THE ANALYSIS OF CANADA'S HEALTH THROUGH SOCIAL MEDIA USING MACHINE LEARNING

A Dissertation
Presented to
The Academic Faculty

By

Neel J. Shah
Lakehead University

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science
in the Department of Computer Science

Lakehead University

April 2019

THE ANALYSIS OF CANADA'S HEALTH THROUGH SOCIAL MEDIA USING MACHINE LEARNING

By Neel Shah
Lakehead University

Supervisory Committee:

Approved by:

Dr. Vijay Mago, Supervisor
Computer Science Department
Lakehead University

Dr. Yimin Yang
Computer Science Department
Lakehead University

Dr. Gautam Srivastava
Department of Mathematics & Computer Science
Brandon University

Date Approved: April 29, 2019

The quieter you become, the more you can hear.

- Ram Dass

To my pmarents for their unconditional support.

ACKNOWLEDGEMENTS

Foremost, I offer my sincere gratitude to my advisor Dr. Vijay Mago who gave me this excellent opportunity and supported me throughout my thesis with his patience and knowledge whilst allowing me the room to work in my own way. My sincere thanks to Dr. Salimur Choudhary for the all the encouragement and suggestions during the study. I would like to thank you Mr. Darryl Willick for his constant support for developing system on High Performance Computing infrastructure. I would also like to thank Mr. Andrew Heppner and Mr. Malaravan Balachandran for the crucial edits. I would like to acknowledge the financial support from NSERC Discovery Grant. Last but not the least, I would like to thank my family and every other person who has helped me directly or indirectly throughout the journey.

ABSTRACT

Real-time online data processing is quickly becoming an essential tool in the analysis of social media for political trends, advertising, public health awareness programs and policy making. Traditionally, processes associated with offline analysis are productive and efficient only when the data collection is a one-time process. Currently, cutting edge research requires real-time data analysis that comes with a set of challenges, particularly the efficiency of continuous data fetching within the context of present NoSQL and relational databases. In this thesis, I demonstrate a solution to effectively address the challenges of real-time analysis using a configurable Elasticsearch search engine. We are using a distributed database architecture, pre-build indexing and standardizing the Elasticsearch framework for large scale text mining. The results from the Elasticsearch engine is visualized in almost real-time.

We focused on taking our solution to the challenges of real-time data processing is to apply it on social media to conduct a large scale health analaysis in Canada. Social media a crucial database that provides information on a variety of topics such as health, food, feedback on products, and many others. At present, people utilize social media to share their daily lifestyles, for example, where they are going, what exercise are they doing, or what are they eating. By analyzing the information, collected from these individuals, the health of the population can be gauged. This analysis can become an integral part of the government's efforts to study the health of people on a large scale. This is because public health is becoming the primary concern for many governments around the world, and they believe it is necessary to analyze the present scenario within the population before creating any new policies. Traditionally, governments use a door to door survey, for example, a census, or hospital information to decide their health policies. This

information is limited and sometimes takes a long time to collect and analyze sufficiently enough to aid in decision making. Our approach is to try to solve such problems through the advancement of natural language processing algorithms and large scale data analysis. Results show, the proposed method provides the solution in less time with the same accuracy when compared to the traditional one.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Overview

When conducting health analysis there are many factors which affect the quality of life, which are complicated to measure. Presently, the techniques used to measure the quality of life are traditional surveys [1]. However, the problem with this techniques is the type of data, collection of data, cost, the degree of randomness and time involved with the surveys. Due to this method, the chances of errors are increased and causes an inaccurate view of the state of health in Canada. This therefore has an impact on health policies and programs that lead to major health issues not being addressed with efficient and cost-effective solutions. Due to this new method of analysis are needed and new tools that conduct this type of analysis need to be created. One unique solution is to use social media to conduct data analysis. This is effective because the yearly growth of social media users is 13% on average in Canada. Canada is a good representation with regards to the rest of the world [2] because Canada had a population of 36.79 million in 2018, and among them, 33.05 million were internet users. From 2017-2018 alone Canadian's social media penetration has reached 68% of the total population with 25.56 million people. An average Canadian spends approximate 6 hours of there time per day on the internet. This naturally generates an enormous amount of data and information which can be cultivated to form trends using an analytical engine like Elasticsearch. Out of all the social media platforms, Twitter has the most significant amount of activity with 7.2 million monthly active users all over Canada [2], and the raw data collected includes all types of information from an account of

daily activities to political views. Twitter also provides a qualitative source of information that is a good measure of all social media platforms, that is why we are considering Twitter for analysis of public health [3].

Pertaining to the sheer amount of data generated by social media like Twitter, their is a significant challenge when converting this data into tangible results [4] [5]. Preprocessing in real-time makes this much more difficult, especially when the data is textual and unstructured [6] or crowd sourced [7]. Solutions in the fields of cloud computing and storage are growing at rapid speed, but when [8] cloud-based analytics is limited by network inefficiencies, and recurring costs for the computational resources that are required to perform analysis in real-time picking, the right text analysis tool to sort through 100M tweets are crucial [9]. For our research we used Elasticsearch which is a distributed search and analytical engine. This engine allows for real-time data transformations, search queries, document stream processing and indexing at a relatively high speed. Additionally, Elasticsearch can index numbers, geographical coordinates, dates and almost any datatype while supporting multiple languages (i.e., Python, Java, Ruby). The speed of the Elasticsearch engine is based on its ability to perform aggregation, searching and processing the index of the data [10]. This tool helps us to preform an accurate and concrete analysis on the 100M tweets. We also used Hadoop which is a distributed batch computing platform, using the MapReduce algorithm, that includes data extraction and transformation capabilities. While the platform is based on NoSQL technology that makes uploading unstructured data easy, its query processing HBASE does not have advanced analytical search capabilities like Elasticsearch. Elasticsearch is a text search and analytics tool with a visualization plugin for real-time analysis with an open source license. Furthermore, Elasticsearch hosts plugins for Hadoop and Spark to reduce the distance between the two different technologies and allows for a hybrid system to be implemented

[11]. By using Elasticsearch and Hadoop we were able to analyze 100M tweets and process them with a great amount of efficiency and accuracy creating results that were on par with the Canadian governments' health analysis research.

1.2 Motivation

Public health is becoming the primary concern for many governments around the world, and they believe it is necessary to analyze the current scenario within the population before creating any new policies. Traditionally, governments use a door to door survey, for example, a census, or hospital information to decide their health policies. This information is limited and sometimes takes a long time to collect and analyze sufficiently enough to aid in decision making. Our approach is to try to solve such problems through the advancement of natural language processing algorithms and large scale data analysis. To do all complex analysis we also need basic overview of the data in real-time for large scale scoial-media data.

1.3 Contribution

This thesis introduces a solution to host the large social-media dataset on Elasticsearch framework and then perform health analysis for Canadian's perspective. Part one is about efficiently implementing real-time analytical tool for social-media data using Elasticsearch. It also gives a light on the current limitation of most common database used for large-scale real-time analysis and how Elasticsearch can become a solution for this problem. Locally implemented and configured distributed Elasticsearch for Twitter data analysis in LUHPCC (Lakehead University High Performance Computing Center) helped us to do basic analysis of Twitter data in real-time. It also helped to get and store large scale of social-media data efficiently for more complex research in our case Health analysis of Canadian Population. In second part of thesis we demonstrate the limitation of current social-media health

analysis system in detail and solution to overcome from that problem using NLP (Natural Language Processing).

CHAPTER 2
A FRAMEWORK FOR SOCIAL MEDIA DATA ANALYTICS USING ELASTICSEARCH AND KIBANA

2.1 Introduction

The exponential growth of online data poses a significant challenge in the process of fetching a representative data set that can be translated into tangible results [4, 5]. Pre-processing in real-time adds another layer of complexity, especially when the data is textual and unstructured [6] or crowd sourced [7]. Solutions to processing big data sets in the fields of cloud computing and storage are growing at rapid speed, but when we consider big data on a scale of petabytes [8], cloud based analytics are limited by network inefficiencies for transporting the data; and recurring costs for the computational resources required to perform analysis in real-time [9]. Access and privacy also pose a challenge in cloud based storage as server administrators maintain the rights to view both the data and its flow. Security solutions such as encrypted searching are not feasible to implement specific to real-time analysis because of computational limitations [12]. Currently, the top three tools used for analyzing large databases are Elasticsearch, Hadoop and Spark [13]. Elasticsearch is a distributed search and analytical engine which allows for real-time data transformations, search queries, document stream processing and indexing at a relatively high speed. Additionally, Elasticsearch can index numbers, geographical coordinates, dates and almost any datatype while supporting multiple languages (i.e., Python, Java, Ruby). The speed of the Elasticsearch engine is founded on its ability to perform aggregation, searching and processing the index of the data [10]. Hadoop is a distributed batch computing platform, using

the MapReduce algorithm, that includes data extraction and transformation capabilities. While the platform is based on NoSQL technology that makes uploading unstructured data easy, its query processing HBASE does not have advanced analytical search capabilities like Elasticsearch. Elasticsearch is a text search and analytics tool with a visualization plugin for real-time analysis with an open source license. Finally, Elasticsearch hosts plugins for Hadoop and Spark to reduce the distance between the two different technologies and allows for a hybrid system to be implemented [11].

Tools that support the management of large data sets and real-time data fetching include *relational* (MySQL, Oracle Database, SQLite), *Graph*(Neo4j, Oracle Spatial) and *NoSQL* (MongoDB, IBM Domino, Apache CouchDB). Limiting factors related to all types of databases include lack of support for full-text searches in real-time. While NoSQL is functional for full text searching it lacks reliability when compared to relational database models [6]. Traditional databases require that the data is first uploaded and then the administrator must actively decide which data should be indexed which adds one more layer of processing making it infeasible for real-time analysis. Elasticsearch provides a solution to these limiting factors [6] by providing a highly efficient data fetching and real-time analysis system that:

- performs pre-indexing before storing the data to avoid the need to fetch and query specific data in real-time;
- requires limited resources and computing power in relation to traditional solutions; and
- provides a system that is distributed and easy to scale.

The capacity for Elasticsearch to contribute to high efficiency, real-time data analysis is enhanced through a standardized configuration process, *shard size* management and standardizing the data before upload into Elasticsearch and demon-

strated through a discussion of both the working architecture as well as a real-time visualization of social media data collected during December 2017 and May 2018, a repository of over 1 billion twitter data points.

This chapter gives an introduction about the how social-media is utilized on large in real-time. We then discuss related works about the similar current research being done in our field of social-media data-analysis in real-time. We use these related works to discuss limitations in most common used databases and how we can overcome them. We then go into our methodology which includes: basic overview of Elasticsearch, implemented distributed Elasticseach in LUHPCC, optimize configuration ,and updated data format for storing data. We analyze our final results and form a detailed health analysis for Canada. We finally conclude by summarising our results and discuss future possibilities for Elasticsearch system.

2.2 Related Work

Marcos [9] suggests that cloud computing is elastic in nature as the user can adjust it as per his/her data needs from processing power to storage. While it does seem ideal in theory, cloud computing comes with several challenges including both network inefficiency in data transport as well as issues related to data privacy and access control. Additionally, Hashem refers to 'data stabbing', which are problems associated with storing and analyzing the heterogenous and complex structure of big datasets [14]. As a solution, other authors such as Oleksii [6] support and highlight the benefits of Elasticsearch as a tool for real-time analysis in modern data mining repositories. In this research we attempted to address and resolve problems associated with data preprocessing and efficiency while also discussing the elastic cluster framework in more depth. Currently there are very few research studies on frameworks for big data analysis in real-time although several discuss the application of practices in manufacturing [15] and gene coding [16].

Some researchers have used Elasticsearch cluster via a logstash plugin and MySQL databases for heterogenous accounting information system [17]. In these research data is monitored using MySQL server before inserting it into Elasticsearch. The researchers observed that there might be an issue of duplication of data and storage space, but the architecture ensures flexibility and modularity for the monitoring the system. They choose Elasticsearch as text search engine in real-time which allows them to search historical data [17]. Mayo Clinic healthcare system [18] developed a big data hybrid system using Hadoop and Elasticsearch technology. In healthcare, real-time result is essential for effective decision making. Before that, they used traditional RDBMS database to store and process data. But, it lacks integration between different platforms and inability to query/ingest of healthcare data in a real-time or near real-time. In Mayo Clinic system [18] Hadoop is used as a distributed file system and on top of it Elasticsearch works as a real-time text search engine. When there is a need for raw data Hadoop is used, and for real-time analysis Elasticsearch is used. Their experimentation showed very promising results, like searching 25.2 million HL7 records took just 0.21 second [18].

Designsafe web portal by Natural Hazards Engineering Research(NHER) [19] analyze and share experimental data in real-time with researchers across the world. The user of their system sends the large amount of data which is stored in distributed NFS. During the preprocessing of the data, which includes analysis of string and basic cleaning, indexing the data and make it compatible for Elasticsearch. This model allows users in a different location to query the same experimental data which is computed in different part of the world in real-time. All present environments needs to be correctly configured as per the data and the requirements [19].

2.3 Limitation

As Elasticsearch is designed to be used for real-time analysis, there are databases which provide functions that perform better in offline mass data analysis such as NoSQL databases (e.g., MongoDB) that support MapReduce [6]. Elasticsearch does not support MapReduce as it instead relies on the inverted index [20]. Additionally, Elasticsearch can be slow when new data is added to the index and it currently lacks support for more popular data formats (e.g., XML, CSV) and only supports JSON format which can be challenging for users unfamiliar with JSON [21]. The important challenge is to implement present Elasticsearch database server efficiently on locally to get overall view of large scale Twitter data in real-time. Next section will give details explanation of Elasticsearch and implemented system in LUHPCC.

2.4 Methodology and Implementation of Distributed Elasticsearch

2.4.1 Elasticsearch

Elasticsearch is a database manager that is crucial to my research because it is one of the fastest real-time text search engine which will be able to easily analyze tweets for health analysis. The following is a bit more on the history of Elasticsearch. Elasticsearch was started in 2004 as an open source project called *compass*, which was based on Apache Lucene [22]. Elasticsearch is a distributed and scalable full-text search engine written in Java that is stable and platform independent. These features combined with requirement specific flexibility and easy expansion options are helpful for real-time big data analysis [23]. We use this database manager as a base for our twitter analysis but configured it in order to be optimize is results and improve accuracy. We will discuss some of the general functions of Elasticsearch to provide context for the Elasticsearch configuration and data standardization and

shard management procedure resulting from this research.

