



Ain Shams University
Faculty of Computer & Information Sciences
Information System Department

Event Detection on Social Media

By

Marwan Ahmed Mohamed	[Information Systems]
Riham Mohsen Sayed	[Information Systems]
Ahmed Hamdy Abdelsattar	[Information Systems]
Rana Hamdy Abdelkader	[Information Systems]
Hussien Ibrahim Hassan	[Information Systems]
Sama Hesham Sayed	[Information Systems]

Under Supervision of

Prof. Dr. Nagwa Badr
T.A. Amira Ali

Information Systems Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

June 2023

Acknowledgements

All praise and thanks to ALLAH, who provided us with the ability to complete this work. We hope to accept this work from us.

We are grateful of *our parents* and *our family* who are always providing help and support throughout the whole years of study. We hope we can give that back to them.

We also offer our sincerest gratitude to our supervisors, ***Prof. Dr. Nagwa Badr, and T.A Amira Ali*** who have supported me throughout my thesis with their patience, knowledge and experience.

Finally, We would like to thank our friends and all people who gave support and encouragement.

Abstract

Event detection on social media faces several challenges and problems as Social media platforms are often filled with a lot of noise and irrelevant information that can make it difficult to identify relevant events, Social media messages are often short and lack context, which can make it challenging to accurately identify and categorize events, and Social media generates a large volume of data, making it difficult to process and analyze all the information in a timely manner. The speed at which information spreads on social media means that events can quickly evolve and change, making it challenging to keep up with the latest developments, and it could be difficult to identify and verify accurate information. There are a lot of methods that can be used in event detection from these methods. Methods used in collecting data is keyword based methods by identifying relevant keywords and hashtags related to an event and using them to filter and classify social media messages , and machine learning algorithms which can be trained to automatically identify and categorize events based on patterns in social media data. There have been several recent developments and discoveries in the field of event detection. Some of the main results and newly observed facts like Deep learning techniques such as CNN, RNN and it has shown promising results in event detection on social media, and Emotion analysis it involves identifying and categorizing emotions in the messages which can improve the event detection process In conclusion, event detection on social media is a challenging and dynamic field that requires sophisticated algorithms and techniques to overcome the noise, ambiguity, and volume of data associated with social media platforms.

Abstract

يواجه اكتشاف الأحداث على وسائل التواصل الاجتماعي العديد من التحديات والمشكلات ، نظرًا لأن منصات الوسائط الاجتماعية غالبًا ما تمتلئ بالكثير من المعلومات الغير مهمة و التي تحتوي على احداث غير مهمة و ذلك يمكن أن يجعل من الصعب تحديد الأحداث المهمة بالنسبة للمستخدم ، غالبًا ما تكون رسائل وسائل التواصل الاجتماعي قصيرة وتفتقر إلى السياق الذي يمكن أن يجعل تحديد الأحداث وتصنيفها بدقة أمرًا صعبًا ، وتنتج وسائل التواصل الاجتماعي حجمًا كبيرًا من البيانات في الوقت الحالي الذي يجعل من الصعب معالجة وتحليل جميع المعلومات في الوقت المناسب. تنتشر المعلومات على وسائل التواصل الاجتماعي بسرعة فائقة مما يعني أن من الممكن أن تتطور الأحداث وتتغير بسرعة ، مما يجعل مواكبة الأحداث أمرًا صعبًا مع أحدث التطورات ، وقد يكون من الصعب التحقق من دقة المعلومات. هناك الكثير من الطرق التي يمكن استخدامها لتجميع المعلومات و التحقق من دقتها. الأساليب المستخدمة في جمع البيانات هي طرق تعتمد على الكلمات الرئيسية من خلال تحديد الكلمات الرئيسية ذات الصلة المهمة المتعلقة بحدث ما واستخدامها لتصنيف رسائل الوسائط ، كما تستخدم خوارزميات التعلم الآلي التي يمكن تدريبها لتحديد الأحداث وتصنيفها تلقائيًا استنادًا إلى الأنماط في بيانات وسائل التواصل الاجتماعي.

كانت هناك العديد من التطورات الأخيرة و الاكتشافات في مجال الكشف عن الأحداث. بعض النتائج الرئيسية وحقائق تم ملاحظتها حديثًا مثل تقنيات التعلم العميق مثل CNN و RNN و التي قد أظهرت نتائج واعدة في اكتشاف الأحداث على وسائل التواصل الاجتماعي ، وتحليل المشاعر ، و كذلك تتضمن تحديد وتصنيف المشاعر في الرسائل التي يمكن أن تحسن عملية الكشف عن الحدث. في النهاية ، يعد اكتشاف الحدث على وسائل التواصل الاجتماعي أمرًا صعبًا و كذلك المجال الديناميكي الذي يتطلب خوارزميات وتقنيات متطورة للتغلب على الضوضاء والغموض وحجم البيانات المرتبطة بمنصات التواصل الاجتماعي.

