 machine learning implementations that runs in the browser. The people who answered through the web application were able to see a graph that shows the initial result at that time. In order to ensure that the participant only answers the survey once, their IP were stored in

the database as so the next time they visit the site, they will be redirected to the result page. After getting the data from the paper questionnaire, the information was added to the web application to obtain the final result. The gathering of data took place from the 2<sup>nd</sup> to the 15<sup>th</sup> of September. The interview type questionnaire was conducted through group interview both in person and online. The web application was deployed on the 8<sup>th</sup> of September. A link of the web application was given to people through word of mouth or messaging application like Facebook and WhatsApp (see Figure 2).

WEBSITE OF MY THESIS SURVEY: <https://marco-thesis.herokuapp.com/>

Hello guys, attached to this message is the website of my thesis survey. Please answer with your name and country of residence. For those who feel neutral about their future with Machine Learning / Artificial Intelligence, you need not to answer this questionnaire. Depending on your device, you might see links to some Machine Learning implementations that runs in the browser (Image Recognition and Object Detection). Only one answer is allowed per IP address and you will be redirected to the result page once you answered. If for some reason you are not able to visit the website, just message me here with the answer to the following:

1. Do you consider yourself as an IT Professional?  
Yes / No

2. Are you excited about your future with Machine Learning / Artificial Intelligence?  
Yes I am / No I am worried

For further reading visit :  
<https://www.dataversity.net/Future-machine-learning-artificial-intelligence/>

Thank you! ☐

Figure 2. Sample message sent to participants.

As shown in Figure 2, this is a sample message sent to participants through Facebook. The message included instruction on how to participate on the study and the URL of the web application.

### 3.3 Research Instrumentation and Sampling Design

The study used one set of questionnaires with two main parts. The first part is the personal information of the respondent and the second part is the question about whether they are excited or worried about the future of machine learning. The study examined people's perception about their future with thinking computer systems. The researcher initially planned to use quota sampling to ensure appropriate sample to be included in the study such that there are equal respondents from those who have and who does not have a background in the Information Technology but in the end used simple random sampling.

### 3.4 Statistical Treatment

The result of the study was analyzed in three ways. The first approach was a conclusion is made based on the total number of respondents and if they were excited or worried about their future with machine learning. This was just be a straight comparison of the two option. The second approach was divided into two groups – the group that have IT background and the group does not have. After obtaining and dividing the data from the two groups, the researcher calculated the percentage using the formula  $\frac{f}{n} * 100$ .  $f$  is the total number of respondents who answered that specific item.  $n$  is the total number of respondents in that group and 100 is a constant.

## 4 Web Application Development Technologies and Tools

This chapter introduces the development process from front-end to backend development of the web application the researcher built to gather and present data.

## 4.1 Front-end Development

The client-side was built using HTML, CSS and Javascript. Additionally, ReactJS was used as the library of choice. ReactJS is a popular library used in creating single page web application and user interface. It was built by Facebook to address some of the challenges related with large-scale, data-driven websites. [13, p. 1] Aside from its popularity, ReactJS was chosen because it is very small and it has a huge community that can support the development process. ChartJS was used as well to display the pie charts to represent the data gathered. It is the library of choice in presenting data in a visual manner because it is easy to use and very straightforward.

## 4.2 Back-end Development

The server-side was built using NodeJS is a free open source server-side scripting language that was built over JavaScript. This means that web applications can be written from front-end to back-end by just knowing one programming language. NodeJS uses v8, the virtual machine that powers Google Chrome which gives Node a huge improvement in performance because it removes the middleman favoring straight compilation into native machine code over implementing bytecode or by means of an interpreter. [14, p. 4] On top of NodeJS, Express – a minimal web app framework – was used to make AJAX requests from an API much easier.

## 4.3 Database

PostgreSQL was used as the database in this web application. It is a general-purpose object-relational database management structure, the most innovative open source database system. PostgreSQL was designed to run on UNIX-like platforms. It also designed in such a way that it is possible to run it on various platforms and operating system such as Mac OS X, Solaris and Microsoft Windows. PostgreSQL is chosen in this project because the developer needed a simple relational database where the structure was defined and the data were retrieved.



#### 4.4 Text Editor and Command Line

Visual studio code was chosen as the text-editor for this project. It is a lightweight but capable source code editor which runs on the computer and is available for Windows, Mac OS and Linux. This text-editor comes with built-in support for JavaScript and accepts additional packages for both ReactJS and NodeJS hence makes it a perfect editor for this project. GitBash was used in this project as the main terminal. It is an application for windows environment which provides an emulation layer for a Git command line experience. GitBash was also used to run the development server on the sideline during the development phase.

#### 4.5 DevOps

Deployment of the web application was done using Heroku. Heroku is a platform as a service constructed on a managed container system, with combined data services and a useful system for deploying and running up-to-date apps. Heroku developer experience is an app-centric approach for software delivery, integrated with today's most popular creator tools and workflows. [16] This platform was chosen simply because it saves a lot of time. In order for a person to run a web app, they need server, whether it is a physical one or a virtual one. The next one is configuring a web server followed by an application server and if needed a database server. Heroku helps with all of those steps so that the developer/engineer does not have to configure everything manually.

#### 4.6 Version Control

Version control systems are a group of software tools that aids an individual and/or a team to accomplish changes to source code over time. Version control preserves track of all the modifications that the developer/s made and save it to a different kind of a database. If a mistake is made, developer/s can easily rollback to the previous working version using very little codes. [20] The researcher chose GitHub to be the version control on this project as it is easy to use and is available for free.

## 5 Project Implementation

This section discusses the process of web application development from designing the application architecture to the actual coding and deployment.

### 5.1 Web Application Architecture

The web application architecture defines the connections between application, databases, and middleware systems on the web (see Figure 3). This facilitates design, development, and deployment of the web application.