2.4.2 System Overview

Elasticsearch is the database manager of choice for this article. The following section explains how it was implemented in our system. Figure 2.1 illustrates the framework for real-time analysis of very large scale data based on Elasticsearch and Kibana [24]. In the first step, the Twitter API is used for scraping twitter data (approximately 1400 tweets per minute) that is stored in a MongoDB database, which is installed on a Network Attached Storage (NAS) with a capacity of 16TB. The twitter data is transferred to preprocssing units which handle the data and transfer it to High Performance Computing (HPC) infrastructure in *almost* real-time. As traditional databases, including MongoDB, are not efficient enough to handle real-time query, we transfer the processing and analysis of data to Elasticsearch, which is implemented via HPC lab resources. Before uploading the data, we standardize the twitter object for Elasticsearch and use multithreading to upload the data for better real-time performance and to shorten the gap between receiving and processing data. When a user needs any data, a query will be sent to Elasticsearch using the Kibana front-end. Elasticsearch processes that query and sends the query result object (JSON format) to Kibana, where Kibana shows the query object to the user.

Within the general functioning of the search engine, Elasticsearch uses a running instance called a node which can take on one or more roles including a master or a data node (see section 2.4.1, Figure 2.2). Data-set clusters within Elasticsearch require at least one master and one data node, however it is possible that a cluster can consist of a single node since a node may take on multiple roles. The only data storage format compatible with Elasticsearch is JSON and therefore requires data mapping for producing functional analysis and visualizations due to the un-

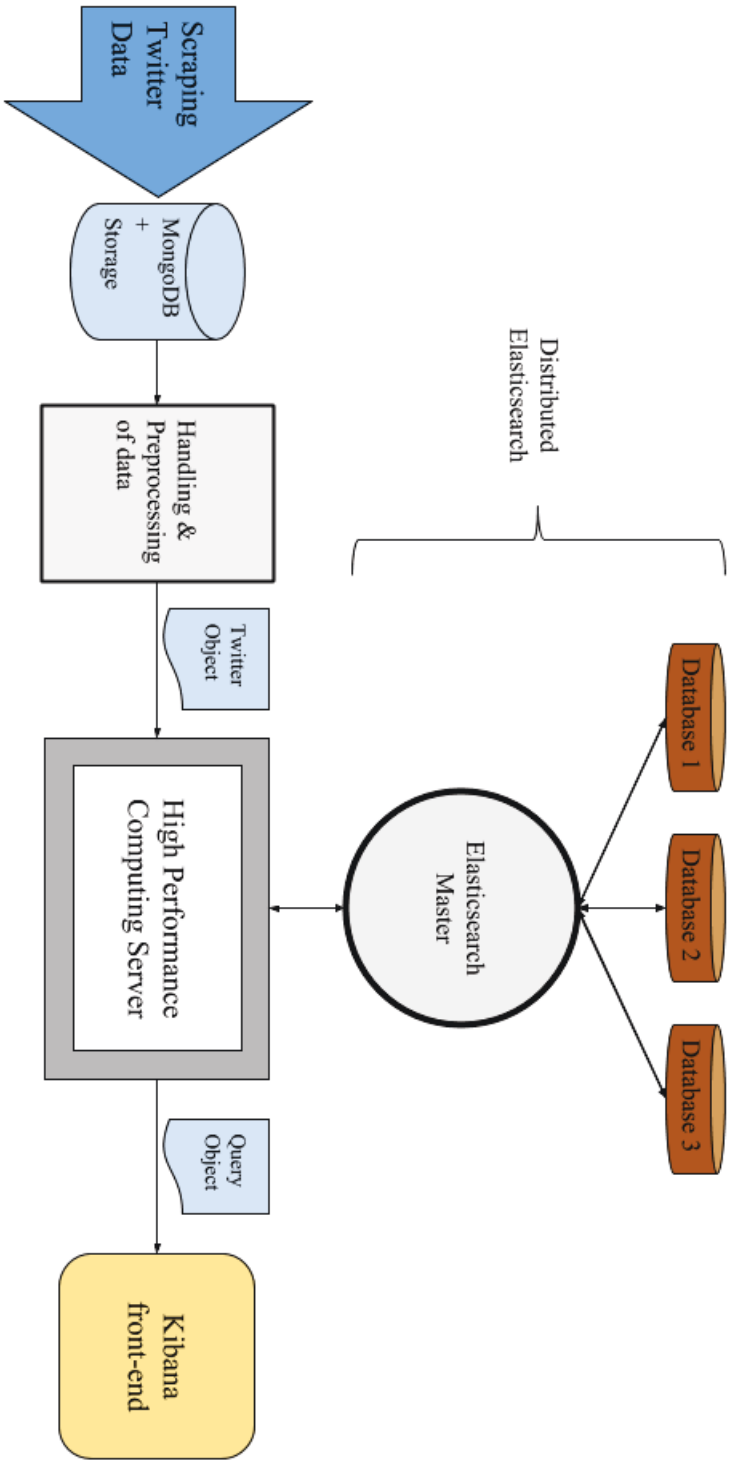

Figure 2.1: Framework for real-time analysis using Elasticsearch

structured format of the twitter data. We observed that reliance on the JSON format makes the system more flexible than MySQL and other RDBMS, but less than MongoDB. While a traditional database such as RDBMS use tables to store the data, MongoDB uses BSON (like JSON) format, and Elasticsearch uses an inverted index via the Apache Lucene architecture to store the data [22]. A typical index in Elasticsearch is a collection of documents with different properties that have been organized through user defined mapping that outlines document types and fields for different data sources; similar to a table in an SQL database. The index is then split into *shards* housed in multiple nodes where a shard is part of an index distributed on different nodes. Within the Elasticsearch framework, the inverted index allows a more categorical storage of big data sets within nodes and *shards* so that real-time search queries are more efficient. Elasticsearch uses RESTfull API to communicate with users, see Table 2.1 for a basic architecture comparison. Additionally, there are different libraries such as Elasticsearch in Python [25] and Java [26] for better integration.

Table 2.1: Comparison between Elasticsearch and RDBMS basic architecture

Elasticsearch	**RDBMS**
Index	Database
Mapping	Table
Document	Tuple

Backbone:

While Elasticsearch is a powerful tool, a model is required to optimize functionality for the purpose of real-time big data analysis specific to social media. The purpose of this research is to provide (i) a specific configuration file to optimize the organization of the data set, (ii) an optimized *shard* size for maximum efficiency in storage and processing, and (iii) a standardized structure for data fields present within Twitter to eliminate over-processing of irrelevant information When the

data is stored in Elasticsearch, it stores the data in an index first, and then the index data is stored as an inverted-index using an automatic tokenizer. When we search in Elasticsearch, we get a 'snapshot' of the data, which means that Elasticsearch does not require the hosting of actual content but instead links to documents stored within a node to provide a result through the inverted index. These results are not real data but a representation of the query's linkages to all associated documents stored in each node. As a component of this project, the following configuration file was developed and can be replicated in Elasticsearch on any HPC by editing the configuration files as per number of nodes and capacity of server. Table 2.2 describes the basic configuration file for Elasticsearch.

Table 2.2: Master and Data node configuration file

Master node config file	Data node config file
cluster.name: dsla node.name: m1 node.master: true node.data: true path.data: /data/nshah5/dataset path.logs: /data/nshah5/log network.host: x.x.x.x network.bind_host: 0 network.publish_host: x.x.x.x discovery.zen.ping.unicast.hosts: ["x.x.x.x"] bootstrap.system_call_filter: false	cluster.name: dslab node.name: d1 node.master: false node.data: true path.data: /data/nshah5/dataset path.logs: /data/nshah5/log network.host: x.x.x.x network.bind_host: 0 network.publish_host: x.x.x.x discovery.zen.ping.unicast.hosts: ["x.x.x.x"] bootstrap.system_call_filter: false

Here, the name of a cluster is *dslab* and a cluster name is necessary, even if only a single node is present. As the Elasticsearch is a scattered database, where one or many nodes work as heads and others as data, this parameter is used to interconnect all the nodes in the cluster. We can create numerous clusters with the same hardware using different instances of Elasticsearch and different configuration files.

Table 2.3 is an example of a configuration file features for any Elasticsearch node. In every node for the distributed Elasticsearch we have to configure the

Table 2.3: Elasticsearch node configuration file features

Config file properties	**Explanation**
cluster.name	It is the name of cluster where present node will join.
node.name	It gives the name of your current node
node.master	The role of master-eligible is decided based on true or false function (Boolean function). The master node manages the overall state of the cluster including node monitoring, index creation and deletion, and *shard* to node assignments.
node.data	The role of data is decided based on true or false function (Boolean function). It stores the physical data *shards*, performs reads, writes, searches and aggregations. Any node can be master and data, both or individual.
path.data	The location of the actual data in present node is represented.
path.logs	Location where the logs of the present nodes are stored. Logs are important to diagnose problems and monitor working status.
network.host	It's an address of the present node which is unique for the individual node in the cluster.
network.publish_host	It's a public address where other nodes communicate with the present node.

same file in each and every instance. When the data is stored we use the index to store a specific type of data similar to a data-set in MySQL. The performance of Elasticsearch is based on the mapping of the index and how we size the *shards* of the data set. The formula to decide the size of the *shards* is given in Equation 2.1.

$$Number\ of\ shards = (Size\ of\ index\ in\ \ GB)/50 \tag{2.1}$$

The reason behind the consideration of using 50 GB as a *shard* size is due to the architecture in Elasticsearch. The architecture supports 32 GB index size and 32 GB cache memory so ideally the *shard's* memory should be less than 64 GB and through experimentation we observed that the best results are achieved at *shard* size of 50 GB.

2.4.3 Updated Data Structure

Table 2.4: Difference between normal and updated structure

Original tweet structure	**Updated structure**

{
“??Tweet”:{
“User”??:{
“Id”??:
“Name”??:
}
},
...
}
	{
	“Id”:
	“Name”:
	...
	}

We used Elasticsearch to analyze 250+ million out of 1 billion tweets scraped between December 2017 and May 2018 using the Twitter API. Since the Twitter API response is in JSON format and contains unstructured and inconsistent data the sequential collection of all data fields within the tweet JSON object is not guaranteed. Standardization of the data and conversion into a structured format is therefore necessary for Elasticsearch mapping so that each field of data is present

when loaded into the index. To optimize the Elasticsearch we changed the storage format of the tweet so that all the data is required to be at **depth level** one in JSON format. Table 2.4 depicts the basic example of restructured data in Elasticsearch.

2.4.4 Configuration of the Elasticsearch

Live social media streaming data is stored in elastic clusters. Each elastic cluster contains 6 nodes, with each node having 2 threads and 12GB of memory. Within these 6 nodes one node works as a master and the remaining 5 work as data nodes. Architecture of the elastic cluster is shown in Figure 2.2.

2.5 Analysis and Results

2.5.1 Elasticsearch results

Table 2.5: Search query result of "pizza" keyword

Result of keyword "pizza" from all tweets from database

{
"took": 4060,
"timed out": false,
"shards": {
"total": 106,
"successful": 106,
"skipped": 0,
"failed": 0
}
 	
"hits": {
"total": 192118,
"max_score": 15.110959,
"hits": [???]
}

As we mentioned previously, the data is stored as an inverted index that is optimized for text searches and therefore very efficient. For example, if we search for the keyword "pizza" within the context of all tweets (250+ millions) in Elastic-

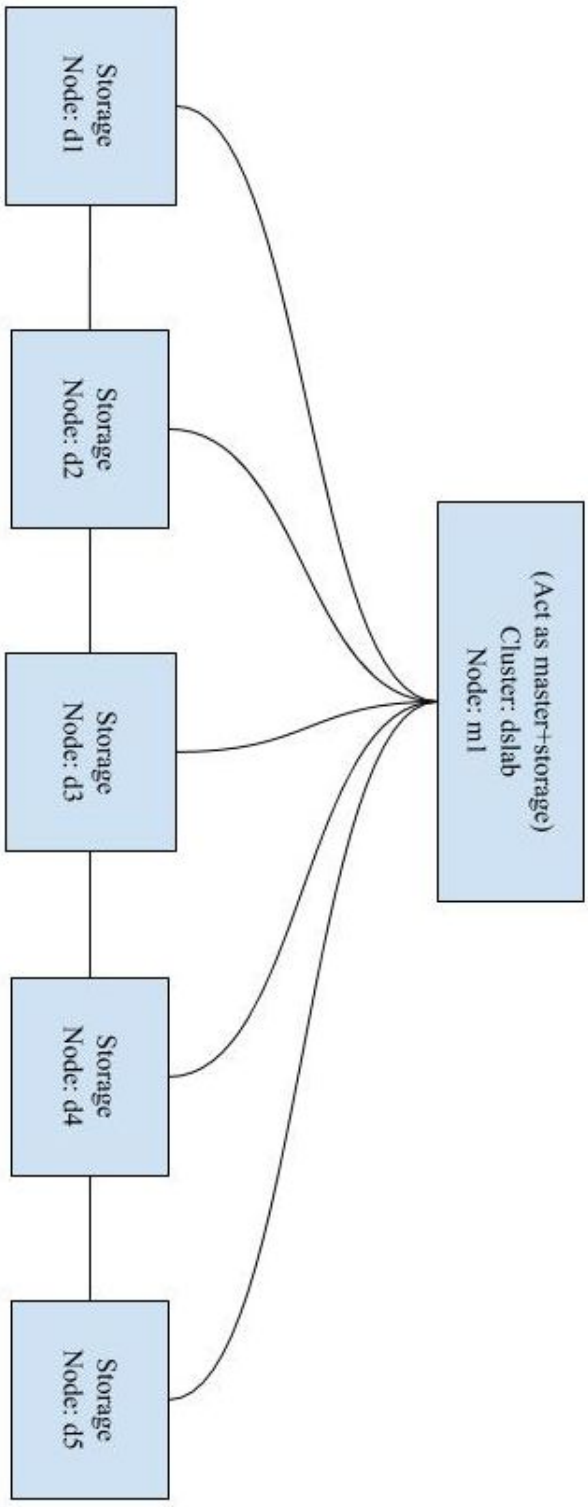

Figure 2.2: Elasticsearch cluster architecture hosted on the HPC at Lakehead University

search, the time taken is 4060 milliseconds (4.06 seconds) to find a total of 192,118 tweets where the "pizza" keyword is present in tweet text. Table 2.5 shows the example of the keyword "pizza" text search query response from Elasticsearch. Figure 2.3 shows a pie chart of tweets mapping the geographical distribution by nation of "pizza" tweets where the United States alone is responsible for 47% of total tweets and other countries excluding the top five are 30%, which is 77% of total tweets. Additionally, the visualization shows the time taken to perform the query is 13ms (0.013 second). Figure 2.4 shows five most used languages in the tweet text related to "pizza" where the English language is used in more than 77% tweets while Spanish is used 12%, Portuguese at third spot with 6%, French at 3% and Japanese at 2% tweets. In this instance Elasticsearch took 17ms for query processing.