Table of Contents

Acknowledgements.....	ii
Abstract.....	iii
List of Figures.....	vii
List of Tables.....	viii
List of Abbreviations.....	ix
Chapter 1: Introduction.....	1
1.1 Problem Definition.....	2
1.2 Motivation.....	2
1.3 Objectives.....	2
1.4 Methodology	3
1.5 Time plan.....	4
1.6 Thesis Outline	6
Chapter 2: Literature Review	7
Chapter 3: System Architecture and Methods	12
3.1 System Architecture	13
3.2 Description of methods and procedures used	15
Chapter 4: System Implementation and Results	17
4.1 Dataset	18
4.2 Description of Software Tools Used	19
4.3 Step up Configuration (hardware).....	21
4.4 Experimental and Results	22
Chapter 5: Run the Application.....	32
Chapter 6: Conclusion and Future Work.....	47
6.1 Conclusion.....	48
6.2 Future Work.....	48

References.....49

List of Figures

Figure 1.5.1: Time Plane.....	5
Figure 3.1: System Architecture.....	14
Figure 4.1: Data Sample.....	18
Figure 4.4.1: NLP Detection of Event.....	24
Figure 4.4.5-1: Naïve Bayes Distribution of Categories.....	25
Figure 4.4.5-2: Random Forest Distribution of Categories.....	26
Figure 4.4.5-3: Logistic Regression Distribution of Categories.....	27
Figure 4.4.5-4: SVM Distribution of Categories.....	28
Figure 4.4: Visualization of Algorithm Accuracies.....	31
Figure 5.1: Splash Screen.....	33
Figure 5.2: Login Screen with valid data.....	34
Figure 5.3: Reset Password Screen.....	35
Figure 5.4: Sign Up Screen.....	36
Figure 5.5: Home Page.....	37
Figure 5.6: Search with invalid event.....	38
Figure 5.7: Search with valid event.....	39
Figure 5.8: Saved Events.....	40
Figure 5.9: Tweet's full content.....	41
Figure 5.10: Share Event.....	42
Figure 5.11: Manage profile.....	43
Figure 5.12: Change Name.....	44
Figure 5.13: Change Email.....	45
Figure 5.14: Change password.....	46

List of Tables

Table 2.1: Related Works.....11

Table 4.4.1: Comparison between used Algorithms.....29

Table 4.4.2: Comparison between Accuracies.....30

List of Abbreviations

API:	Application Programming Interface
BST:	Binary Search Tree
CNN:	Convolutional Neural Network
GB:	Gigabyte
GPS:	Global Positioning System
GRU:	Gated Recurrent Units
HD:	Hard Drive
LSTM:	Long Short-Term Memory
NLP:	Natural Language Processing
RAM:	Random Access Memory
RNN:	Recurrent Neural Network
SIM:	Subscriber Identity Module
SVM:	Support Vector Machine
UI:	User Interface
UX:	User Experience
Wi-Fi:	Wireless Fidelity

Chapter One

Introduction

1.1 Problem Definition

Now days people spend a lot of time to know the news that happens all over the world. There is vast amount of unstructured and noisy data that exists on different platforms. Social media users often use slang, non-standard language which makes it difficult to identify and extract meaningful information. Finding the news that's important to me is providing to be time-consuming task. It's taking a lot of effort to filter through all the noise.

1.2 Motivation

Social media platforms like Twitter have become the most important means of communication in recent years. It is observed that 9 out of 10 Twitter users use Twitter to post current happenings, news, and real-time events as Tweet. Automatic identification of events from social media might give more credible results than traditional news media, as users actively participate in contributing the information and generating content. Event detection from social media data is applicable to various government and industrial applications like emergency management during disasters, spread of contagious and infectious disease, campaigning events, instant outbreaks like floods, earthquake, and bomb blast; quickly spreading communicable diseases like swine flu and bird flu; public gatherings like conferences, ceremonies, fest; end-route traffic by detecting traffic congestion due to public events, and election campaigns and protests. All these applications act as motivation to carry this research work.

1.3 Objectives

We aim to automatically identify and categorize events mentioned on social media platforms. This application will gather all the events that happens all over the world to be easy to users to reach the events that he/she interested in.

1.4 Methodology

- Research design: The research we did involves an overview of the idea of the project and related works to this idea it was and this related works involves various types of research design such as Experimental, Correlation, and Diagnostic and we used Twitter social media platform to collect our data.
- Data collection: In data collection we used Twitter API to collect data through Twitter as a lot of people uses Twitter to publish the important events more than any other social media platform.
- Data analysis: Once the data has been preprocessed, the next step is to identify these data by applying Machine Learning algorithms. We used statistical method to classify and predict categorizes in our data which was logistic regression and natural language processing to detect events in the data.
- Ethical considerations: Event detection on social media raises several ethical considerations that we must consider such as Privacy, Informed Consent, Bias, Misinformation, Data Security, and Transparency.
- Limitation: Event detection on social media has several limitations that we must consider such as Data Quality, Data Quantity, Bias, Contextual Understanding, False Positives and False Negatives, and Privacy Concerns.

1.5 Time plan

1/10/2022 – 1/12/2022: Planning, Research, and Data Collection .