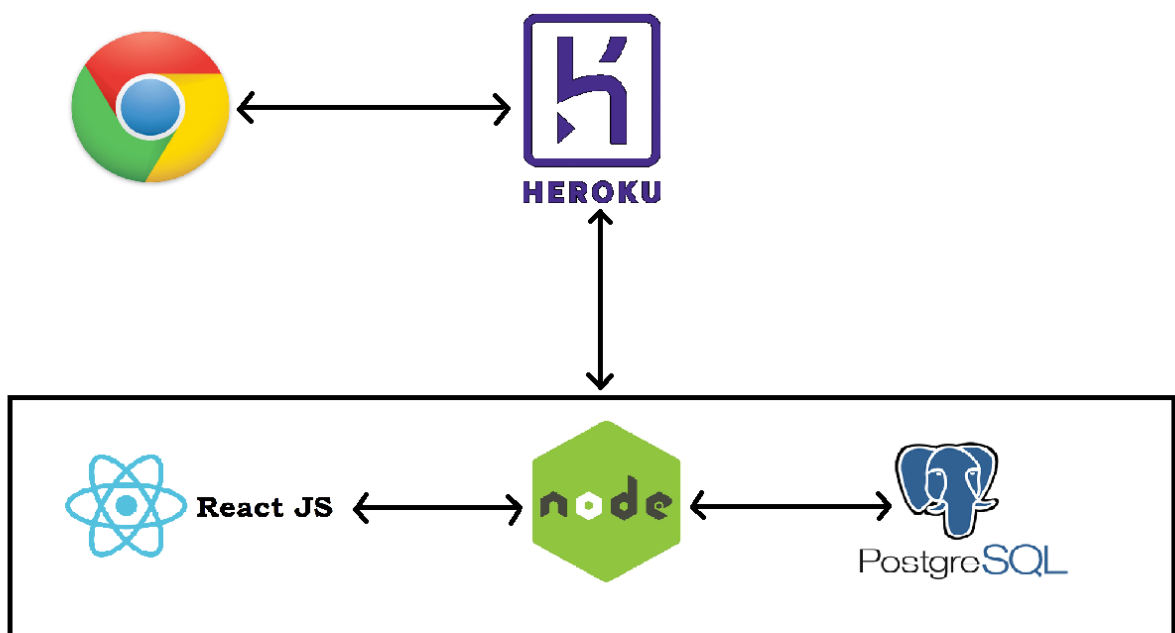


Figure 3. Web Application Architecture Diagram

Based on the diagram illustrated in Figure 3, the end-user / participant who has access to a device capable of connecting to the internet and has browser capability navigates to <https://marco-thesis.herokuapp.com> and is requested to provide basic information about their selves and a couple of questions. [Appendix 2] After which when the submit button is clicked, the single-page application sends the information to the backend which is handled by NodeJS. NodeJS then processed the data and saves it to the database which is PostgreSQL. The end-user however will be redirected to the result page that shows a doughnut pie chart that illustrates the number of participants and their answers. The graph uses chartJS, a powerful JavaScript framework that focuses on data visualization.

It works in a way that it sends a get request from the API and the backend request those data from the database.

## 5.2 Client-side Implementation

The client-side was carried out using ReactJS. It is a powerful and popular single page application framework. To start the implementation the researcher needs to install react. *npm install create react-app* is the command needed to start the react project. This web application however used a ready-made react application with express. [17]

*-git clone [https://github.com/bradtraversy/react\\_express\\_starter.git](https://github.com/bradtraversy/react_express_starter.git)* - This command should be executed on the command line after the developer create a folder for the project. This clones the repository to the local computer. This was chosen to save time in configuring everything from scratch.

*-npm install* – This command installs a package and any package it depends on. By default, npm installs all the modules listed as dependencies on package.json file. This is however for the server side and more on the client-side next.

*-npm run client-install* – This command is the same as npm install however it installs all the modules that can be found on *client/package.json* file.

*-npm run server and npm run client* – These two commands run on the development and client servers. By default, react applications run in localhost port 3000

Above are the steps used by the researcher to build the web application. Pre-requisites included knowledge of navigating through GitHub and the use of terminal.

The folder structure of the client-side development environment. Node modules folder contains all the packages that was installed using the `npm run client-install` command (see Figure 4)