Figure 2.5 shows the devices used to tweet with 38% of tweets coming from the iPhone twitter app, the Android twitter app was used for 29%, twitter web clients were used for only 11% and Twitter lite and Tweetdeck combined were used for around 7%. Other sources were indicated for the remaining 15% tweets. This query took 11ms to execute, which is quite reasonable given the structure and amount of data.

The above results demonstrate the efficiency of this data analysis system in that all three tasks (fetching the data, performing descriptive analysis and creating graphs), were accomplished in less than 15 seconds from a database size of 250+ million tweets. Clearly, this framework has proven suitable for the analysis of large text data in real-time without losing accuracy. It also shows that the restructuring and standardization procedures used on the data assisted in optimizing the accuracy of the results and efficiency of the processes in a context with limited resources.

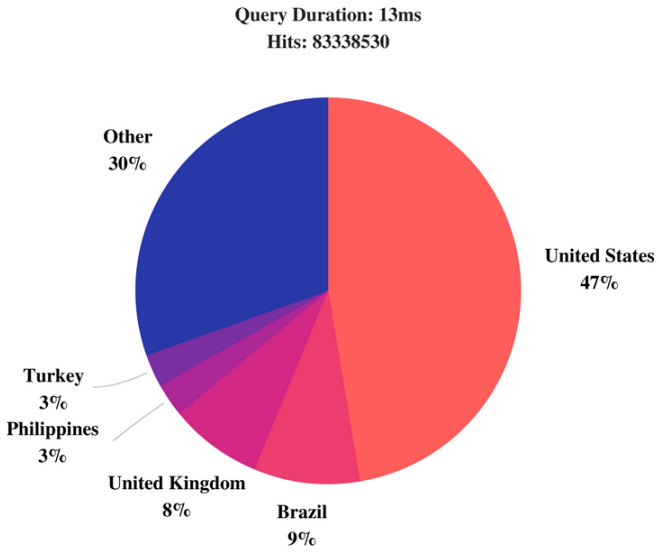

Figure 2.3: Location: Real-time analysis of Twitter data for the term “pizza”

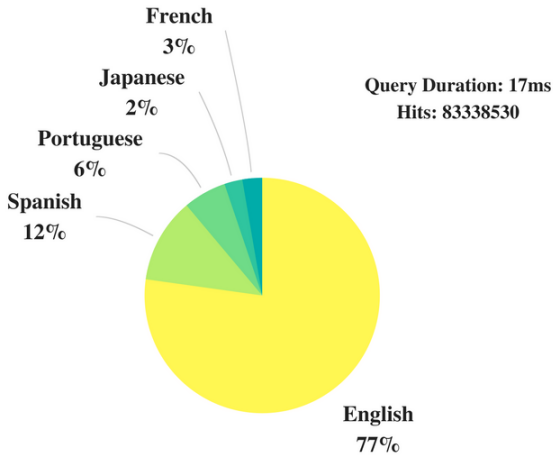

Figure 2.4: Language: Real-time analysis of Twitter data for the term “pizza”

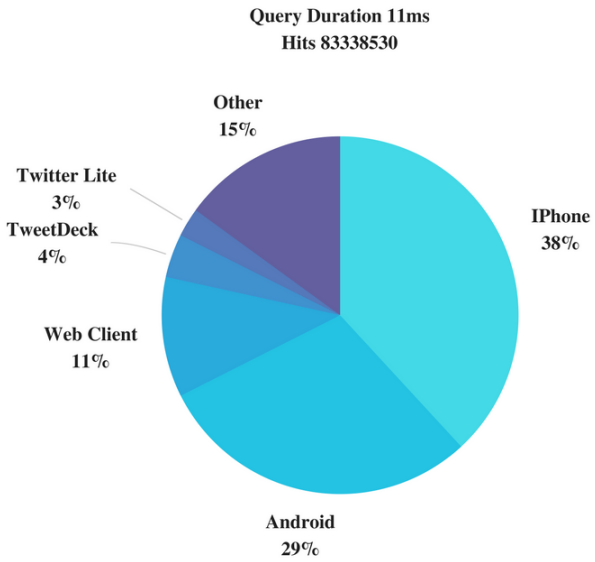

Figure 2.5: Source: Real-time analysis of Twitter data for the term "pizza"

2.5.2 Kibana- Visualization Dashboard

In addition to Elasticsearch being efficient for real-time analysis, extended plugins such as Kibana [24] and logstash [27] make it convenient for functional representations of big data in real-time. It is part of the *elastic stack* and is freely available under open source license. Kibana has multiple standard visualizations available by default and simplifies the process of developing visualizations for end users with a drag and drop feature. As Kibana is backed by the Elasticsearch architecture, it functions quickly and is efficient enough for real-time analysis. Finally it provides the opportunity for graphical interaction in the process of building and handling queries with an accessible visualization of the cluster health and properties within the database.

At present, the monitoring framework described in this paper is used to display data coming from Twitter stream. For example, in Figure 2.6 we show a snapshot of the *Kibana* dashboard. The top-most plot is a pie chart of tweet source, which

Figure 2.6: Partial view of the Kibana dashboard for the twitter Analysis

displays the results from which device they use to tweet, such as iPhone, web browser etc. The second top-most plot is pie chart of the languages used to tweet. In the middle, first histogram shows the the time and amount of twitter data flow. And, the second shows the word cloud and the bottom left shows the top ten users who are actively twitting. Similar dynamic dashboard creation is possible in minutes without knowledge of any programming knowledge and back-end system understanding.

2.6 Key contributions

The key contribution to Elasticsearch system is to implement it on locally on LUH-PCC. We also standardized the data format for Twitter data for more optimize indexing in Elasticsearch. We presented the optimal way to configure the Elasticsearch through shards equation for getting maximum output from present infrastructure. Also, we assembled the Kibana visualization plugin with Elasticsearch for real-time visualization of Twitter data at large scale as shows in results.

CHAPTER 3

CANADIAN'S HEALTH ANALYSIS THROUGH SOCIAL MEDIA

3.1 Introduction

Every year internet access is increasing at 7% rate around the world [2]. The yearly growth of social media users in Canada is almost twice the growth of internet access with 13% on average. Canada is a good representation with regards to the rest of the world as Canada had 36.79 million population in 2018, and among them, 33.05 million were internet users. This is almost 90% of total population. This shows how deeply social media and internet has penetrated the Canadian society. As internet access and quality increases, it creates an ideal condition for the growth of social media and other online activity. From 2017-2018 alone, Canadian's social media penetration has reached 68% of the total population with 25.56 million people and reasons behind the exponential growth of social media users is mainly due to the technological advancement of smartphones and qualitative internet services (with an average speed of internet 45.64 Mbps) [2]. An average Canadian spends approximately 6 hours of their time every day on the internet. 89% of the total population use the internet daily for various activities [2]. Smartphones are essential for social media, as they enable users to share their activities with ease of accessibility when compared to traditional social media devices such as computers. An example of this is a camera integrated into the social media app to upload content without the hassle of the conventional equipment instantly. In Canada, smart-phone users are growing with the rate of 6% every year which will increase the consumption of social media, internet and different online services [2]. This naturally generates an enormous amount of data and information which

can be cultivated to form trends. Twitter has the most significant amount of activity at 7.2 million monthly active users all over Canada [2], and the raw data collected includes all types of information from a review of restaurant or product, political views, user's like-dislike, daily routine, or other data [4]. Since Twitter provides a qualitative source of information that is a good measure of all social media platforms, we are considering Twitter for analysis of public health [3].There are many factors which affect the quality of life, which are complicated to measure. Presently, the techniques used to measure the quality of life are traditional surveys [1]. However, the problem with these techniques is types of data, a collection of data, cost, the degree of randomness and time involved with the survey. Due to this method, the chances of errors are increased. This will affect the decision of health policies and monitoring as it is not a proper representation but a skim of the actual state of health due to the reason above.

By studying the health of the population, trends can be formed with regards to prevalent health conditions. For example diabetes, cancer, heart conditions [28, 29]. Many of these health conditions are correlated with nutrition and level of daily physical activity. The government knows this and conducts surveys, and programs to analyze the current health of the nation [30]. By doing so they can put in appropriate policies and programs in order to help the population stay healthy and active.

This chapter is constructed firstly with an introduction to give a general idea of social media and how it is utilized for large scale data analysis. We then discuss related works to get an idea of the similar current research being done in our field of health analysis. We use these related works to discuss limitations in data analysis and how we can overcome them. We then go into our methodology which includes: data cleaning, creating a database, phrase detection,and model training. We analyze our final results and form a detailed health analysis for Canada. We

finally conclude by summarising our results and discuss future possibilities.

3.2 Related Work

Google Flu Trends was real-time flu detection tool based on Google search query [31]. If people searched for a solution to cure the flu or about any medical information related to the flu, the algorithm takes that information and considers their location as a flu affected area [31]. But that algorithm was proven imperfect and needed a more sophisticated approach to solving it. In Paul and Drendze's health analysis article [32], there is a correlation when comparing cancer tweets, with higher obesity and smoking tweets. They also found a negative relationship between health care coverage and tweets posted about diseases [32]. With more sophisticated algorithms, the accuracy of the data increases and this can be used to discover more true trends when looking at Twitter for health analysis.

In [33], Shawndra found that people who search about the sodium content per recipe is correlated with the number of people admitted in the emergency room of a major urban Washington hospital for congestive heart failure [33]. J Eichstaedt's found the sentiment analysis of tweet language, which outperforms the traditional socioeconomic surveys for predicting heart disease at the country level [34] They correlated the growth of negative emotions in Twitter with the risk factor of heart disease on a large scale [34]. This shows that social media analysis can be more effective than traditional surveys and may be the next step of methodology for future analysis done by the government.

Culotta [35] analyzed tweets which contain the daily habits of the account holders. The results were a "deep representation" of the US community in regards to their daily negative engagement concerning their routine such as watching television, playing or reading [35]. Abbar also did an analysis of data on Twitter for caloric analysis at the country level. They classified food-related tweets and

found the caloric value of such food. This analysis gave a brief understanding of the food habits of the people in different demographic areas [36]. Subsequently, Lexicocalorimeter is one of the most sophisticated approaches towards the health analysis of people country level. This is done by utilizing social media. Lexicocalorimeter is an online instrument that is designed for measuring social, physical and psychological examination at a large scale. Sharon et al. developed it for public health monitoring and to create health policies through data-centric comparison of communities at all scales. Oversimplification exists in data analysis and basically means that the data is being classified in basic categories. by doing this only look at what the data presents instead of looking deeper into the meaning or relevance of said data. This can cause bias as one bit of data and show one thing but mean another thing. And example of this is a piece of data from a Twitter account that says ”the test was a piece of cake”. This is an idiomatic expression that has very little to do with food. Instruments like the Lexciocalorimeter will take this data as a food tweet and at it to is trends. This causes error and inaccurate trends and due to our models need have a resistance to oversimplification in order to get more accurate results.

Lexicocalormeter extracts text related to caloric input and caloric output and calculates their caloric content [37] [38]. They also use food phrases from a 450-plus database and physical activity phrases for a 550-plus database. The second step is to group categorically similar words and phrases into small pieces called lemmas. They then assign caloric values to it, based on the food and physical activity. To get these lemmas, they use a greedy selection algorithm [39].

Food caloric value is represented as C_{in} and activity caloric value is represented as C_{out}. C_{rat} [39] is calculated as shown in Equation 3.1.

$$C_{rat} = \frac{C_{in}}{C_{out}} \tag{3.1}$$

And to find out the average caloric value of the different provinces or countries, they count the frequency of all food and activity related words and then assign caloric values to all words. Then, the standard C_{rat} formula is used to compute the caloric ratio of each place. In this paper, they consider 80.7 kilograms as the average weight for metabolism equivalent of tasks; this is subtracted from the calorie's physical activity value. More details can be found in [39].

3.3 Limitation

For simplicity, the Lexicocalorimeter [39] didn't use any filter for tweets beyond their geographic locations [39]. This causes bias in the dataset because the user may live and eat in different locations. This causes the users eating habits to affect another location's dataset instead of affecting there home locations dataset. For example the user might be from Thunder Bay and go on a trip to eat in Toronto. With Lexicocalorimeter's current filter the user's data will affect the dataset gathered from two separate locations instead just Thunder Bay like it should. This causes a loophole in the dataset that will cause inaccuracy.

Lexicoclaorimeter's dataset is quite limited with 451 food phrases [40]. Since this 451 food phrases only has the most common food names it is not a good database of the food itself. But when people talk about the food, it can be called anything such as the name of special food in certain restaurant [40]. Different cultures have different foods and this is very important in a country as diverse as Canada. So the database of food phrases in our model must be large in order to accommodate for all possibilities in order to be accurate.

Another limitation is that the Twitter account may talk about food or an activity in a metaphorical perspective. Food words are commonly used in idiomatic expressions in the English language. Some examples include: ”Bring home the Bacon”, ”Cry over spilt milk” and ”Cup of tea.” Lexicoloricmeter will still consider

these phrases as food items in their systems and then assign values to them. The instruments approach cannot solve that problem, and this creates bias data in the system. An example of this includes if a person tweets the phrase ”you are the apple of my eye,” the present algorithm will consider apple as a food. But in this case, it is not related to food. Also, a lack of NLP understanding of approach creates higher chances for the bias output [41]. Due to this, unnecessary data will enter the dataset and create false trends, over-fitting and decrease accuracy of the overall analysis.