1/12/2022 – 1/1/2023: Research, and Data preprocessing.

1/1/2023 – 1/4/2023: Machine Learning, UX/UI designs, and Front-End.

1/3/2023 – 31/5/2023: Edits.

1/4/2023 – 30/6/2023: Back-End.

1/6/2023 – 1/7/2023: Final Version.

It's shown in *Figure 1.5-1* the visualization of this plan it shows the schedule of activities we worked on.

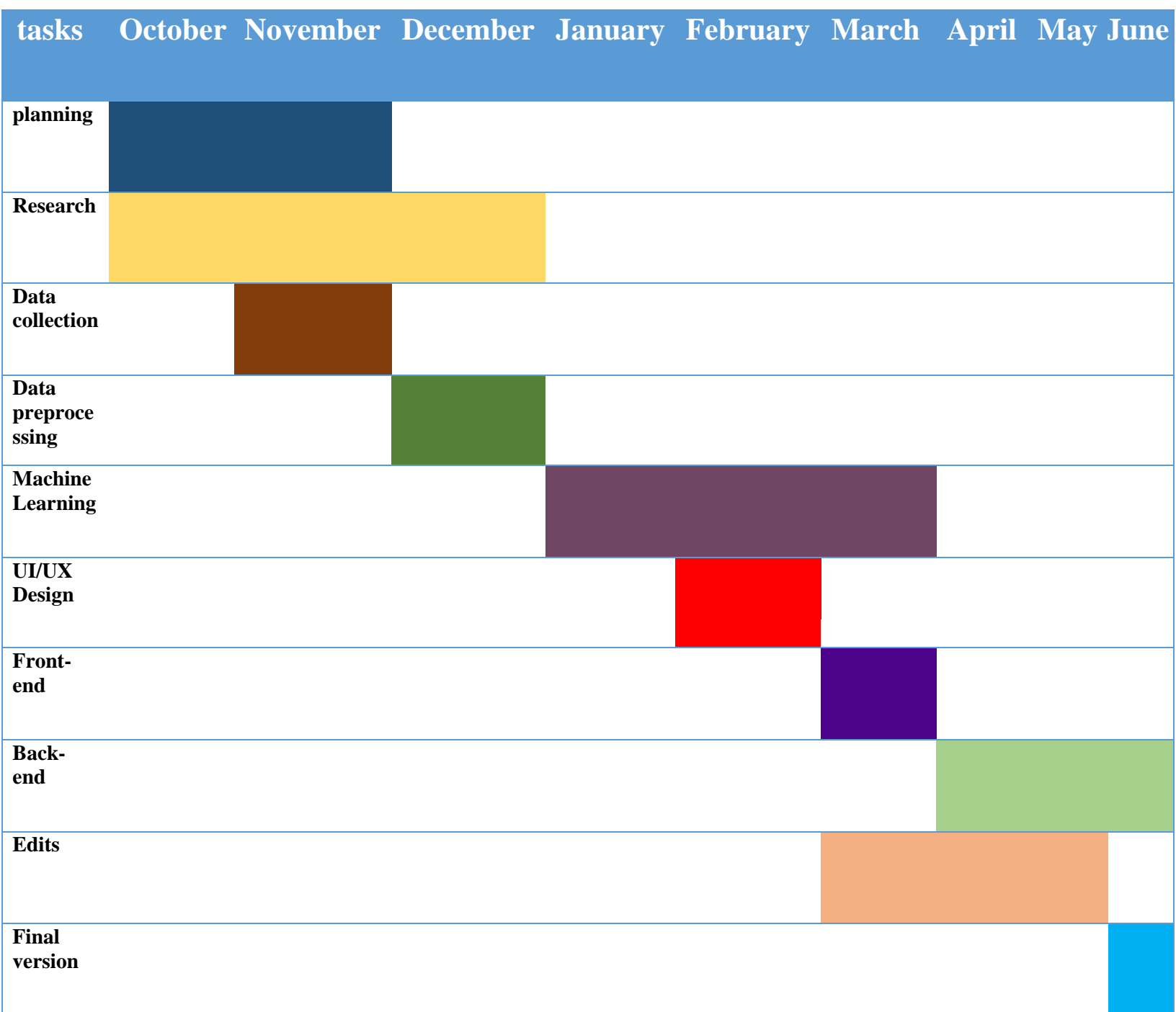


Figure1.5-1 Time Plan.

1.6 Thesis Outline

Chapter 1 “Introduction”

In this chapter, we talk about the definition of the problem, the reasons for choosing this project "Event Detection For Social Media" and our goal that we want to achieve.

Chapter 2 “Literature Review”:

In this chapter, we talk about general introduction, background, different approaches used from other papers and our comments on the survey we have done, and which approach we found the best.

Chapter 3 “System Architecture”

In this chapter, we talk about Data Acquisition, preprocessing and we explained all the general architectures that we used in our project in detail, and the general flow of each architecture.

Chapter 4 “System Implementation and Results”

In this chapter, we explained the best practices of the architectures that we used and also we specified the flow of our final model and it's hyper parameters.