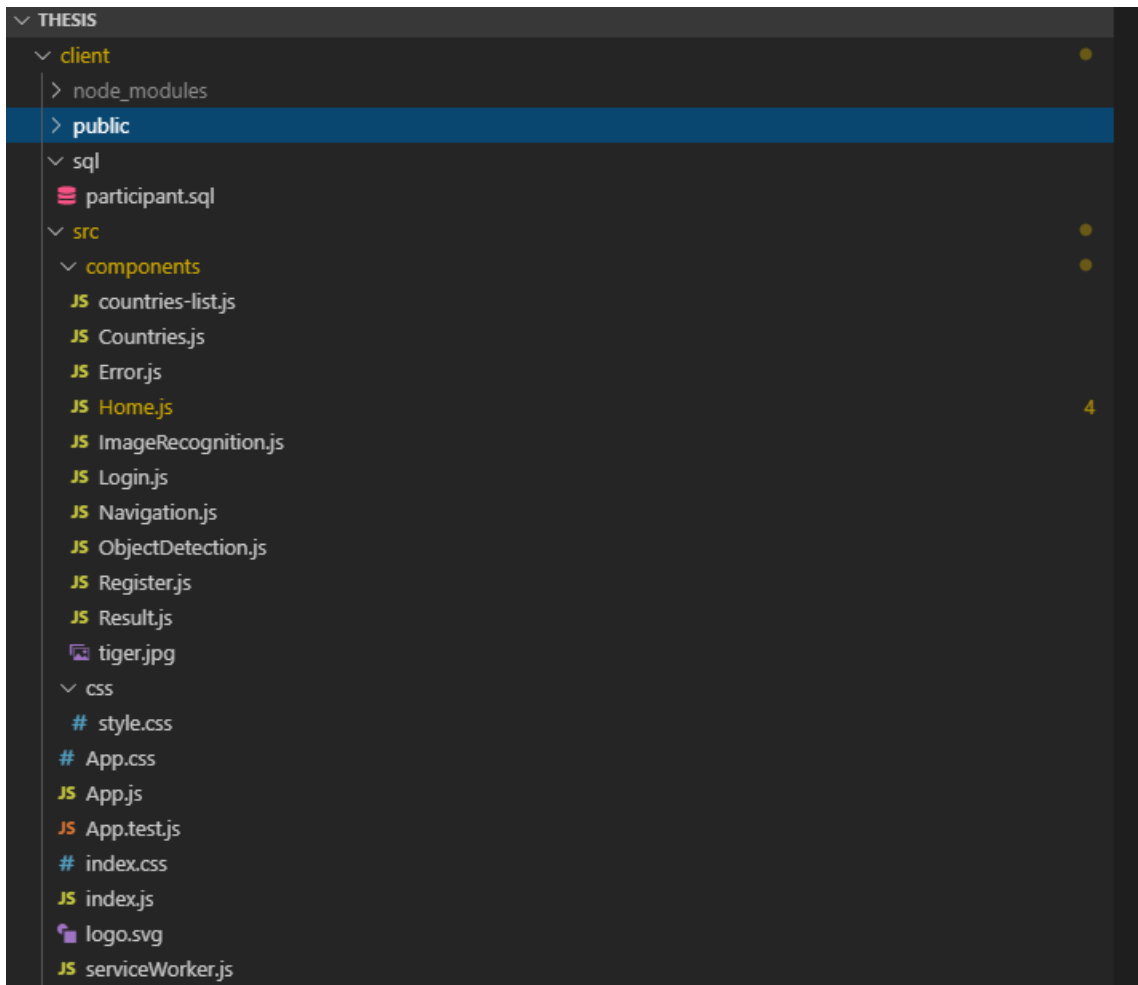


Figure 4. Client-side Folder Structure

Figure 4 shows the public folder which contains the static files such as images that is served on the web application. Under SQL folder is the participant.sql, this file contains the code to run to create a database table that stores the information about the participants. Src folder comprises the different components of the web application. The beauty of react is that components can be created and re-used all over the web application. This makes the application easier to organize and understand since there is no need to rewrite everything from scratch. The styling of the web application can be found under the css folder. Some stylings can also be found in different components as it is a common practice in react to add inline stylings in the component itself.

The homepage contains the basic information of the participant and the questionnaire. It also provide some privacy consent and further reading for people to understand the topic at hand (see Figure 5).

Home Image Recognition Object Detection

NAME / NICKNAME:

COUNTRY:  
Select...

Do you consider yourself as an IT professional?  
 Yes  No

Are you excited about your future with Machine Learning / Artificial Intelligence?  
 Yes I am excited 😄  No I am worried 😞

PRIVACY CONSENT: In Accordance with international law, all data provided here (name, country, etc) will only be used in this research and will be destroyed as soon as the study is over.

NO IMAGE AND/OR VIDEO IS TAKEN ON THIS WEBSITE

Further reading:  
<https://www.dataversity.net/Future-machine-learning-artificial-intelligence/>

Figure 5. Web Application Home Page

Figure 5 displays the home page of the application. This is done by rendering the Home.js component discussed in Figure 2. The navigation at the top of the page has three links namely Home, Image Recognition and Object Detection.

Chart JS was chosen to visualize the result in a doughnut chart. The graph is also capable of filtering the results based on their profession (see Figure 6).

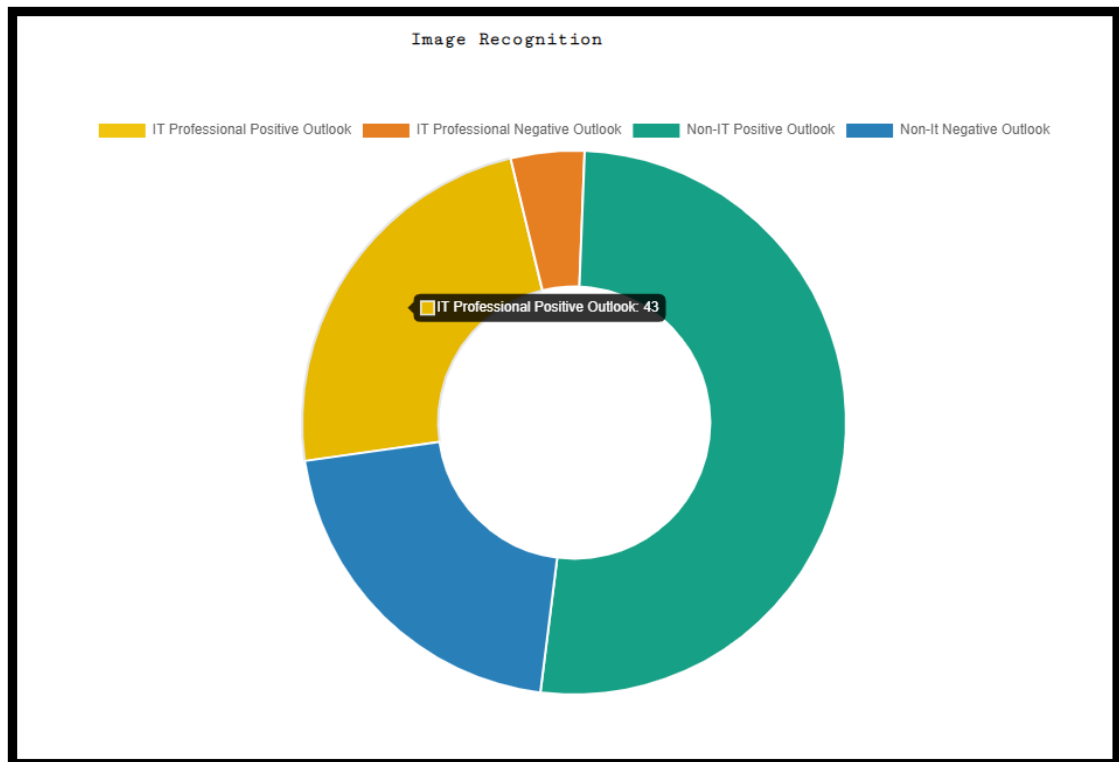


Figure 6. Result Page

Figure 6 illustrates the result of the data gathered. The graph is divided into 4 namely IT professionals with positive outlook about their future with machine learning, IT professionals with negative outlook about their future with machine learning, Non-IT professionals with positive outlook about their future with machine learning, and lastly Non-IT professionals with negative outlook about their future with machine learning. .

The image classification component was rendered using ReactJS and the algorithm was processed by TensorFlow.JS. This component used pre-trained model [18] to classify an image (see Figure 7).



Figure 7. Image Classification Page

Figure 7 above shows the Image Classification component. In the web application, the end user can add an image they have on their device and let the application predict the object that they uploaded. It does not save any image in the database rather analyzes it in the browser. This page is one of the two components of the web application that showcased machine learning that runs in the browser.

Figure 8 shows the real-time object detection algorithm to classify and locate different objects.