3.4 Methodology

In the design of our system, I focused on the system for the large scale analysis of social media data with regards to health analysis. The focus of the system is on training NLP model based on a large amount of data that is processed to get factual information about the health of Canadians. Figure 3.1 shows the architecture of a health analysis system. It is divided into two subsections, training part (offline mode) and analysis component (active system). As shown in the Figure 3.1 the first step is to collect raw data. To manage and process this data, I use an Elasticsearch system which is designed and developed through Elasticsearch locally at Lakehead University's High-performance computing facility [42]. It can handle and analyze Terabytes of text data within a few seconds. This helped us to get the necessary data very efficiently from the pool of data. Once this is done, our next step is data cleaning.

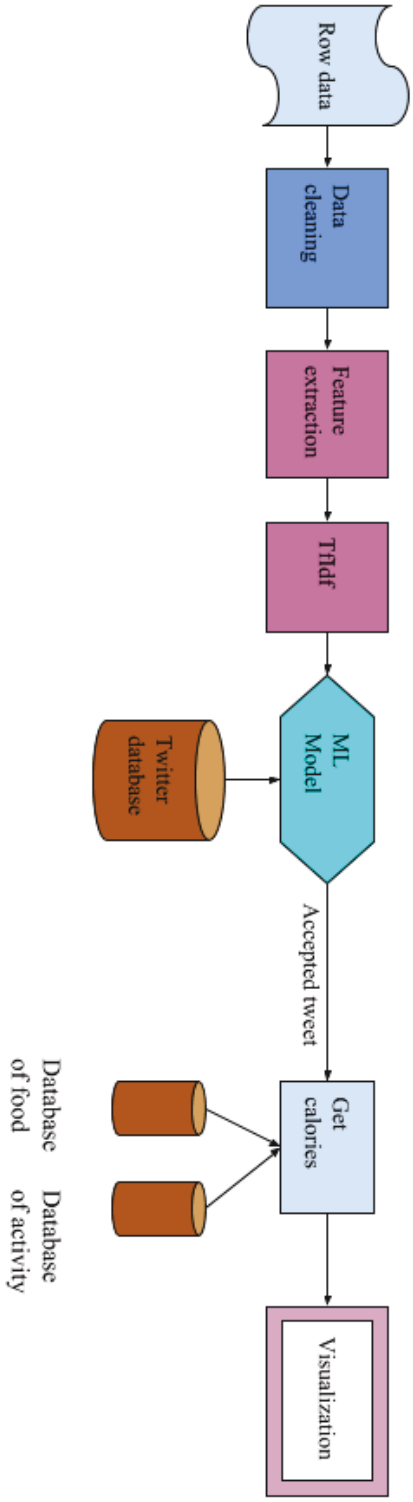

Figure 3.1: Architecture of analysis system

3.4.1 Data Cleaning

Data cleaning is crucial when I deal with a user's raw data such as tweets, feeds or chats. This raw data is not structured or cleaned unlike typical formats such as blogs or essays. When tweets are written, it includes hashtags, slang words, emojis, emoticons and unstructured data. Because of that, such data is used as a feature in the model as an input; to make this data more sensible and more reliable for the model.

In the first step of data cleaning, I convert all Emojis or Emoticons to there respective meaning through the "emot" open source library [43]. It helps to understand the text when we have a name that is not in the database. When we have emojis that are related to food, it will be easy to understand that tweet is related to food. In the next step, I convert all text into the lower case which makes word matching and processing easy in further processes. Step 3 is removing stop words which help to eliminate unnecessary features in our model such as the, a, an, when, what etc. Next step is to remove special characters. In the last step, I remove the numbers. This is because numbers are not useful for identifying whether the text is related to food or not and it also not source information for our analysis. And removing all unnecessary text or data will limit the size of the feature matrix and speed up the training and classification task.

Step 1: Emoji or Emoticons gave there respective meaning

Original text: ″I am getting 2 old to be mango Gonna retire soon and be joesh #ROFL :-)″

Processed text: ″i am getting 2 old to be mango gonna retire soon and be joesh #ROFL Happy face smiley″

Step 2: Covert all text in lower-case characters

Original text: ″I am getting 2 old to be mango Gonna retire soon and be joesh #ROFL″

Processed text: ″i am getting 2 old to be mango gonna retire soon and be joesh #rofl happy face smiley″

Step 3: Removing stop words

Original text: ″i am getting 2 old to be mango gonna retire soon and be joesh #rofl″

Processed text: ″getting 2 old mango gon na retire soon joesh #rofl happy face smiley″

Step 4: Removing special characters

Original text: ″getting 2 old mango gon na retire soon joesh #rofl″

Processed text: ″getting 2 old mango gon na retire soon joesh rofl happy face smiley″

Step 5: Removing numbers

Original text: ″getting 2 old mango gon na retire soon joesh rofl″

Processed text: ″getting old mango gon na retire soon joesh rofl happy face smiley″

As some features are directly propositional to the speed of model training. So as the feature increases the speed of the model, training is also increased [44]. Here, I am not removing hashtags because it gives valuable information. For example, when users talk about specific foods which are not common but hashtags will include #burger#delicious than I can quickly identify that the user is talking about a burger or some other food. So, I just removed the special characters while keeping the hashtagged text.

3.4.2 Database

To calculate the caloric value, we need two types of datasets: first for food and its caloric values and second for activity and its caloric-burn value. When I try to gather a dataset for food, I find out there is not even a single dataset available which includes the different types of food items and their nutrition values. At present, Canadian Food Nutrient Database and USDA Food Composition Databases are the main sources of information related to food and nutrition facts of the food in Canada. But the limitation with these databases is the lack of data in terms of cuisines such as "Chicken masala" or "Penne arrabiata." Usually, people tweet about cuisines or dishes they eat during lunch or dinner at a restaurant or any other place. It means the present dataset is very domain-oriented for things such as fast food, vegetables or frozen foods, but they will not contain all the major types of food that people talk about on social media as shown before. Now, there are two more problems after getting the dataset: to find what food the users talk about and to find the caloric value of that specific food. To solve this problem, we need a new dataset that combined different food domains which contain all major foods and their different nutrition values. That is why, I created a new dataset called "Food in one" which includes a combination of all open source datasets such as the Open Food Facts which is a major source of food names, Canadian Nutrient File and

USDA Food Composition Databases. Table 1 shows the structure of current food dataset:

Table 3.1: Food database

Name	**Data**
food_name	Name of the food
food_ingredients	Ingredients use to make the food
fat_100g	Fat per 100g of food
energy_100g	Energy value per 100g of the food
carbohydrate_100g	carbohydrate value of that food at 100g

Our newly created dataset contains 338,889 foods with all there information.This is an open source database at datalab.science. This includes all different types of major food sources like fruits, vegetables, fast food and regular food. In our dataset, more than 70% food items are from the US, Canada, and France. This is because our focus is mainly on Canada's health situations.

To understand the nutrition value of all food items in the database I used a normalized KDE. Figure 3.2 is the Normalized Kernel Density Estimation(KDE) diagram of all the food that is present in the dataset. This is along with their nutrition values including fat, carbohydrate and energy per 100 grams. As we can see in the second bar chart energy values mostly lie between mid-range while the fat bar chart has diverse values from an extreme high to an extreme low. This represents the diverse nature of our dataset that includes various type of foods.

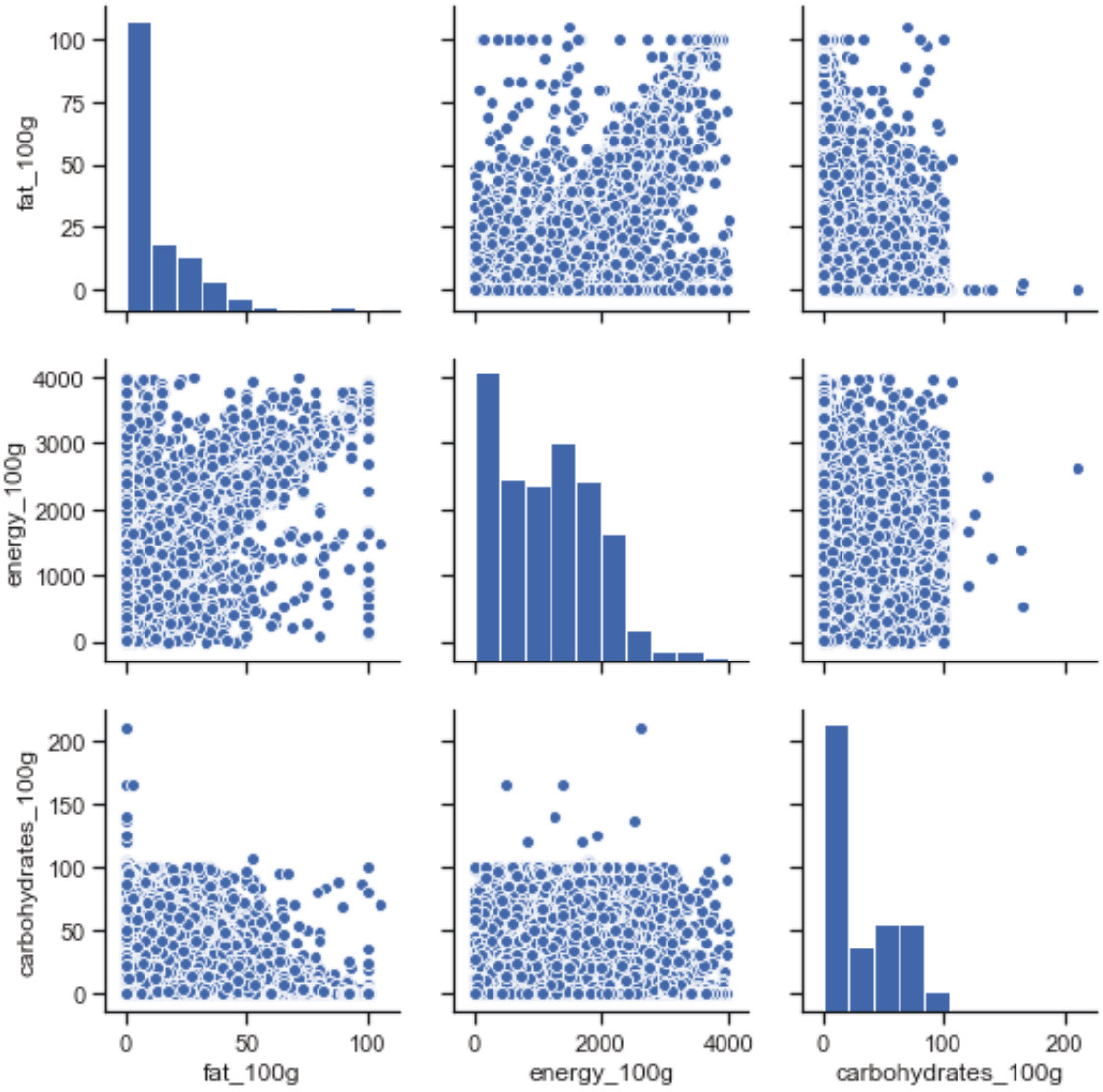

Figure 3.2: Nutrition value of all food in database

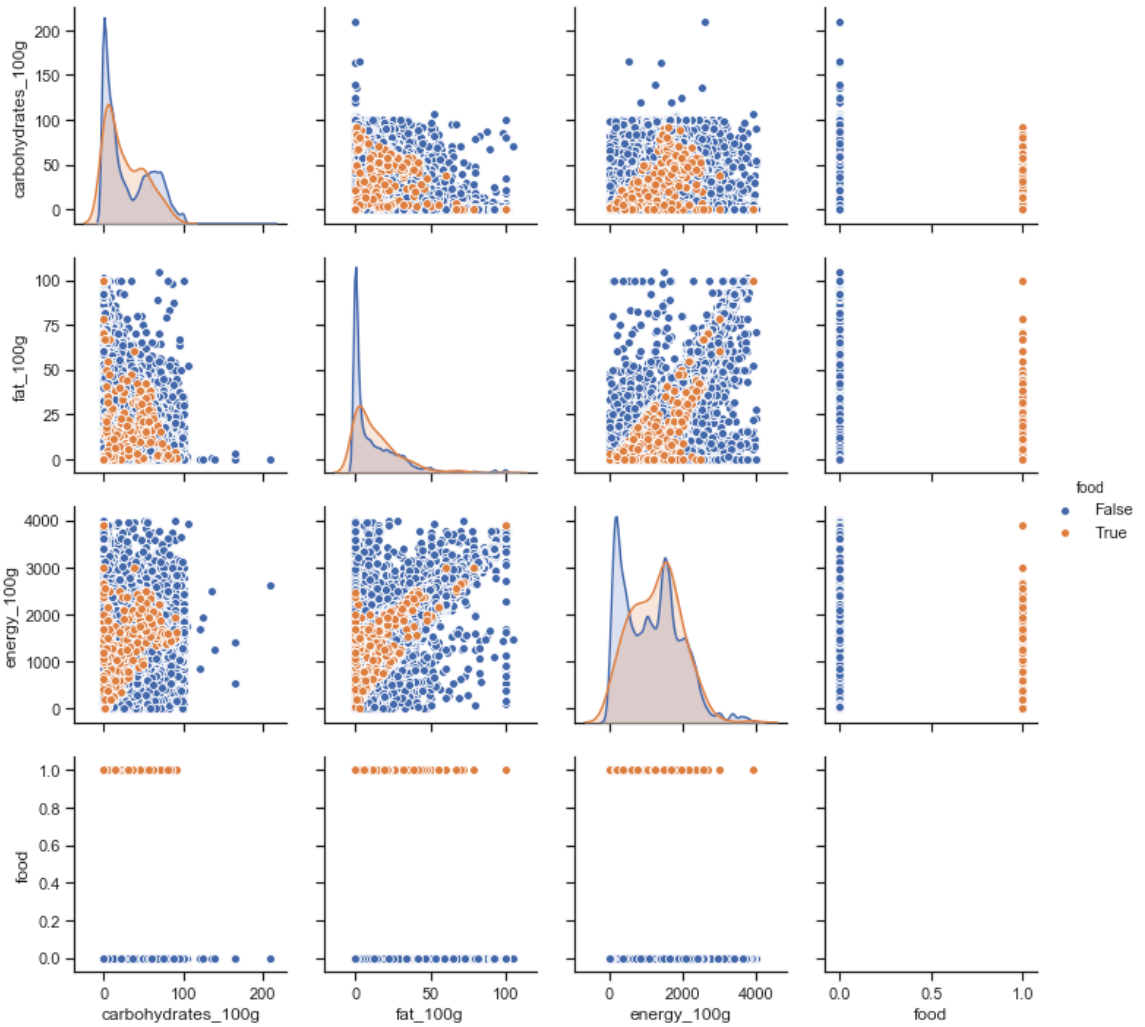

Figure 3.3: Nutrition values of vegan vs non-vegan food in database

Figure 3.3 shows the normalized KDE graph of nutrition values of vegan and non-vegan food. Orange represents the vegan food, and blue represents the non-vegan food. The results show that the distribution is quite similar for products with "Vegan" labels. As shown in the Figure 3.3 non-vegan food has high fat and energy values when compared to vegan foods on average. While the scatter graph, between carbohydrates and fat, shows a vegan diet has a lower energy value when compared to non-vegan foods with regards to the same amount of carbohydrate. The last raw scatter graph shows that all our food is categorized as vegan or non-vegan food.