Chapter 5 ” Run the Application”

User Manual It describes in detail how to operate the project along with screenshots of the project representing all steps.

Chapter 6 “Conclusions and Future Works”

In this chapter we state the conclusion and results that we got from our work and the work we plan to do in future.

Chapter Two

Background And Literature Review

Introduction

Event detection on social media has become increasingly important in recent years, as social media has become a key source of information for individuals and organizations around the world. Social media platforms such as Twitter, Facebook, and Instagram have become important channels for people to share news, opinions, and experiences. This wealth of information has created new opportunities for event detection, which can be used to identify and monitor events as they unfold.

The significance of event detection on social media lies in its potential to provide insights into a wide range of events, including natural disasters, political events, public health emergencies, and others. By detecting events on social media, researchers and practitioners can gain insights into the characteristics of the events, such as their location, timing, and severity. This information can be used to inform decision-making in a variety of domains, including crisis management, public health, and marketing analysis.

Background

Event detection on social media is a field of study that involves the identification and monitoring of events as they unfold on social media platforms such as Twitter, Facebook, and Instagram. There are basic principles and procedures of event detection on social media such as:

1. Data Collection: The first stage of event detection on social media is data collection. This involves collecting social media data from platforms such as Twitter, Facebook, or Instagram. The data can be collected using APIs provided by the platform or through web scraping techniques.
2. Preprocessing: Once the data is collected, it is preprocessed to remove noise and irrelevant information. This may involve techniques such as filtering out spam or removing duplicate posts.

Preprocessing can also involve normalizing the data to ensure that it is in a consistent format.

3. Feature Extraction: After preprocessing, features are extracted from the data using Natural language processing technique to represent the content of the posts. Features can include text-based features such as keywords, hashtags, and named entities, as well as metadata features such as the user who posted the content or the location of the post.
4. Classification: Once the features are extracted, the data is classified into different event categories. This can be done using machine learning algorithms such as support vector machines, decision trees, or neural networks. The classification model is trained on a labeled dataset of social media posts to learn patterns in the data and classify new posts.
5. Post-processing: After classification, post-processing techniques are applied to refine the results and improve the accuracy of the event detection. This may involve techniques such as clustering similar events or using temporal analysis to ensure that events are not detected multiple times.
6. Visualization: Finally, the results of event detection are visualized to provide insights into the events and their characteristics. This can include visualizations such as maps, timelines, or graphs to show the spread and evolution of events over time.

The procedures of event detection on social media can vary depending on the specific research question or application. For example, some researchers may focus on detecting events related to natural disasters, while others may focus on detecting events related to political events or public health emergencies. Additionally, the specific techniques and algorithms used in each stage may vary depending on the specific research question or application.

The previous studies and works

There are a lot of previous studies and works that discuss event detection on social media from these works and studies:

1- A tutorial on event detection using social media data analysis:

Numerous analyses in data science have been conducted for extracting information from social media. This paper synthesizes the findings of many of those projects to discuss current applications, challenges, and open problems regarding the analysis of social media data. And the main goal in creating this article is to bring together a number of unique strategies that, when correctly coupled, should significantly improve event detection using social media data analysis. They collected data using Twitter social stream and the algorithms used in this paper is CNN, LSTM, and GRU. And achieved accuracy was 88.27%.

2- Road traffic event detection using twitter data:

The increasing number of vehicles, social events, lane closures, roadworks, adverse weather, and other unexpected incidents have a negative impact on traffic flow and cause traffic congestions. Therefore, those causes, namely events (incidents), should be detected in an efficient and timely manner in order to support decision making and set management strategies to reduce or eliminate congestion. They collected data used in this paper is from Twitter and the algorithms used in this paper is Naïve Bayes, SVM, and Logistic Regression. And achieved accuracy was 90.

3- Event detection in Twitter: A machine-learning approach based on term pivoting:

Twitter gives users a voice to share ideas, opinions, and experiences with friends and the general public. Owing to the large user base on Twitter, the platform provides real-time information about what happens in the world. Detecting events and harvesting references to them from Twitter is therefore a highly valuable goal. The data used in this paper was collected from Twitter API based on the list of Dutch words. The algorithms used in this paper was Clustering, Classification, Document frequency, SIM, and BST. And achieved accuracy was 86%.

Citation	Dataset	Algorithm	Accuracy
A tutorial on event detection using social media data analysis [2022].	Twitter social stream.	<ul style="list-style-type: none">• CNN• LSTM• GRU	88.27%
Road traffic event detection using twitter data [2019].	Twitter data.	<ul style="list-style-type: none">• Naïve Bayes• SVM• Logistic Regression	90%
Event detection in Twitter: A machine-learning approach based on term pivoting [2014].	Twitter API on the basis of a seed list of Dutch words and a list of the most active Dutch users.	<ul style="list-style-type: none">• Clustering• Classification• Document frequency• SIM• BST	86%

Table 2-1. Related Works.