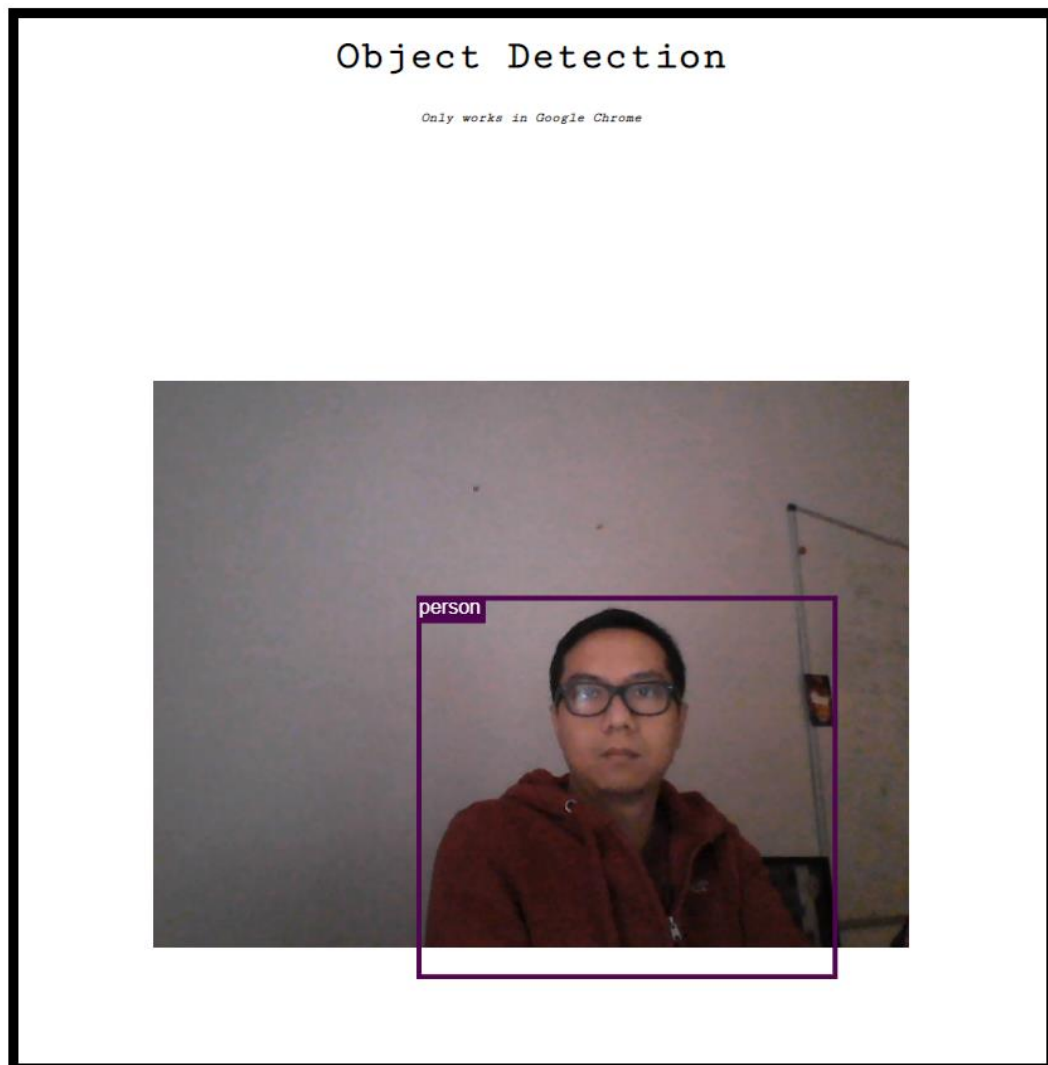


Figure 8 above is the second component that showcases machine learning that runs in the browser. It is real time multi object detection algorithm that is called YOLO short for You Only Look Once. This implementation however is much slower than the real YOLO algorithm as it uses the browser and the webcam/camera to detect objects.

Figures 7 and 8 are included in the web application to give the participants of the survey some idea on how machine learning works although it is very limited as it only runs in the browser and uses OpenGL to handle the processing power instead of a dedicated graphics card.



### 5.3 Server-side Implementation

The server-side or backend was carried out using NodeJS and Express. Express is a web framework that let us to construct a web application to handle several different HTTP requests at a definite URL. [19] Express helps us respond to requests with route support so that they may respond to specific URLs. NodeJS and Express was chosen to perform some tasks like running the server and communicating with the database to build an API.

Figure 9 shows the command to run to install express using the terminal.

```
$ npm install express
npm WARN react-flags-select@1.1.10 requires a peer of react@15.x but none is installed. You must install peer dependencies yourself.
npm WARN rollup-plugin-visualizer@1.1.1 requires a peer of rollup@>=0.60.0 but none is installed. You must install peer dependencies yourself.
npm WARN THESIS@1.0.0 No description
npm WARN THESIS@1.0.0 No repository field.

+ express@4.17.1
updated 1 package and audited 479 packages in 2.566s
found 0 vulnerabilities
```

Figure 9. Express JS installation

To install express, simply type in `npm install express` on the terminal. A successful installation will show a message that the package has been installed and the user can start working on the project.

Listing 1 shows the boilerplate needed to run the web application on localhost port 5000.

```
1  const express = require('express');
2  const app = express();
3  const port = 5000;
4  const bodyParser = require("body-parser");
5  const cookieSession = require('cookie-session');
6  const db = require('./db')
7
```

Listing 1. Declaring the modules in server.js, copied from Appendix 1

Declaring the modules that were installed using `npm install` happens in the server.js file as shown in Listing 1.

Figure 10 shows the boilerplate needed to include PostgreSQL to the web application.

```
2  const pg = require('pg');
3
4  const pg_client = new pg.Client({
5    user: 'postgres',
6    database: 'postgres',
7    password: 'postgres',
8    port: 5432,
9    idleTimeoutMillis: 60,
10 });
11
12 pg_client.connect();
13
14 function addParticipant(admin, ip, name, country, is_it, is_positive) {
15   return pg_client.query(
16     `INSERT INTO participant (is_admin, ip, name, country, is_it, is_positive)
17     VALUES ($1, $2, $3, $4, $5, $6);`,
18     [admin, ip, name, country, is_it, is_positive]
19   );
20 }
21
22 function getParticipant(ip) {
23   return pg_client.query(
24     `SELECT * FROM PARTICIPANT WHERE ip = $1`, [ip]
25   )
26 }
27
28 const getResult = () => {
29   return pg_client.query(
30     `SELECT COUNT (CASE WHEN (is_it = 'true' AND is_positive = 'true')
31     then 'it_positive' END) AS it_positive,
32     COUNT(CASE WHEN (is_it = 'true' AND is_positive = 'false')
33     then 'it_negative' END) AS it_negative,
34     COUNT(CASE WHEN (is_it = 'false' AND is_positive = 'true')
35     then 'nit_positive' END) AS nit_positive,
36     COUNT(CASE WHEN (is_it = 'false' AND is_positive = 'false')
37     then 'nit_negative' END) AS nit_negative
38     FROM participant`
39   )
40 }
41
42 module.exports = {
43   addParticipant,
44   getParticipant,
45   getResult
46 };
```

Figure 10. PostgreSQL sample commands

The database in this project was handled by PostgreSQL. As shown in Figure 10, line 2 shows how to include the *pg(postgres)* module to the project.