Input: List of food name in dataset

Result: Food is Vegan or Non-Vegan

FoodDataset;

while *Food_in_FoodList* **do**

    name = Food;

    if *name contains "vegan"* **then**

        flag = True;

    else

        flag = False;

    end

end

Algorithm 1: Identifying food is vegan or non-vegan

To differentiate between vegan and non-vegan food, we first find out if the word "vegan" is present beside the name of the food in the database as shows in Algorithm 1. Then we also add vegetables and fruits as well as juice in the vegan food category. Any other foods are considered as non-vegan food.

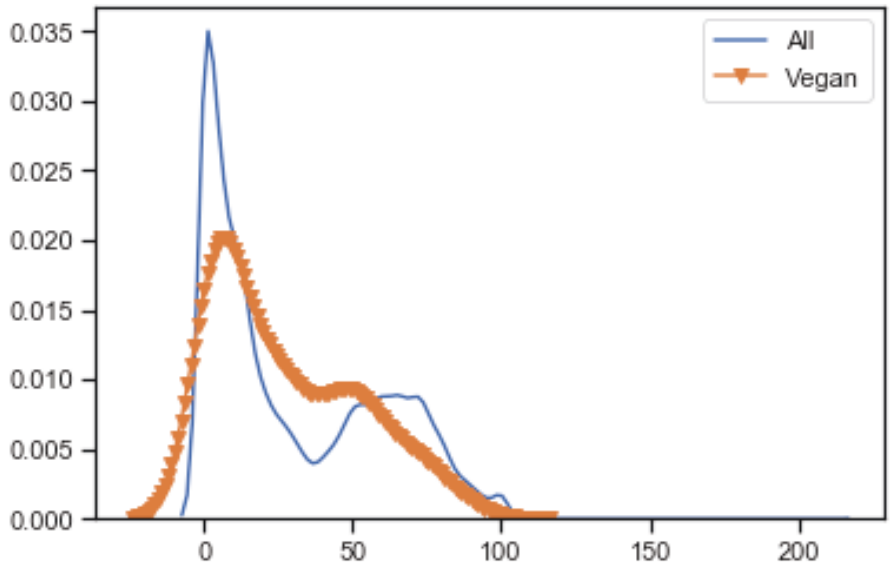

Figure 3.4: KDE of carbohydrates per 100g

Figure 3.4 shows the KDE graph of carbohydrates per 100g concerning the distribution between vegan and non-vegan foods. It also shows that some non-vegan food has high carbohydrate content than vegan foods. While, in other aspects of nutrition, the gap between vegan and non-vegan food is not as big.

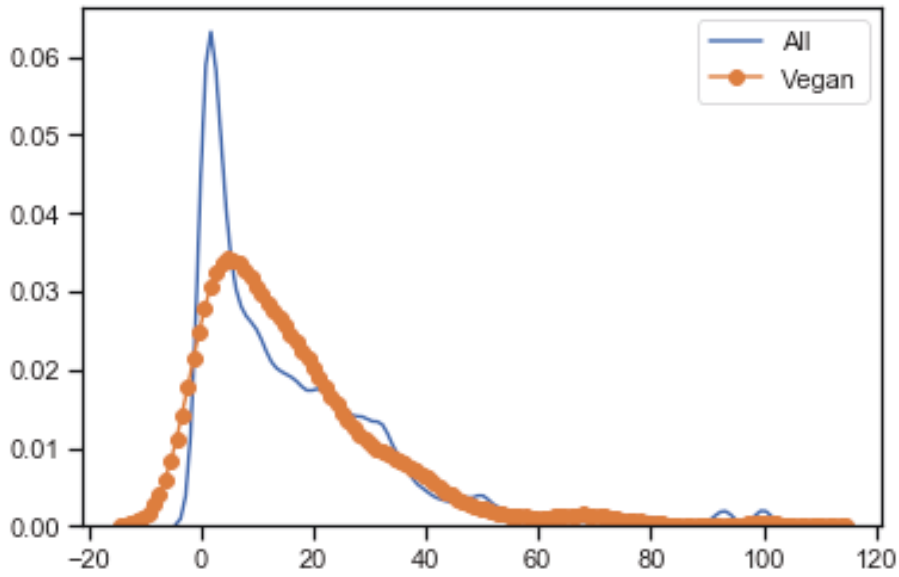

Figure 3.5: KDE of fat per 100g

Figure 3.5 shows the KDE graph of fat per 100g distribution between vegan and non-vegan foods. It also shows that some non-vegan foods have sharply high-fat content than vegan products. In other aspects, there is not much of a big difference between vegan and non-vegan foods, for example, fat content.

Figure 3.6 shows the KDE graph of energy per 100g distribution between vegan and non-vegan food.

In order to analyze public health, the second database we need is an activity database [39]. Where we can take the average activity time and relate it to caloric values. I chose to go with most common activities that are done and that are posted by people on social media. It now contains 1400 different activities and their caloric values that are available. To calculate the average caloric value, I fixed weight

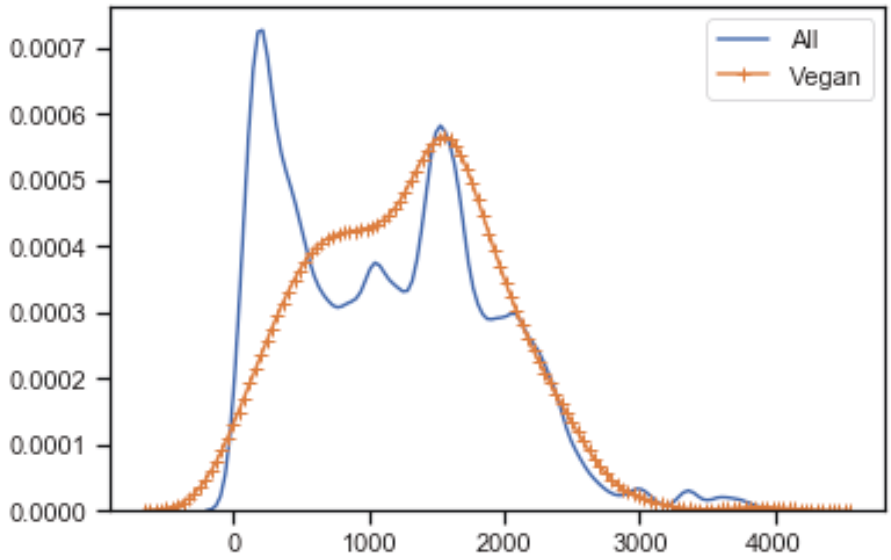

Figure 3.6: KDE of energy per 100g

and metabolism with average Canadian weight that is 80.3 kg. Table 2 shows the attributes of the activity dataset:

Table 3.2: Activity database

Name	**Data**
activity_name	Activity name
caloric_value	Caloric value of the activity

To analyze public health on a large scale, I am considering the Twitter dataset as the primary source for data. This will be used to do basic querying and analysis of the system at a large scale; I have developed an Elasticsearch based analysis system for real-time querying and searching of Twitter data [42]. From that system, we have taken 99,999,986 tweets between 2018 and 2019 for this paper.

3.4.3 Phrase Detection

In social media, a text data phrase gives more information than a single word. In the algorithm presented previously [39], one of the limitations is the inability to

understand the phrases of multi-words. For example, when anyone tweets ”You are the apple of my eye”, it considers apple as food. In our system apple of my eye is considered as a single phrase, which has a specific meaning. One common example is ”you are a smart cookie,” where ”smart cookie” has meaning as a phrase. To overcome this limitation I have added new features during the training of the NLP algorithm. This algorithm is developed by Jake Ryland, and it is based on the distance between two words. This text partitioning algorithm is based on William's fine-grained text segmentation algorithm. It considers the whole text as two parts: word and non-word tokens. The important feature of this algorithm is that it considers non-word tokens as a linker between two words. For example, in the phrase ”apple of my eye,” ”of” and ”my” non-words works as a joiner of the single phrase with singular meaning.

Input: List of words of the text - tokens
Result: List of tokens as Phrase - lexemes

```
phrase_detection(tokens):
lexemes[]
N = length(tokens)
while N do
    index = (N+1):1
    foreach i in index do
        form = join(token[0:i])
        remaining = tokens[i:N]
        if form related lex then
            lesemes = lexemes.add(form)
            if length(tokens)=1 then
                pass
            else
                tokens = reamining
            end
        end
        break
    end
end
return lexemes
```

Algorithm 2: Phrase detection algorithm

Algorithm 2 is a phrase detection algorithm that uses a concatenation operation. That links tokens together to create forms and then finds out how related the form is to the lexicon. If the form not correlated with the lexicon then the next possible form is analysed. If the form is related to the lexicon it is considered as a phrase.

Algorithm 2 is based on Boundary-based multi-word expression segmentation with text partitioning by J. Williams [45]. This algorithm focuses on the next possible word pair, which means a lower precision and efficiency for complex bound phrases. But the phrase information will be derived from a gold standard dataset. For example, Supersense-tagged Repository of English with Unified Semantic [45] and Riter and Lowlands dataset of superscience-annotated tweets for the SemEval [45]. Due to that pre-information of the phrases finding, results with simple and common phrases are easy. Below is the result of the phrase extraction algorithm I used:

text: ”I saw the sweet potatoes.”
phrase:”[’sweet’, ’potatoes’]”

text: ”My daughter is an apple of my eyes.”
phrase: ”[’apple’, ’eyes’, ’daughter’]”

Our results show that, the phrase detection algorithm will analyze the text and predict phrases for example the phrases ”sweet potatoes” and ”apple eyes daughter” are a single phrase. Even though our data cleaning process removes stop words the phrase detection algorithm can still detect a complicated sentence like ”my daughter is the apple of my eye” as the phrase ”apple eyes daughter”.

3.4.4 Model Training

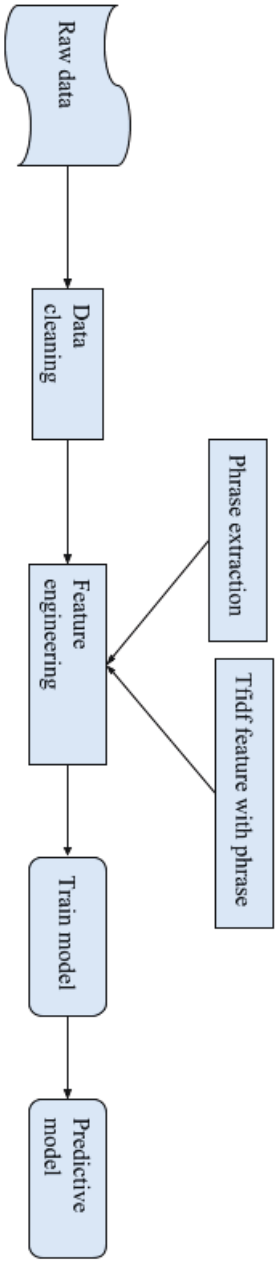

Figure 3.7: Processing pipeline of system

In the next step, I train the machine learning model for classification of the tweet. As shown in Figure 3.7, our first step is to take the raw data and clean it (explained in data cleaning subsection under methodology section). After cleaning the data, I then perform feature engineering. In feature engineering, I create tf-idf and phrase extraction. Here, I was using phrases as a single word and used it as a feature in tf-idf. After that, I have trained the model and then used that pre-trained model for binary classification of the tweet for food and non-food tweets.

I use two types of features: tf-idf with phrases and word embedding (2-D feature space) with 2-gram. Tf-idf is used for Naive Bayes, Logistic regression, and Random forest. But to use a complex deep-learning model, such as LSTM, BART or R-LSTM, I need large amount of training data. At present, not a single large scale training data is available for food classification which limits us to use basic deep-learning model CNN, Shallow neural network or RNN. Furthermore, I can not use 1-dimensional feature for the neural network. Instead I use word embedding as a feature in neural network. The word embeddings and phrases are selected as features for Shallow neural network, Convolutional neural network and Reinforcement neural network.

3.5 Analysis and Results

In previous papers, researchers tried to find the caloric value through non-NLP or basic NLP algorithms. Because of that, the false positive rate of data is high, and this will decrease the accuracy of the result. False positive errors will start to increase as the data amount increases. This affects the accuracy and accountability of the system. Many models are available for the classification of text, but in our case, the binary form of classification is much easier than multi-class classification. It also removes the necessity of advance deep learning algorithms.

The first algorithm that I have tested is Logistic Regression(LR). This measures

the relationship between one or more categorical dependent and independent variables. It will be estimated through logistic (sigmoid is more common presently) function.