Chapter Three

System Architecture and Methods

3.1 System architecture

Figure 3-1:

Shows that System Architecture we have 3 layers. First layer is Presentation Layer, Presentation Layer is the user interface and communication layer of the application, where the end user interacts with the application. Its main purpose is to display information to and collect information from the user so we put in it that we use flutter application as an interface for the user. Second layer is Application Layer, in this layer we apply data preprocessing to dataset we collected to remove noise, unwanted data, stop words, lemmatization and stemming, tokenization, and data cleaning. After the preprocessing the data is ready for feature extraction according to these features the model will consider if this tweet is an event or not. Then the model is build to detect the events in the data according to the features by training and fitting the model on the data. The model is evaluated to predict the data to ensure that it is predicting the right events. The last layer is database layer, Database layer is where we get the data using Twitter platform through API.

**PRESENTATION
LAYER**



MOBILE INTERFACE

**APPLICATION
LAYER**

Data Preprocessing

- Tokenization
- Stop words removal
- Stemming and lemmatization
- Data cleaning

Feature Extraction

Model Building

Model Evaluation

**DATA BASE
LAYER**



API



TWEETS

File that contains clean, detected and categorized data to the flutter app.

File that conations the tweets from Twitter API.

Figure 3-1.System Architecture

3.2 Description of methods and procedures used

Model:

- The NLP Algorithm detects the events in the data sets.
- The Machine Learning Algorithm SVM classifies the events to categories.

System:

- Login:

The user could login to system if he has registered before using his email and password.

- Register:

If it's the users first time visiting the application the user must register to the system first by entering his name, email, and password.

- Saved:

The user could save any tweet he wants in his saved list and could visit it any time.

- Manage profile:

The user has the ability to manage his profile by changing name, password, and username.

- Filter category:

The user can navigate between categories by choosing the category he wants to see it's events.

- Forget password:

The user could recover and change his password if he forgot it by entering the email the he registered by and an email will be sent to this account giving him the ability to type in new password.

- Like tweet:

The user could like a tweet.

- Search a tweet:

The user could search a tweet by typing it's name in the search box and I will appear to him.

- Open a tweet:

By clicking on the tweet in the home page the tweet will be opened in another page so the user will be able to see it's full content.

- Share a tweet:

The user could share a tweet by clicking on share and the tweet content will be copied.

- Sign Out:

The user could sign out of the application.

- The system has the ability to detect if the user has registered before or not by checking the database and if not it wouldn't allow his login without registration.
- In registration the system will be able to detect the mismatch between the password and the confirm password.
- In login if the user has checked the remember me checkbox or not and if it's checked the system will save his login data to make it easier in login the other time.
- The system could detect if any of the textboxes is empty in login and register page to not allow the missing of user data in login and registration.

Chapter Four

System Implementation and Results

4.1 Dataset:

4.1.1 Data Acquisition:

Our dataset contains 35,000 tweets that have been labeled with categories. We have four categories which are sports, politics, business, finance. The tweets were collected using the Twitter API by randomly sampling tweets that contain certain keywords. Fields are:

- Date: represents the date and time when each tweet was posted on Twitter.
- User: represents the Twitter handle or username of the user who posted each tweet.
- Category: represents the category to which each tweet belongs.
- Tweet: represents the text content of each tweet.

Figure 4.1-1 displays a data sample of our dataset which are:

Column1	Date	User	Category	Tweet
0	0 2023-02-01 01:53:01	https://twitter.com/2icyboi	Sports	NHL Picks For The Night:\n\n(Singles not parla...
1	1 2023-02-01 01:52:08	https://twitter.com/SinCitySpreads	Sports	NBA FREE PLAY: \n\nCleveland Cavaliers -2\n\nU...
2	2 2023-02-01 01:47:32	https://twitter.com/LeEpicTroll	Sports	Jake paul is an extraordinary boxer.\n\nJake p...
3	3 2023-02-01 01:40:52	https://twitter.com/Man_Cave_Picks	Sports	ðŸ\ue0a0ENCAA Pick\nUNLV Vs CO State 6PM PST\nðŸ\ue0a0P...
4	4 2023-02-01 01:39:26	https://twitter.com/Man_Cave_Picks	Sports	ðŸ\ue0a0ENCAA Pick\nTexas A&M Vs Arkansas 4PM P...
5	5 2023-02-01 01:37:58	https://twitter.com/BetsATM4	Sports	ðŸŒˆ Tuesday - 01/31/23 ðŸŒˆ\n\nA: Kansas Stat...
6	6 2023-02-01 01:37:16	https://twitter.com/thecointhieves	Sports	CASHED FREE PLAY YESTERDAYæœœ...\n\nGIVEN Yâ€™™ALL...
7	7 2023-02-01 01:37:12	https://twitter.com/Man_Cave_Picks	Sports	ðŸ\ue0a0ENCAA Pick\nVCU Vs Davidson 4PM PST\nðŸ\ue0a0Pi...
8	8 2023-02-01 01:35:11	https://twitter.com/Man_Cave_Picks	Sports	ðŸ\ue0a0ENCAA Pick\nWest Virginia Vs TCU 6PM PST\nðŸ...
9	9 2023-02-01 01:31:03	https://twitter.com/classcitylocks	Sports	1/31/23 NBA Prop Picks Pt 2\n\nNikola Jokic o2...
10	10 2023-02-01 01:30:00	https://twitter.com/PlaybookFB	Sports	#SuperTuesday #ESPN\n\n#PBXperts Drive The \$Lane...