Line 4-10 - Creates a pg client which is required to build a connection to the database and to declare configuration for the postgres database.

Line 12 – Establish a connection between the server and the database. This is the command use to start the communication between the main backend to the database.

Line 14-19 – Function *addParticipant* is the first of the three database query function used in this web application. The main task of this function is adding an entry to the database once a participant submitted his/her answers from the web application.

Line 22-26 – The second function is called *getParticipant*. This function handles the checking of the IP address of the participant to make sure that only one user per IP address can vote to avoid multiple submission. This function is called immediately when the user goes to the application and if the IP has already been added to the database, this will return the value to the server and the server will immediately recognize that this user has already participated with the survey and will redirect them to result page.

Line 27-39 – The last database function is called *getResult*. This query is run every time the *result* page is visited. The main functionality of getResult is for the server to fetch all the data from the database and send it to chartJS for data visualization.

The list above discusses steps are the ones used by the developer in building the database.

```
1 CREATE TABLE IF NOT EXISTS participant(  
2     id SERIAL PRIMARY KEY,  
3     ip INET,  
4     is_admin BOOLEAN DEFAULT FALSE,  
5     name VARCHAR(200) NOT NULL,  
6     country VARCHAR(250) NOT NULL,  
7     is_it BOOLEAN,  
8     is_positive BOOLEAN  
9 );
```

Figure 11. PostgreSQL creating table

Creating a database is shown in Figure 11. In a relational database like PostgreSQL, data is stored in a table like structure which is very easy to read and understand.

Figure 12 shows the sample data from a random participant as seen in the terminal.

```
id | ip      | is_admin | name      | country   | is_it | is_positive
---+---+---+---+---+---+---
 4 | 127.0.0.1 | f        | Margeret | Burkina Faso | f     | f
(1 row)
```

Figure 12. Postgres table as seen in the terminal

As shown in Figure 12, the postgres table shows the relationship between data. This makes it really flexible and easy to understand.

## 6 Project Outcome

This chapter summarizes and analyzes the result of the study based on the gathered data. As discussed in Chapter 3.4 Statistical Treatment, this section presents the evaluation of the project from 3 perspectives:

1. An analysis of all the participant's perspective of their future with Machine Learning / Artificial Intelligence.
2. An analysis of people with IT background of their future with Machine Learning / Artificial Intelligence.
3. An analysis of people without IT background of their future with Machine Learning / Artificial Intelligence.

This chapter also discusses the researcher's total experience in implementing the project from developing the web application to the gathering of the data.

### 6.1 Analysis of All Participants

The analysis of all the participants' perspective of their future with Machine Learning / Artificial Intelligence is illustrated in the doughnut chart in Figure 13.

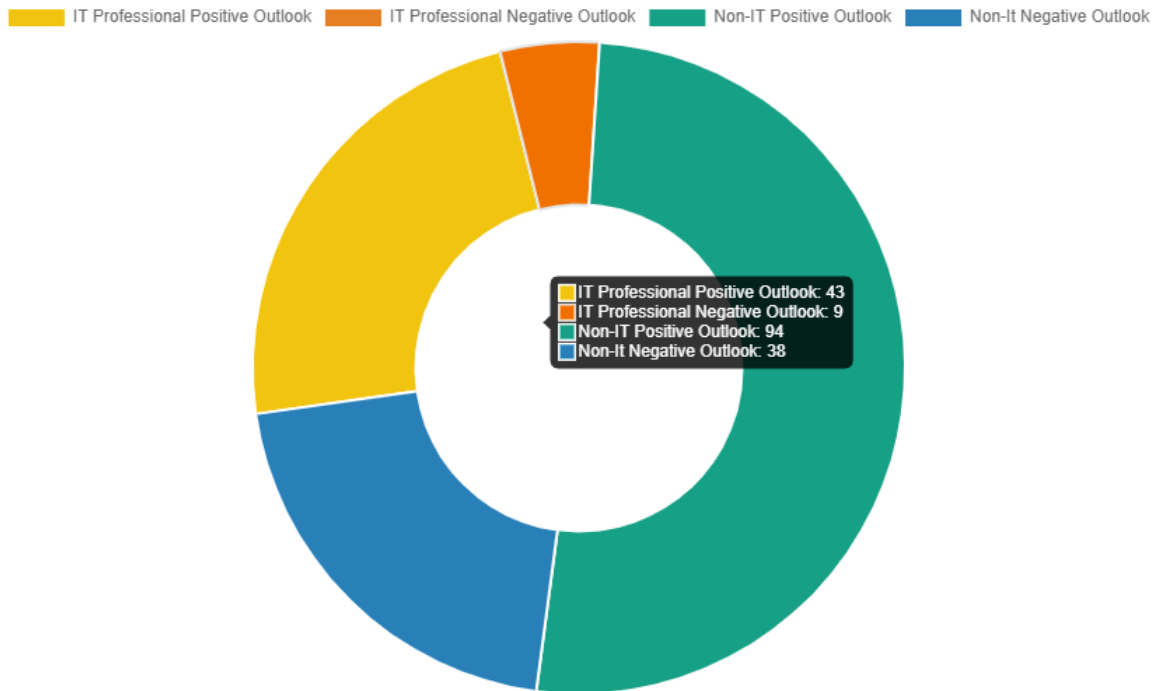


Figure 13. Doughnut chart showing all participants' result data

As illustrated in Figure 13, there is a total of 184 people who participated in the study. Out of the 184 people, 137 people who gave out their opinion said that they are positively looking forward for their future with thinking computers. Using the formula  $\frac{f}{n} * 100$  where  $f$  is the total number of respondents who answered that specific item,  $n$  is the total number of respondents in that group and 100 is a constant, then the values can be substituted like so –  $\frac{137}{184} * 100$ . The result is 74 % rounded to whole number. The people with negative outlook about their future with machine learning / artificial intelligence makes 26% of the total respondents.