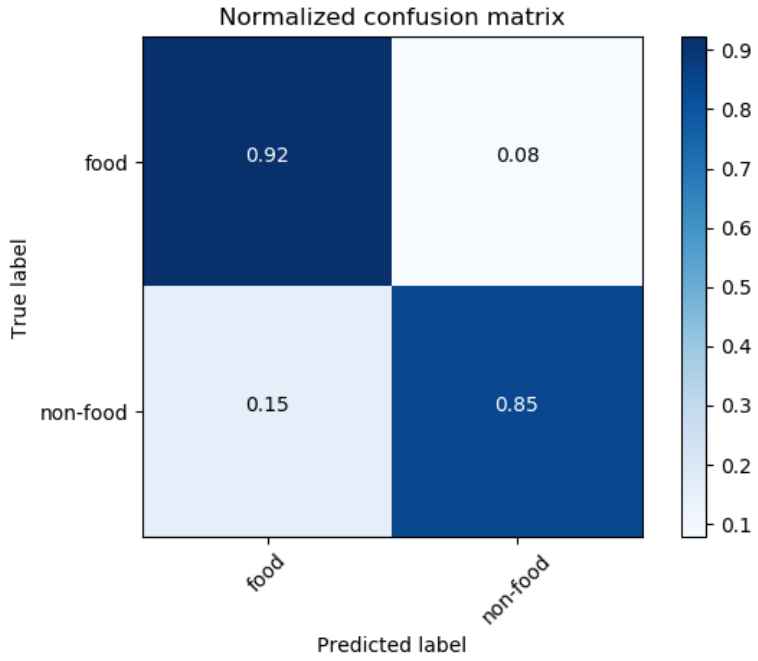

Figure 3.8: Logistic regression confusion matrix

Figure 3.8 shows the confusion matrix of LR. It also shows that algorithm can successfully identify 92% of food tweets. While recognizing only 85% of the non-food tweet as non-food. But the false positive ratio is very high, 15% which landed into more noisy data.

Figure 3.9 shows the training curve of LR. As we can see, as the number of training samples increases its accuracy is also increasing.

Our second algorithm is the Naive Bayes(NB) algorithm with tf-idf features on a word level. This classification of algorithm techniques is based on Bayes' theorem. This assumes Independence between predictors. Meaning, it implies one feature in the model is unrelated to another feature.

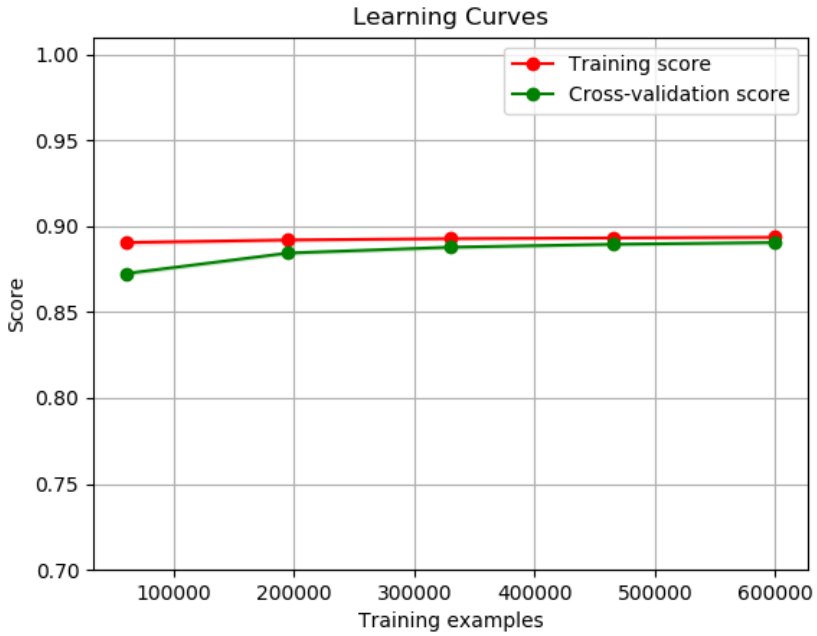

Figure 3.9: Logistic regression training curve

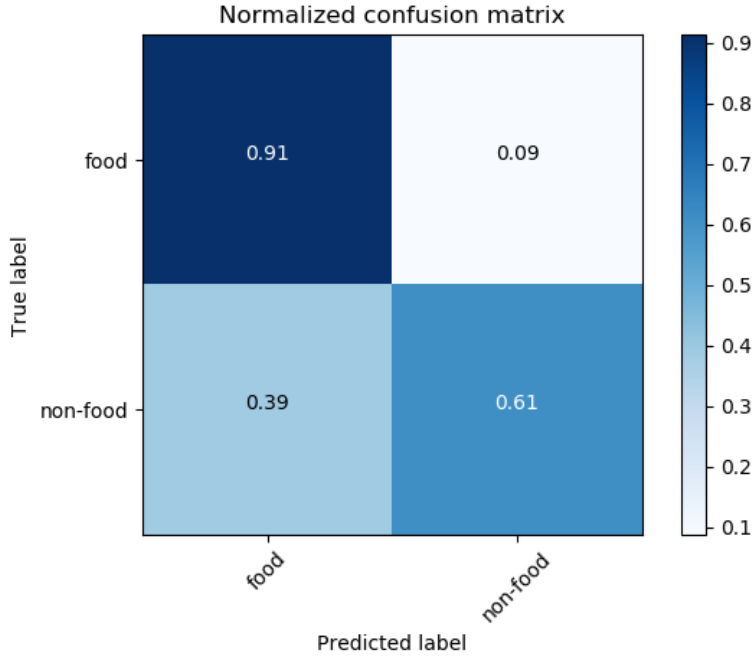

Figure 3.10: Naive Bayes confusion matrix

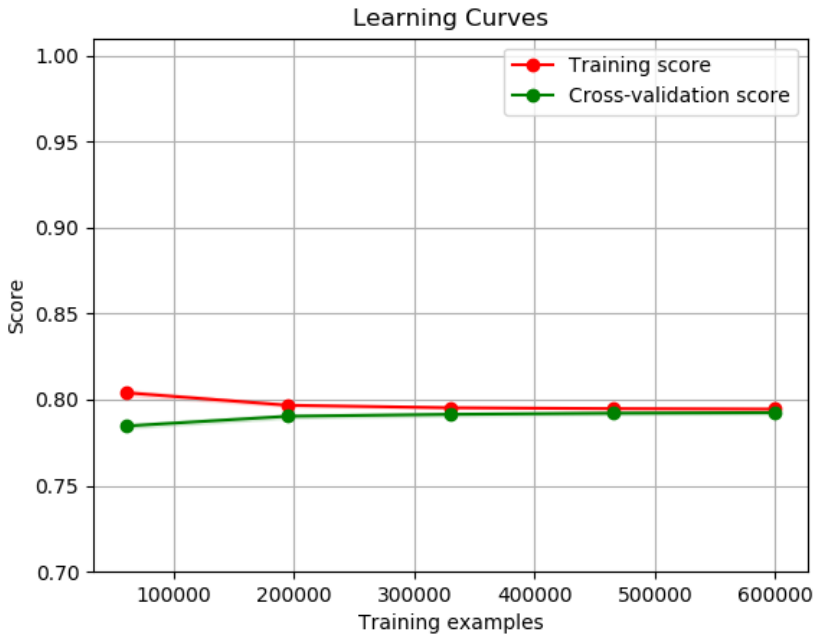

Figure 3.11: Naive Bayes training curve

Figure 3.10 shows the confusion matrix of NB. It also shows that algorithm can successfully identify 91% of food tweets as food tweet, while only 61% of the non-food tweets as non-food tweets. And a high rate of false positive as 39% which is quite high to get accurate results. Figure 3.11 shows the training curve of NB. It also shows the accuracy of the algorithm and also increases constantly with an increment of data. But after 400K samples, the accuracy of the algorithm is almost constant.

The third model is the Random Forest(RF) model. This is a type of bagging model, and it is a part of the tree based model. An advantage of this model is that it gives more accurate pr editions when comparing it to any simple CART or regression model in specific scenarios. Figure 3.12 shows the confusion metric of RF. It also shows that algorithm can successfully identify 97% of the food tweets as food tweets, while 88% of the non-food tweets are recognized as non-food tweets. On the other side, the false positive rate is also as low as 12%. This result shows the

highest accuracy among the other tested models. Figure 3.13 shows the learning curve of RF. It also shows that the accuracy of the algorithm increases as the data size increases.

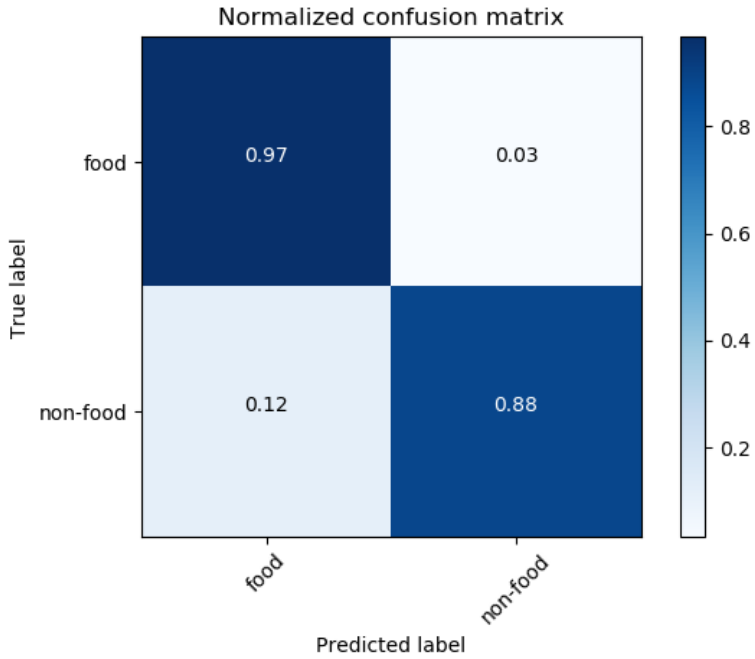

Figure 3.12: Random forest confusion matrix

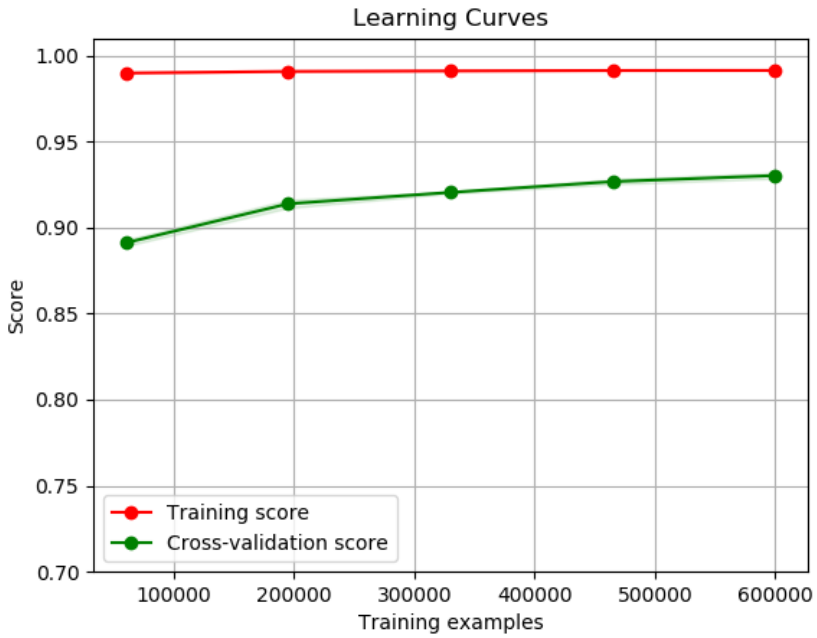

Figure 3.13: Random forest training curve

Table 3.3: Model accuracy

Model	**Accuracy**
Naive Bayes	79.202%
Linear Regression	89.155%
Random Forest	93.406%
CNN	60.142%
RNN-GRU	60.034%
SVM	56.031%

Table 3.3 shows the accuracy of different models. From above the analysis, the random forest model gives us the best results for the binary classification of tweets in food and non-food categories. The next step is to get information about the calories and the user's activity based on our dataset. We only focus on three different values: caloric value they gain from their food (C_{in}), the caloric value

they burn from their activity (C_{out}), and their caloric ratio from the first two values (C_{rat}). (Equation 1). Those three values are co-related to the 37 measures of the well being and health. In Lexicocalorimeter, they found a statistically strong correlation between high blood pressure, inactivity, diabetics and obesity rates [39]. Now to count these three values C_{in}, C_{rat}, C_{out} we depend on the benefits we get from an individual tweet.

Input: Tweet text (T),list of food, dictionary of caloric value of food with food name as key
Result: C_{in}

```
Global Frequency_Matrix = {}
Global FoodList = list_of_food
Global FoodDict = dictionary of caloric value of food with food name as key

Calorie_consumption(text):
phrase = phrase_detection(text)
while word in text do
    if phrase in FoodList then
        cal = FoodDict[phrase]*Frequency(phrase);
        Frequency_Matrix[phrase]++;
    else
        if word is FoodList then
            cal = FoodDict[word]*Frequency(word);
            Frequency_Matrix[word]++;
        else
            return 0
        end
    end
    return 0
end

calorie = 0
Get_C_in(Tweets):
while tweet in Tweets do
    calorie = calorie + Calorie_consumption(tweet)
end
return C_in= calorie/sum(Frequency_Matrix)
```

Algorithm 3: C_{in} algorithm

Algorithm 3 gives the value of C_{in} value for C_{rat} as shown in Equation 3.1. In the

Input: Tweet text (T)
Result: C_{out}
Global Frequency_Matrix = {}
Global ActivityList = list_of_activity
Global ActivityDict = dictonary of caloric value of activity with activity name as key.

Calorie_burn(text):
while *word in text* **do**
    if *word in ActivityList* **then**
        cal = ActivityDict[word]*Frequency(word);
        Frequency_Matrix[word]++;
    else
        return 0;
    end
    return 0
end

calorie = 0
Get_C_{out}(Tweets):
while *tweet in Tweets* **do**
    calorie = calorie + Calorie_burn(tweet)
end
return C_{out}= calorie/sum(Frequency_Matrix)

Algorithm 4: C_{out} algorithm

first step we take the tweets in our dataset and get the caloric value of each tweet using the Calorie_consumption function. In that function we take the text and find any food phrases. In the second step, the individual words or phrases found in the tweet and compared it to the food dataset. If it present, we take caloric value of the word or phrase from food dictionary. Then we multiply the calorie of that food with the frequency of that word in the text. It is then stored in the Frequency matrix and to get normalized C_{in} value we sum all calorie values from all the tweets and divide by the sum of the frequency matrix.

Algorithm 4 gives the value of C_{out} value for C_{rat} as shown in Equation 3.1. In the first step we take the tweets in our dataset and get the caloric values of each possible word in the tweet. Than we check each word from each tweet with the activity dataset. The dataset will give a caloric burn value for each tweet. To normalized the C_{out} value we take the summation of caloric values and divide it by the frequency of each activity phrase. To count the C_{out} value, we need to count how much calories a person can burn from a particular activity. For that, we make an assumption where 80.7 kilograms is the standard average weight of Northern American adult.