Figure 4.1-1. Data Sample.

4.1.2 Data Preprocessing:

This data went through data preprocessing to remove noise, unwanted data, stop words, lemmatization and stemming, tokenization, and data cleaning. After data preprocessing data become 26886 Tweets.

4.2 Description of Software Tools Used:

4.2.1. Python (Python Software Foundation, Version 3.10.12, 2023): We used Python to write the code for data preprocessing, analysis and visualization. Python is widely used as it is ease of use and has many libraries available for data analysis and machine learning.

4.2.2. Colab (Google Collaboratory, latest release at the time of development): We used Google Colab for developing and running machine learning models. It is free and allow users to write and run Python code in Jupyter Notebook environment.

4.2.3. Pandas (Pandas Development Team, Version 1.5.3, 2023): We used Pandas library in Python to manipulate and analyze data in tabular form. It provide powerful tools for working with structured data including data cleaning, filtering, and aggregation.

4.2.4. Scikit-learn (Scikit-learn developers, Version 1.2.2, 2023): We used Scikit-learn library in Python for machine learning tasks as classification and regression. It provide tools for machine learning including data preprocessing, model selection and evalution.

4.2.5. NumPy (Version 1.22.4, 2022): We used Numpy library in Python as it efficiency, ease of use and versatility in handling large dataset. It provide set of tools for working with array and matrices including random number generation, linear algebra and mathematical operations.

4.2.6. Sastrawi (Version 1.0.1, 2017): We used Sastrawi library in Python to create a new instance of the stemmer, which can then be used to stem words in Indonesian text. It provide tools and options for customizing the stemming process to suit a variety of applications and use cases.

4.2.7. NLTK (Natural Language Toolkit, Version 3.8.1,2023): It is a popular open-source library for NLP in Python. se from NLTKWe u library "TweetTokenizer" class as itis specialized tokenizer for processing Twitter data, which can contain a variety of unique features such as hashtags, mentions, and emoticons.

4.2.8. Matplotlib (Version 3.7.1, 2022): We used Matplotlib library in Python to create a wide variety of plots and visualizations.

4.2.9. spaCy (Version 3.5.3, 2023): It is a popular open-source library for NLP in Python. It provides a range of tools and capabilities for working with text data, including tokenization, part-of-speech tagging, named entity recognition, dependency parsing. It used in research and industry for a variety of natural language processing applications.

4.2.10. Flutter (Version 3.7.5, 2023): We used Flutter to develop high quality mobile application efficiency and effectively. It also provide us tools to create responsive user interface. The development environment used for Flutter was Android Studio.

4.3 Step Configuration (Hardware):

4.3.1. Processor: A high-performance processor is essential for running complex machine learning algorithms. A minimum requirement could be a quad-core processor with a clock speed of 2.0 GHz or higher.

4.3.2. RAM: The amount of RAM is also important for performance and multitasking. A minimum of 4 GB of RAM would be recommended for running the app smoothly.

4.3.3. Storage: The app will likely require storage space for the data and machine learning models. A minimum of 32 GB of internal storage should be sufficient for most use cases. Additionally, expandable storage options through microSD card slots would be useful.

4.3.4. Display: The app should have a high-resolution display to provide an immersive and user-friendly experience. A Full HD (1080p) or higher resolution display with a minimum size of 5.5 inches would be recommended.

4.3.5. Battery: Running machine learning algorithms can be resource-intensive, so a high-capacity battery is essential. A minimum of 3,000 mAh battery capacity would be recommended.

4.3.6. Sensors: The app may require various sensors such as GPS, accelerometer, gyroscope, and compass for detecting events and providing location-based information.

4.3.7. Connectivity: The app will need to connect to the internet to collect data from social media platforms and to provide real-time updates. The device should have Wi-Fi, Bluetooth, and cellular connectivity options.

4.4 Experimental and Results:

After research it was concluded that the best algorithms in text classification is Support Vector Machine, Naïve Bayes, and Logistic Regression. And we used Natural Language Processing to detect events in the data set.

- We collected the data using Twitter platform through API.
- Then this data went through data preprocessing to remove noise, unwanted data, stop words, lemmatization and stemming, tokenization, and data cleaning.
- After the preprocessing the data is ready for feature extraction according to these features the model will consider if this tweet is an event or not by using NLP Algorithm.
- Then the NLP model is build to detect the events in the data according to the features by training and fitting the model on these features.
- Then these events are classified according to their categories using machine learning algorithms.
- After the training and fitting is done on the data the model is ready to evaluate and predict new data to ensure that it is predicting the right events according to their categories.