## 6.2 Analysis of Participants within IT Industry

The analysis of the opinions of the people with IT background of their future with Machine Learning / Artificial Intelligence is shown in the doughnut chart in Figure 14..

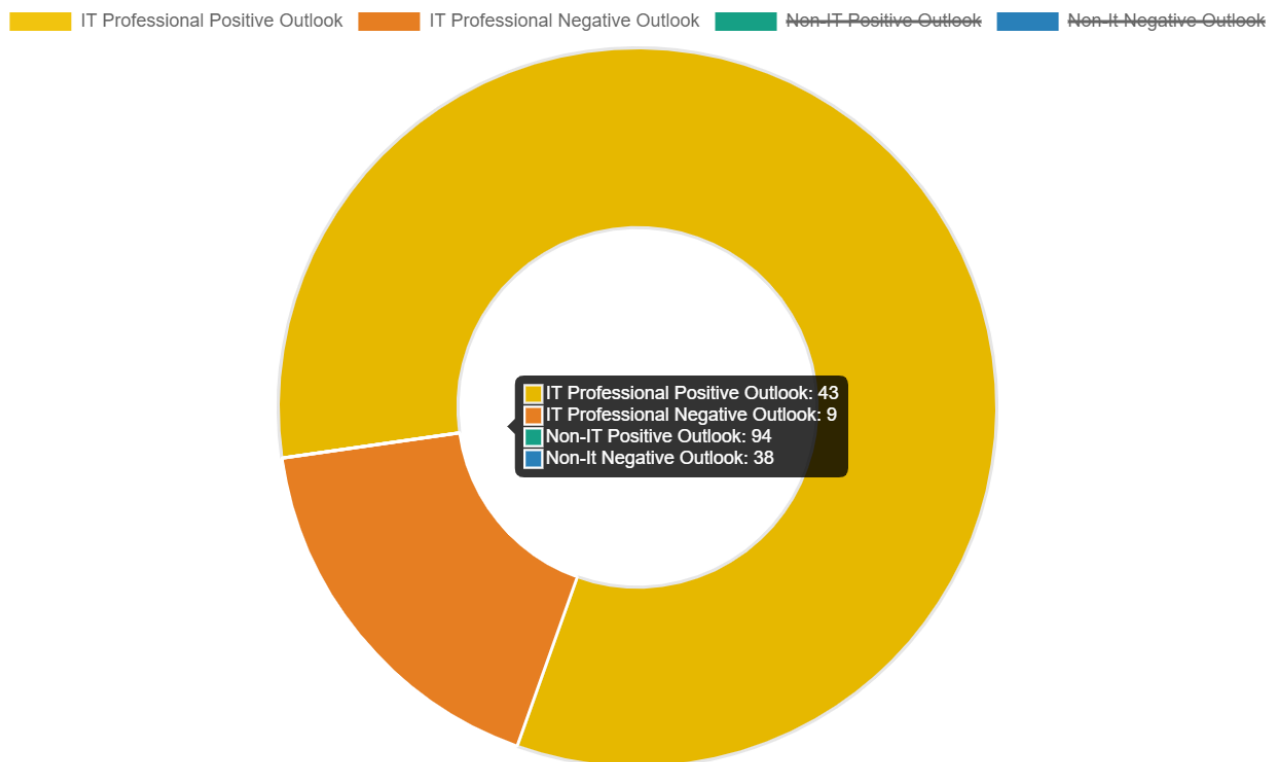


Figure 14. Doughnut chart of participants with IT background

As illustrated in Figure 14, out of the 52 people who participated, 43 respondents are looking forward for their future with thinking computers. Using the formula  $\frac{f}{n} * 100$  where  $f$  is the total number of respondents who answered that specific item,  $n$  is the total number of respondents in that group and 100 is a constant, then the values that were obtained can be substituted like so—  $\frac{43}{52} * 100$ . People who are positive about their future with machine learning / artificial intelligence comprises 83% while those who are sceptical makes up 17%.

### 6.3 Analysis of Participants outside IT Industry

The analysis of the opinions of the people without IT background of their future with Machine Learning / Artificial Intelligence is illustrated in the doughnut chart in Figure 15.