Table 3.4 is the result of the 100M tweets gathered between 2018 and 2019. We choose 50K tweets pertaining to food and 50k tweets pertaining to activity from each province and territory randomly, which combine to form our 100M tweet dataset [46]. A free Twitter API was used to collect the data without any filters and therefore makes our collection of tweets random. Table 3.4 shows the top 10 foods in Canada, and it clearly shows that junk food and hot drinks are the most common.

Table 3.4: Top 10 Food in Canada

Rank	**Food**	**No. Tweets**
1	coffee	38,785
2	burger	35,166
3	pizza	34,369
4	noodles	27,891
5	cake	18,456
6	pie	17,982
7	juice	16,711
8	tea	16,631
9	fruits	15,987
10	veggies	11,473

Table 3.5: Top 10 Activity in Canada

Rank	**Activity**	**No. Tweets**
1	watching (seeing)	42,489
2	reading	31,762
3	walking	28,127
4	running	27,838
5	drinking	27,339
6	sitting	24,347
7	cooking	22,561
8	skiing	18,947
9	gym	16,585
10	playing	14,191

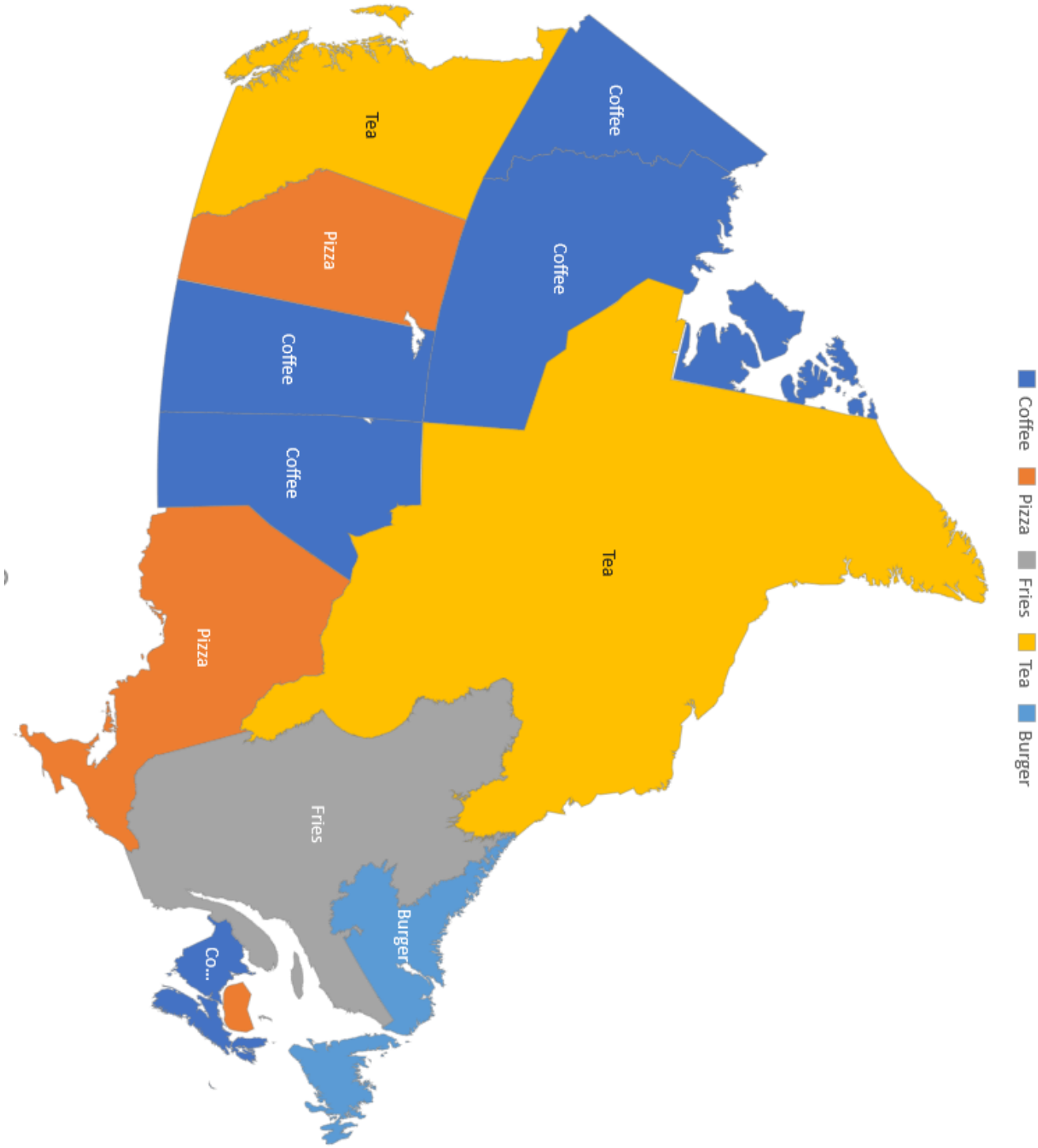

Figure 3.14: Canadian's Tweets on food

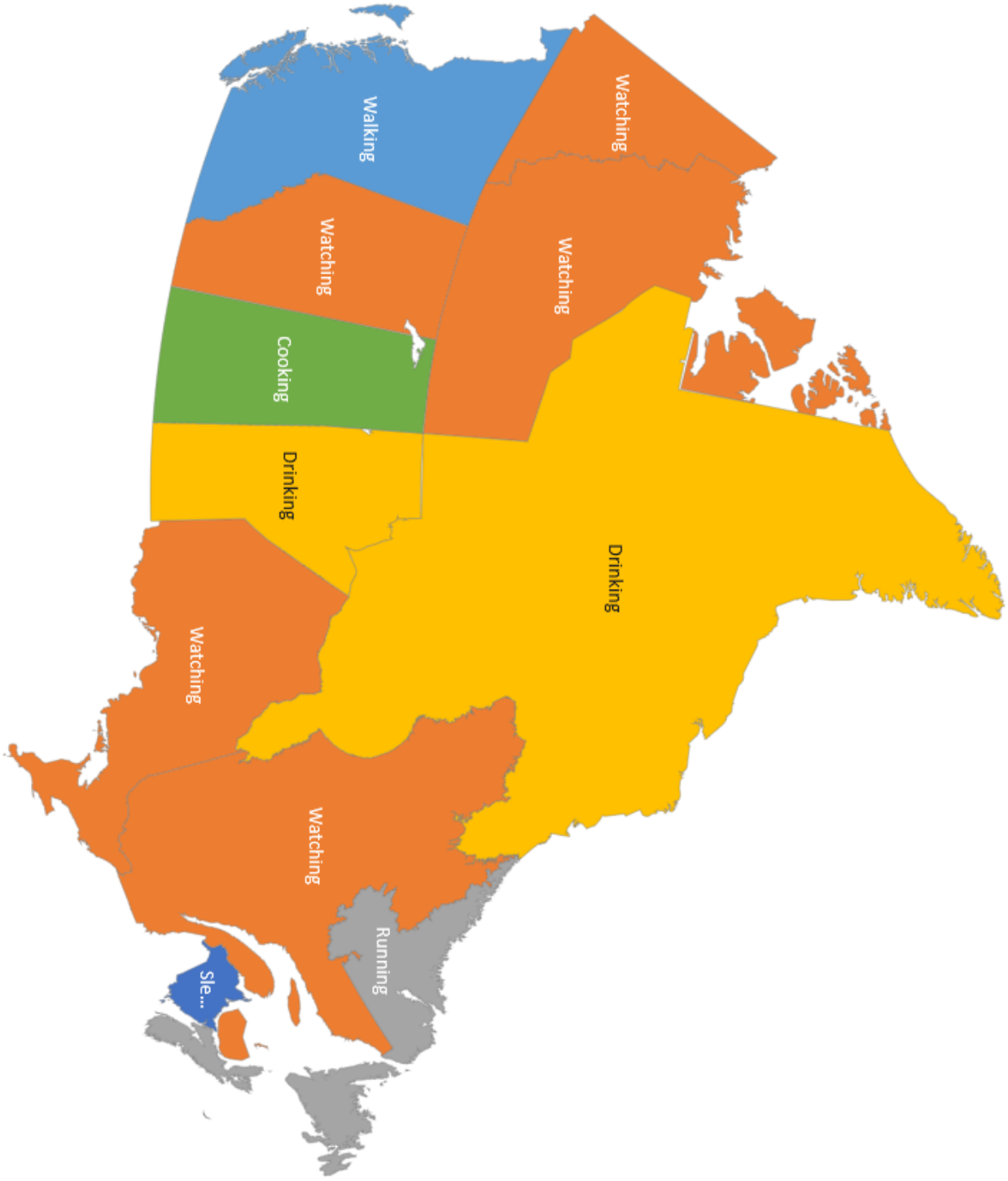

Figure 3.15: Canadian's Tweets on activity

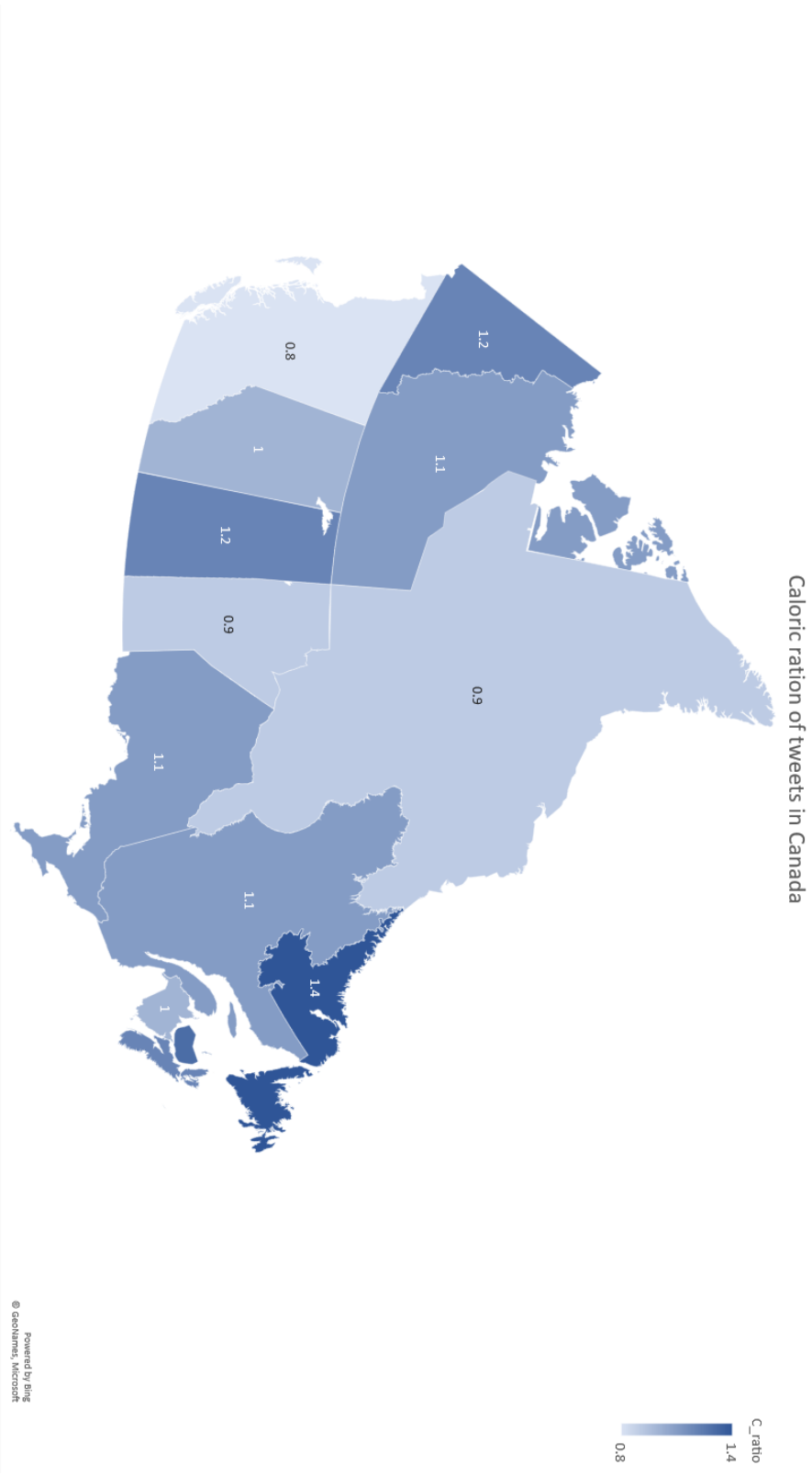

Figure 3.16: Caloric ratio of tweets in Canada

Table 3.5 is the list of the top 10 activities in Canadian tweets. It clearly shows that more and more people are choosing to watch something regularly instead of physical exercise. It also shows that walking and running are the most common exercises people do. This means physical inactivity is increasing throughout Canada.

Figure 3.14 shows the most common foods people tweeted about in different provinces and territories of Canada. As it is seen that in Ontario and Alberta, the most common foods tweeted are Pizza, while Quebec's most common food is Fries. It also shows that coffee is the most common tweeted drink in provinces like Manitoba, Saskatchewan, Yukon and Northwest Territories. As compared to tea, which is more common in British Colombian's tweets.

Figure 3.15 is about the most common activities people tweeted about in different province and territories of Canada. The result shows watching (watching TV) is the most common activity in dense population provinces of Canada, which includes Ontario, Quebec, Alberta, Yukon and Northwest Territories. This shows that there is less physical activity among people in these provinces, which is an alarming situation when looking at the individual's food consumption versus activity they do to burn calories.

If we go with Canadian government numbers from there own health analysis, it shows that Ontario and Quebec have Canada's 38.3% and 23.2% of the population respectively. When we combine both of them, it is 61.5% population (census). As Caloric ratio is highly correlated to blood pressure and obesity [39], which according to the Lexicoloricmeter article that 77.92% population has a higher chance of getting obese, and higher blood pressure which is an alarming situation. Our result correlates to the result given by the Canadian Institute of Health Information report "Obesity in Canada" [47]. It also shows the rapid growth of Obesity in Ontario and Quebec.