4.4.1. NLP Algorithm:

NLP is a subfield of artificial intelligence that focuses on the interaction between computers and human language. In event detection on social media, NLP techniques are used to analyze the text content of social media posts and identify events as they unfold. NLP techniques can be used in several stages of the event detection process. In the feature extraction stage, NLP techniques is used to extract features from the text data that are relevant for event detection. These features can include keywords, named entities, and sentiment analysis, among others. For example, named entity recognition can be used to identify named entities such as people, locations, and organizations mentioned in the social media posts, which can be used to identify events related to those entities. And after applying the NLP on our data set it detects events by accuracy 85% and detects non-events by accuracy 93%.

The visualization of NLP:

Figure 4.4-1 differentiates between tweets that describe events and those that do not:

Where event tweets are 10.7% and non-event tweets are 83.3%:

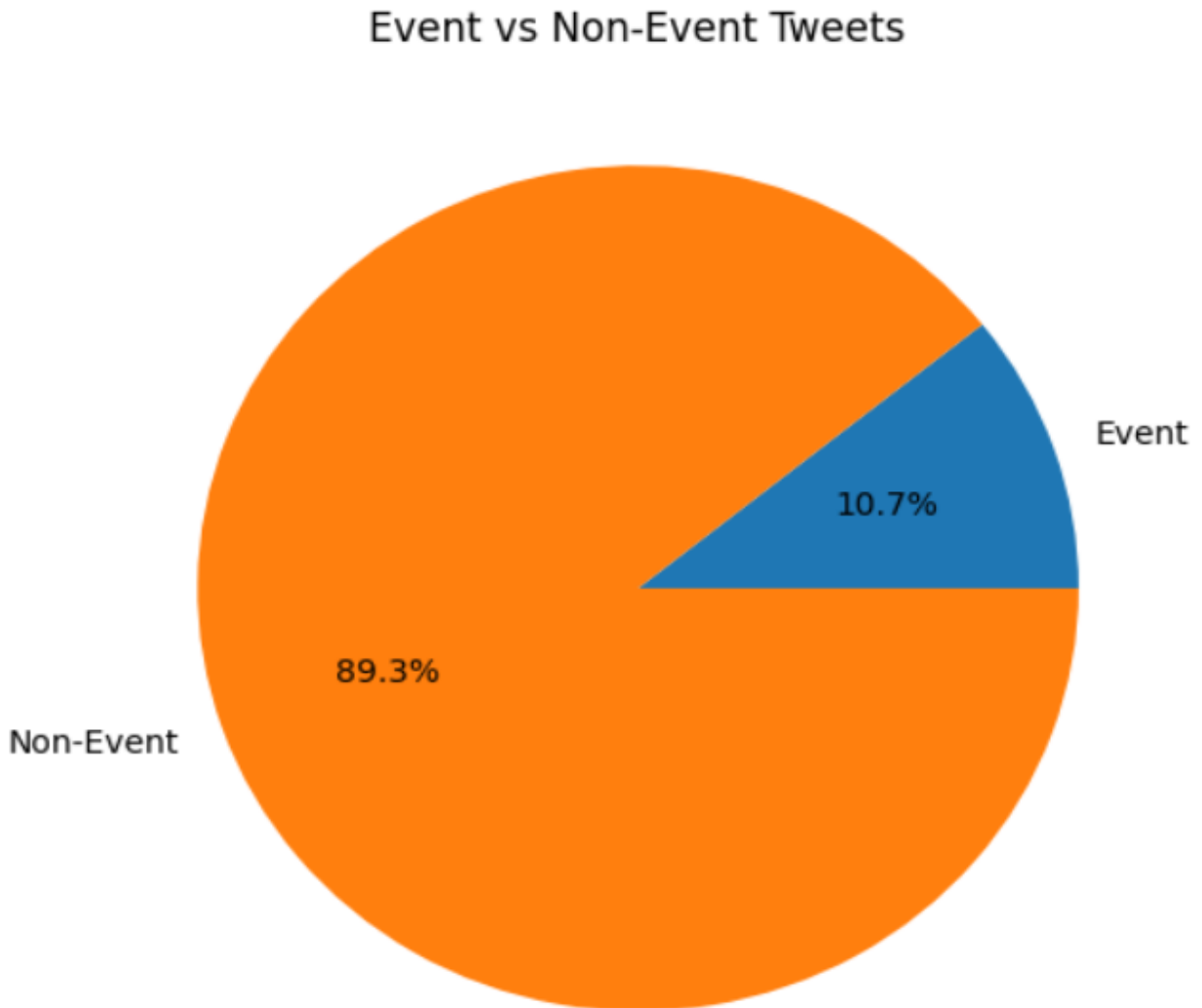


Figure 4.4-1. NLP Detection of Events.