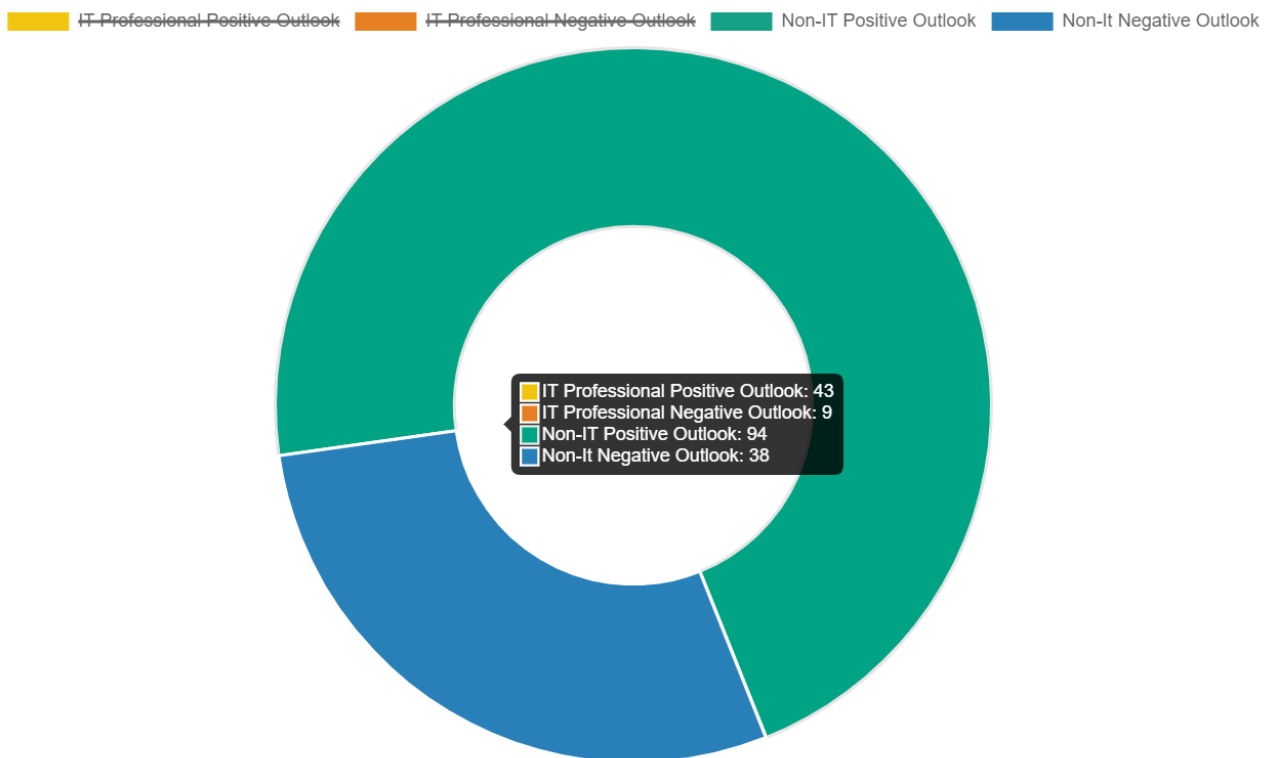


Figure 15. Doughnut chart of participant's without IT background

As demonstrated in Figure 15, there was a total of 132 people who contributed in the research that has no background in IT. Out of the 132 people, 94 people who shared their view said that they are positively looking forward for their future with machine learning / artificial intelligence. Using the formula  $\frac{f}{n} * 100$  where  $f$  is the total number of respondents who answered that specific item,  $n$  is the total number of respondents in that group and 100 is a constant, then the values can be substituted like so  $-\frac{94}{132} * 100$ . Based on the formula, 71% of the participants have positive outlook while 29% said they are distrustful about their future with machine learning / artificial intelligence.

## 6.4 Implementation Experience

The researcher's experience with the project was met with some challenges listed below.

- Time management – This has been a challenge since the researcher is working full-time and had to make certain adjustment with the schedule. Luckily, the company where the researcher is working is considerate enough with arranging the schedule based on the researcher's needs.
- Web application development – The developer who has experience in building web applications encountered some problems with all the stages of the development. The frontend framework chosen in this project was rather complicated as the developer had limited knowledge of reactJS. In addition to that, this is the first time that the developer used chartJS for visualization. This problem was solved by reading and watching a lot of documentations and tutorials and asking questions to developer forums like Stackoverflow. Lastly, the most challenging in the web application development is the deployment. The developer spent couple of days just to deploy the project in the World Wide Web.
- Data gathering – The researcher encountered some problems when it comes to gathering data, specifically finding the right people that can give their opinion. The goal of the researcher was to obtain at least 300 respondents, however due to some problem with the database being corrupted, the total was reduced greatly to just 186. In the beginning, the researcher wanted to have quota sampling which is equal respondents from people who have IT background and those who does not have but in the end, the researcher realizes that this study is not about establishing a correlation between the two groups rather just a general study on people's perspective about their future with machine learning / artificial intelligence.

Overall, the experience of the researcher was generally positive. The challenges stated above were handled accordingly.



## 7 Conclusion

The primary objective of this thesis was to get to know how optimistic or pessimistic people are when it comes to their future with machine learning / artificial intelligence. To accomplish this objective, there should be sufficient data to establish a credible conclusion hence the researcher decided to build a web application to reach as much audience as possible. This proved to be successful as the number of respondents reached 184 which can be considered sufficient to make a reliable assumption.

The following are the conclusions drawn by the researcher based on the results of the study:

- Majority of the participants are optimistic when it comes to thinking computers. This can be concluded since almost three out of four respondents said they are excited about their future with machine learning / artificial intelligence.
- People with IT background tend to be more positive about their future with machine learning / artificial intelligence as 83 % of the respondents said they are excited about the technology that machine learning / artificial intelligence will offer them in the future.
- Having little to no background in IT does not mean people are pessimistic about their future with thinking computers as seven out of ten people who participated in this study viewed it positively.

Overall, people are generally excited about their future with machine learning / artificial intelligence regardless of their profession.

## References

1. Alex Samola and SVN Vishwanathan. Introduction to Machine Learning; 2010.
2. Shai Shalev-Shwartz and Shai Ben-David. Understanding Machine Learning: From Theory to Algorithms; 2014
3. Nataraj Dasgupta. Practical Big Data Analytics 2018
4. Bruce Buchanan. A Very Brief History of Artificial Intelligence 2006
5. David Silver, Julian Scrittwieser, Karen Simonyan. Mastering the Game of Go without Human Knowledge.
6. Viktor Sojanov. Should we Fear AI 2018
7. Sebastian Schuchmann. History of the First AI Winter 2019
8. Netscape and Sun Announce JavaScript URL:  
<https://web.archive.org/web/20070916144913/http://wp.netscape.com/newsref/pr/newsrelease67.html> [Accessed 22 June 2019]
9. Ben Aston. A Brief History of JavaScript 2015 URL:  
<https://medium.com/@benastontweet/lesson-1a-the-history-of-javascript-8c1ce3bffb17> [Accessed 22 June 2019]
10. Jeff Delaney. The weird history of JavaScript 2019 URL:  
<https://dev.to/codediodeio/the-weird-history-of-javascript-2bnb> [Accessed 22 June 2019]
11. Madasamy M. Javascrief brief history and features. URL:  
<https://medium.com/@madasamy/javascript-brief-history-and-ecmascript-es6-es7-es8-features-673973394df4> [Accessed 25 June 2019]
12. Josh Gordon and Sara Robinsin. Introducing TensorFlow.js. URL:  
<https://medium.com/tensorflow/introducing-tensorflow-js-machine-learning-in-javascript-bf3eab376db> [Accessed 25 June 2019]
13. Alex Banks and Eve Porcello. Learning React Functional Web Development with React and Redux
14. Mike Cantelon, Marc Harter, T.J. Holowaychuk, Nathan Rajlich. Node.js in Action
15. PostgreSQL tutorial. What is PostgreSQL. URL:  
<http://www.postgresqltutorial.com/what-is-postgresql/> [Accessed 26 June 2019]
16. <https://www.heroku.com/platform> [Accessed 22 September 2019]
17. [https://github.com/bradtraversy/react\\_express\\_starter](https://github.com/bradtraversy/react_express_starter) [Accessed 22 September 2019]