Figure 3.16 shows the Caloric ratio based on equation 1, where C_{in} is counted

through Equation 2 and C_{out} is counted through Equation 3. If the ratio is greater than 1 that means that the province's consumption is greater than the caloric usage. The opposite is true when the ratio is less than 1. According to Figure 3.16, Yukon, Newfoundland and Labrador, and Saskatchewan's caloric consumption is higher than their caloric burning at this instance. Also when looking at the Northwest Territories, Manitoba, and British Colombia have a caloric burn that is higher than their caloric consumption. Caloric ratio is highly correlated to blood pressure and obesity [39]. When looking at Figure 3.16, 77.92% population has a caloric ration greater than 1.0. This can cause greater chance of getting obese, and higher blood pressure. This population estimation is based on the population numbers from the 2016 Canadian Census and was calculated by when adding up the populations from each individual province with a 1.0 caloric ratio or higher. This is alarming because it represents such a huge population of Canada.

We further decided to look at the obesity percentages for the provinces of Canada to see if Figure 3.16 showed similarities. When looking at the obesity in Canada report, published in 2017, the order of provinces from lowest obesity rate to highest obesity rate is the following: British Columbia, Quebec, Ontario, Alberta, Manitoba, Saskatchewan, Newfoundland and Labrador [47]. When comparing that with the lowest to highest rank in the Figure 3.16, British Columbia is the lowest, Quebec and Ontario are tied, Alberta and Manitoba are switched in ranks, Saskatchewan and Newfoundland and Labrador are the highest. The significance of the comparison above is that it shows a strong correlation between Figure 3.16 and the data from the report. The provinces with the highest ratios also have the highest obesity rates and the opposite is also true. This goes as far as to show that our data can be just as reliable as a published Canadian report. By using this knowledge the Canadian government can target healthy living programs in the provinces that need it like Newfoundland and Labrador.

Figure 3.16 only shows one caloric ration per province from the tweets collected at that instance. In order to become obese you need to constantly have a ratio greater than 1 over the span of weeks. That way you are consuming more calories that you are burning which leads to the gaining of weight. Therefore our Figure 3.16 can be considered one data point in an obesity rate trend and in the future can be added with many other points taken in different times to accurately show the rate of obesity in Canada. By this logic you can also see the trend from one point to another and immediately see if the programs implemented to counteract obesity have worked. The result is going from tweets to a real time analysis of the rate of obesity in Canada.

CHAPTER 4
SUMMARY

4.1 Conclusion

Developing high-performance machine learning model with a limited amount of training data is always a challenge, as it restricts the use of more complex deep learning and neural models. Our model gives 93.406% accuracy in binary classification of food and non-food tweet. This result shows that social media analysis on a large scale with the use of better NLP algorithms can help us to identify food and activity related tweets more accurately. This helps us to gain a larger perspective on daily activities and its effect on people's health. Our results convey a complex relationship between health and social media. The presented approach is faster when compared to traditional survey methods causing data to be readily available as well a close representation of real time. This was all done via Elasticsearch which provides a functional system to store, pre-index, search and query for large scale data in real-time. In particular, the capability of expanding the cluster size without stopping service as per user's requirement makes it suitable for this application. This research provides insights on how to standardize and configure the processes of Elasticsearch which result in increased analysis efficiency. To demonstrate the functionality and interactivity for users, the Kibana plugin was used as an interface. Proper configuration of Elasticsearch and Kibana makes real-time analysis of large scale data efficient and can help policy makers see the results instantaneously and in an accessible format that allows for decision making. The impact of research is huge as it can change how we view data analysis. Now because of social media analysis there is no need to rely on traditional methods for

health analysis that are much more expensive and time consuming. New Data is generated almost instantaneously from social media analysis which allows for real time accurate health reports. Its not a matter of if social media will be used for health analysis, its about making it reach a point where at any moment in time the government can get a detailed report on the health in Canada.

4.2 Future Work

The approach presented is faster compared to traditional survey methods, which make data readily available as well as a close representation of real time. Many promising future works, such as a more dynamic way to calculate calories based on age, gender and work profile, are possible here. One limitation is that it only recognizes the food when our model looks at the tweet, but leaves out the quantity of said food. Our model, for example, can not distinguish 1 apple from 10 apples. Adding sentiment of the text could be used for further classification of text. Furthermore, we can extend this work by analysing pictures' that people post on social media about their activity and food associated with twitter data.

REFERENCES

[1] C. for Disease Control, Prevention, *et al.*, *Health-related quality of life: Wellbeing concepts*, 2011.

[2] W. A. Social, "Global digital report 2018," *Https://wearesocial.com/blog/2018/01/global-digital-report-2018*, 2018.

[3] P. S. Dodds, K. D. Harris, I. M. Kloumann, C. A. Bliss, and C. M. Danforth, "Temporal patterns of happiness and information in a global social network: Hedonometrics and twitter," *PloS one*, vol. 6, no. 12, e26752, 2011.

[4] P. Cervellini, A. G. Menezes, and V. K. Mago, "Finding trendsetters on yelp dataset," in *Computational Intelligence (SSCI), 2016 IEEE Symposium Series on*, IEEE, 2016, pp. 1–7.

[5] E. Belyi, P. J. Giabbanelli, I. Patel, N. H. Balabhadrapathruni, A. B. Abdallah, W. Hameed, and V. K. Mago, "Combining association rule mining and network analysis for pharmacosurveillance," *The Journal of Supercomputing*, vol. 72, no. 5, pp. 2014–2034, 2016.

[6] O. Kononenko, O. Baysal, R. Holmes, and M. W. Godfrey, "Mining modern repositories with Elasticsearch," in *Proceedings of the 11th Working Conference on Mining Software Repositories*, ACM, 2014, pp. 328–331.

[7] Q. Liu, S. Kumar, and V. Mago, "Safernet: Safe transportation routing in the era of internet of vehicles and mobile crowd sensing," in *Consumer Communications & Networking Conference (CCNC), 2017 14th IEEE Annual*, IEEE, 2017, pp. 299–304.

[8] M. G. Kim and J. H. Koh, "Recent research trends for geospatial information explored by twitter data," *Spatial Information Research*, vol. 24, no. 2, pp. 65–73, 2016.

[9] M. D. Assunção, R. N. Calheiros, S. Bianchi, M. A. Netto, and R. Buyya, "Big data computing and clouds: Trends and future directions," *Journal of Parallel and Distributed Computing*, vol. 79, pp. 3–15, 2015.

[10] D. Cea, J. Nin, R. Tous, J. Torres, and E. Ayguadé, "Towards the cloudification of the social networks analytics," in *Modeling Decisions for Artificial Intelligence*, Springer, 2014, pp. 192–203.

[11] J. Bai, "Feasibility analysis of big log data real time search based on hbase and elasticsearch," in *Natural Computation (ICNC), 2013 Ninth International Conference on*, IEEE, 2013, pp. 1166–1170.

[12] C. Bsch, P. Hartel, W. Jonker, and A. Peter, "A survey of provably secure searchable encryption," *ACM computing surveys*, vol. 47, no. 2, 18:1–18:51, Aug. 2014, eemcs-eprint-24788.

[13] P. Kumar, P. Kumar, N. Zaidi, and V. S. Rathore, "Analysis and comparative exploration of elastic search, mongodb and hadoop big data processing," in *Soft Computing: Theories and Applications*, Springer, 2018, pp. 605–615.

[14] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan, "The rise of big datai on cloud computing: Review and open research issues," *Information Systems*, vol. 47, pp. 98–115, 2015.

[15] H. Yang, M. Park, M. Cho, M. Song, and S. Kim, "A system architecture for manufacturing process analysis based on big data and process mining techniques," in *Big Data (Big Data), 2014 IEEE International Conference on*, IEEE, 2014, pp. 1024–1029.

[16] G. Stelzer, I. Plaschkes, D. Oz-Levi, A. Alkelai, T. Olender, S. Zimmerman, M. Twik, F. Belinky, S. Fishilevich, R. Nudel, *et al.*, "Varelect: The phenotype-based variation prioritizer of the genecards suite," *BMC genomics*, vol. 17, no. 2, p. 444, 2016.

[17] S Bagnasco, D Berzano, A Guarise, S Lusso, M Masera, and S Vallero, "Monitoring of iaas and scientific applications on the cloud using the elasticsearch ecosystem," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 608, 2015, p. 012 016.

[18] D. Chen, Y. Chen, B. N. Brownlow, P. P. Kanjamala, C. A. G. Arredondo, B. L. Radspinner, and M. A. Raveling, "Real-time or near real-time persisting daily healthcare data into hdfs and elasticsearch index inside a big data platform," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 595–606, 2017.

[19] J. B. Coronel and S. Mock, "Designsafe: Using elasticsearch to share and search data on a science web portal," in *Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact*, ACM, 2017, p. 25.

[20] F. Yang, E. Tschetter, X. Léauté, N. Ray, G. Merlino, and D. Ganguli, "Druid: A real-time analytical data store," in *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, ACM, 2014, pp. 157–168.

[21] K. J. Burkitt, E. G. Dowling, and T. R. Branon, *System and method for real-time processing, storage, indexing, and delivery of segmented video*, US Patent 8,769,576, 2014.

[22] *Elasticsearch- elastic.co*, Available at `https://www.elastic.co/guide/en/elasticsearch/reference/6.2/index.html` (Last Accessed on April 30, 2018).

[23] C. Gormley and Z. Tong, *Elasticsearch: The definitive guide: A distributed real-time search and analytics engine*. " O'Reilly Media, Inc.", 2015.

[24] *Your Window into the Elastic Stack*, Available at `https://www.elastic.co/products/kibana` (Last Accessed on April 30, 2018).

[25] *Python Elasticsearch Client*, Available at `https://elasticsearch-py.readthedocs.io/en/master/` (Last Accessed on April 30, 2018).

[26] *Java Elasticsearch library - Elastic*, Available at `https://www.elastic.co/guide/en/Elasticsearch/client/java-api/6.2/index.html` (Last Accessed on April 30, 2018).

[27] *Getting Started with Logstash*, Available at `https://www.elastic.co/guide/en/logstash/current/getting-started-with-logstash.html` (Last Accessed on April 30, 2018).

[28] A. Karami, A. A. Dahl, G. Turner-McGrievy, H. Kharrazi, and G. Shaw Jr, "Characterizing diabetes, diet, exercise, and obesity comments on twitter," *International Journal of Information Management*, vol. 38, no. 1, pp. 1–6, 2018.

[29] K. Robinson and V. Mago, "Birds of prey: Identifying lexical irregularities in spam on twitter," *Wireless Networks*, pp. 1–8, 2018.

[30] P. Grover, A. K. Kar, and G. Davies, "technology enabled healthi–insights from twitter analytics with a socio-technical perspective," *International Journal of Information Management*, vol. 43, pp. 85–97, 2018.

[31] D. Lazer, R. Kennedy, G. King, and A. Vespignani, "The parable of google flu: Traps in big data analysis," *Science*, vol. 343, no. 6176, pp. 1203–1205, 2014.

[32] M. J. Paul and M. Dredze, "You are what you tweet: Analyzing twitter for public health.," *Icwsm*, vol. 20, pp. 265–272, 2011.

[33] S. Hill, R. Merchant, and L. Ungar, "Lessons learned about public health from online crowd surveillance," *Big Data*, vol. 1, no. 3, pp. 160–167, 2013.

[34] J. C. Eichstaedt, H. A. Schwartz, M. L. Kern, G. Park, D. R. Labarthe, R. M. Merchant, S. Jha, M. Agrawal, L. A. Dziurzynski, M. Sap, *et al.*, "Psychological language on twitter predicts county-level heart disease mortality," *Psychological science*, vol. 26, no. 2, pp. 159–169, 2015.

[35] A. Culotta, "Estimating county health statistics with twitter," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 2014, pp. 1335–1344.

[36] S. Abbar, Y. Mejova, and I. Weber, "You tweet what you eat: Studying food consumption through twitter," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ACM, 2015, pp. 3197–3206.

[37] S. Medvedyuk, A. Ali, and D. Raphael, "Ideology, obesity and the social determinants of health: A critical analysis of the obesity and health relationship," *Critical Public Health*, vol. 28, no. 5, pp. 573–585, 2018.

[38] E. Diener, "Subjective well-being.," *Psychological bulletin*, vol. 95, no. 3, p. 542, 1984.

[39] S. E. Alajajian, J. R. Williams, A. J. Reagan, S. C. Alajajian, M. R. Frank, L. Mitchell, J. Lahne, C. M. Danforth, and P. S. Dodds, "The lexicocalorimeter: Gauging public health through caloric input and output on social media," *PLoS ONE*, vol. 12, no. 2, 2017. arXiv: 1507.05098.

[40] S. E. Alajajian, J. R. Williams, A. J. Reagan, S. C. Alajajian, M. R. Frank, L. Mitchell, J. Lahne, C. M. Danforth, and P. Sheridan Dodds, 2017.

[41] L. McIntyre, G. Jessiman-Perreault, C. L. Mah, and J. Godley, "A social network analysis of canadian food insecurity policy actors," *Canadian Journal of Dietetic Practice and Research*, vol. 79, no. 2, pp. 60–66, 2018.

[42] N. Shah, D. Willick, and V. Mago, "A framework for social media data analytics using elasticsearch and kibana," *Wireless Networks*, pp. 1–9, 2018.

[43] N. Shah, *Open source emoticons and emoji detection library: Emot(stable v2.2)*, https://github.com/NeelShah18/emot, Accessed: 2019-02-28.

[44] R. Batista, K. Pottie, L. Bouchard, E. Ng, P. Tanuseputro, and P. Tugwell, "Primary health care models addressing health equity for immigrants: A systematic scoping review," *Journal of immigrant and minority health*, vol. 20, no. 1, pp. 214–230, 2018.

[45] J. R. Williams, "Boundary-based mwe segmentation with text partitioning," *ArXiv preprint arXiv:1608.02025*, 2016.

[46] P. Mick, M. Parfyonov, W. Wittich, N. Phillips, and M. K. Pichora-Fuller, "Associations between sensory loss and social networks, participation, support, and loneliness: Analysis of the canadian longitudinal study on aging," *Canadian Family Physician*, vol. 64, no. 1, e33–e41, 2018.

[47] Government of Canada, Health info graphics, "Obesity in canadian adults: It's about more than just weight," Public health Agency of Canada, Ontario, Canada, Apr. 25, 2017.