18. <https://ml5js.org/getting-started/hello-ml5/> [Accessed 22 September 2019]
19. <https://medium.com/@LindaVivah/the-beginners-guide-understanding-node-js-express-js-fundamentals-e15493462be1> [Accessed 22 September 2019]
20. <https://www.atlassian.com/git/tutorials/what-is-version-control> [Accessed 22 September 2019]

## Appendix 1: Server-side script – server.js

```
1  const express = require('express');
2  const app = express();
3  const port = 5000;
4  const bodyParser = require("body-parser");
5  const cookieSession = require('cookie-session');
6  const db = require('./db')
7
8  app.use(cookieSession({
9    secret: `secret is secret`,
10   maxAge: 9999999999
11 }));
12
13 app.use(bodyParser.json());
14 app.use(express.static('public'));
15
16 app.get('/', (req, res) => {
17   console.log(`HELLO`)
18   res.send(`HELLO WORLD`)
19 })
20
21 app.post('/addParticipant', (req, res) => {
22   db.addParticipant(
23     req.body.is_admin,
24     req.body.ip, req.body.name,
25     req.body.selectedOption.value,
26     req.body.is_it, req.body.is_positive,
27     req.body.password
28   )
29   .then(function(result) {
30     console.log(`result is ${result}`)
31   })
32   .then(function () {
33     res.json({
34       success: true
35     });
36   }).catch(function (err) {
37     console.log(err);
38   });
39 })
40
41 app.post('/getParticipant', (req, res) => {
42   console.log(req.body)
43   db.getParticipant(req.body.ip).then((resp) => {
44     if (resp.rowCount != 0) {
45       res.json({
46         success: true,
47         name: resp.rows[0].name
48       })
49     } else {
50       res.json({
51         success: false
52       })
53     }
54   })
55   .catch((e) => {
56     console.log(`ERROR AT POST GET PARTICIPANT ${e}`)
57   })
58 })
59
60
61 app.get('/result', (req, res) => {
62   console.log(`getting`)
63   db.getResult().then(async function (data){
64     res.json(data.rows[0])
65   })
66 })
67
68 app.listen(port, () => {
69   console.log(`listening at port ${port}`);
70 });
71
```

## Appendix 2: Client-side script – Home.js

```
1  import React, {
2    Component
3  } from 'react';
4  import Navigation from './Navigation'
5
6  import {countries} from './countries-list.js';
7  import Select from 'react-select';
8  import axios from 'axios';
9
10 class Home extends Component {
11   constructor() {
12     super();
13     this.state = {
14       is_admin: false,
15       ip: "",
16       name: "",
17       selectedOption: null,
18       is_it: "false",
19       is_positive: "",
20       is_new: true,
21     }
22     this.setNewState = this.setNewState.bind(this);
23     this.handleChange = this.handleChange.bind(this);
24     this.handleClick = this.handleClick.bind(this);
25     this.handleOption = this.handleOption.bind(this);
26     this.handleGet = this.handleGet.bind(this);
27     this.handleOptionChange = this.handleOptionChange.bind(this);
28     this.handleAnswer = this.handleAnswer.bind(this);
29   }
30
31   componentDidMount() {
32     axios.get('https://api.ipify.org').then(resp => {
33       if (resp.data) {
34         this.setState({ip: resp.data})
35       }
36     }).then(() => {
37       this.handleGet()
38     })
39   }
40 }
```

```
41   handleGet() {
42     axios.post('/getParticipant', {ip:this.state.ip}).then((data) => {
43       if (data.data.success === false) {
44         console.log(`FALSE`)
45       } else {
46         window.location.replace('/result');
47       }
48     })
49     .catch((e) => console.log(e))
50   }
51
52   handleChange(e) {
53     this.setState({
54       [e.target.name]: e.target.value
55     });
56   }
57
58   handleClick(e) {
59     e.preventDefault()
60     const { is_admin, ip, name, selectedOption, is_it, is_positive} = this.state;
61     axios.post('/addParticipant', {is_admin, ip, name, selectedOption, is_it, is_positive})
62     .then(resp => {
63       console.log(resp.data)
64     })
65   }
66
67   handleOptionChange(e) {
68     this.setState({is_it: e.target.value})
69   }
70
71   handleAnswer(e) {
72     this.setState({is_positive: e.target.value})
73   }
74
75   setNewState(if_admin) {
76     this.setState({
77       is_admin: if_admin
78     })
79     console.log(this.state.is_admin)
80   }
81
82   handleOption(selectedOption) {
83     this.setState({
84       selectedOption
85     });
86     console.log(`Option selected:`, selectedOption);
87   }
88 }
```

```
89   render() {
90     return (
91       <div>
92         <Navigation />
93       <div className="container">
94         <div className = "formInput">
95
96           <form onSubmit = {
97             this.onSubmit
98           } >
99           <label >
100            NAME: < br />
101            <input className="css-2b097c-container" onChange = {
102              this.handleChange
103            }
104            name = "name" />
105          </label><br/ >
106          <label >
107            COUNTRY:
108            <Select value = {
109              this.selectedOption
110            }
111            onChange = {
112              this.handleOption
113            }
114            options = {
115              countries
116            }
117          />
118          </label >
119
120          <div className="question"> Do you consider yourself as an IT professional?</div>
121          <label>
122            <input type="radio" value="true"
123              checked={this.state.is_it === "true"}
124              onChange={this.handleOptionChange} />
125            Yes
126          </label>
127          <label>
128            <input type="radio" value="false"
129              checked={this.state.is_it === "false"}
130              onChange={this.handleOptionChange} />
131            No
132          </label>
133          <br/>
```

```
134 <div className="question"> Are you excited about your future with  
135 <br/> Machine Learning / Artificial Intelligence?</div>  
136 <label>  
137 <input type="radio" value="true"  
138 | | | | | | | | | | checked={this.state.is_positive === "true"}  
139 | | | | | | | | | | onChange={this.handleAnswer} />  
140 | Yes I am excited!  
141 </label>  
142 <label>  
143 <input type="radio" value="false"  
144 | | | | | | | | | | checked={this.state.is_positive === "false"}  
145 | | | | | | | | | | onChange={this.handleAnswer} />  
146 | No I am worried  
147 </label>  
148 <br/>  
149 <br/>  
150 <button id="submitBtn" onClick={this.handleClick}> Submit </ button>  
151 </form>  
152 </div>  
153 </div>  
154 </div>  
155 );  
156 }  
157 }  
158  
159 export default Home;
```