4.4.2. Naïve Bayes:

Naive Bayes is a probabilistic machine learning algorithm that is commonly used for text classification, which is the task of assigning predefined categories or labels to a given text document. It is called "Naive" because it makes a strong assumption that each feature or word in the document is conditionally independent of all the other features, given the class label. In text classification using Naive Bayes, each document is represented as a bag-of-words, which is a vector of word frequencies. The algorithm then calculates the probability of each class given the document's word frequencies using Bayes' theorem, which states that the probability of a hypothesis in “the class label” given some observed evidence “the document's word frequencies” is proportional to the probability of the evidence given the hypothesis multiplied by the prior probability of the hypothesis. The Naive Bayes algorithm assumes that the probability of each word in the document given the class label is independent of all other words, which is a simplifying assumption that allows the algorithm to be trained efficiently on large datasets. This assumption is often violated in practice, but Naive Bayes can still perform well in many text classification tasks.

After applying Naïve Bayes to classify text after NLP we achieved accuracy equals to **83.22%**.

Figure 4.4-2: shows the Percentage of classified tweets in each category.

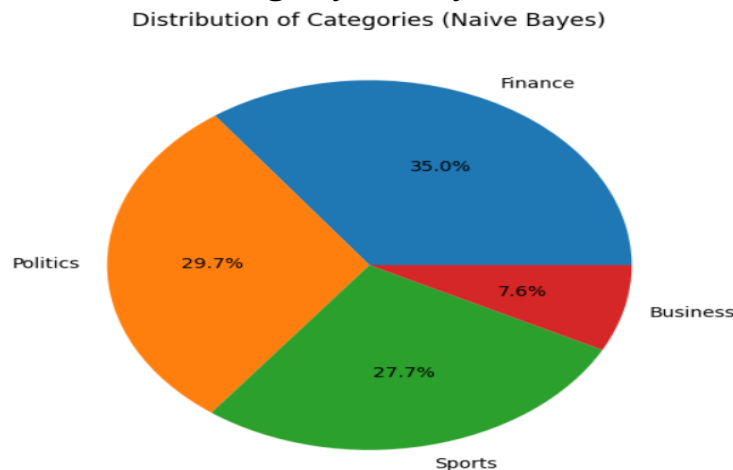


Figure4.4-2.NaïveBayesDistribution of Categories.

And to improve this accuracy we applied Random Forest Classifier.

4.4.3. Random Forest Classifier:

Random Forest Classifier is a popular machine learning algorithm used for classification tasks. It is an ensemble learning method that combines multiple decision trees to improve the accuracy and generalization performance of the model. It's idea is to create a forest of decision trees, where each tree is trained on a random subset of the training data, and a random subset of features is used to make decisions at each node of the tree. By using random subsets of the data and features, the algorithm reduces the risk of overfitting and improves the diversity of the trees in the forest. It's used in various applications such as image classification, text classification, and bioinformatics. It is known for its high accuracy, robustness, and ability to handle large datasets with high dimensionality.

After applying Random Forest Classifier to classify text after NLP we achieved accuracy equals to **84.28%**.

Figure 4.4-3: shows the Percentage of classified tweets in each category.

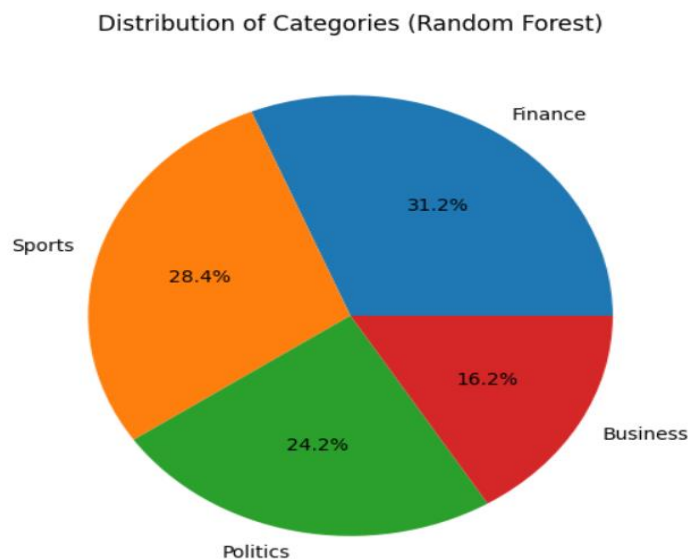


Figure 4.4-3. Random Forest Distribution of Categories.

And for better improvement we applied Logistic Regression.

4.4.4. Logistic Regression:

Logistic regression is a classical statistical machine learning algorithm that is commonly used for binary and multi-class text classification. In text classification, the goal is to assign predefined categories or labels to a given text document based on its content. In logistic regression, the algorithm models the relationship between the input features “words or n-grams in the text document” and the probability of each class using a logistic function. The logistic function maps the input features to a continuous output value between 0 and 1, which can be interpreted as the probability of the document belonging to a particular class. The algorithm learns the weights of the input features that maximize the likelihood of the observed training data. Logistic regression can handle both linear and non-linear relationships between the input features and the class probabilities, which makes it a versatile algorithm for text classification. It can also handle high-dimensional feature spaces, which is often the case in text data. Additionally, logistic regression is a simple and interpretable algorithm that can provide insights into the importance of different input features for classification.

After applying the logistic regression after NLP we achieved accuracy equals to approximately **85.55%**.

Figure 4.4-4: shows the Percentage of classified tweets in each category.

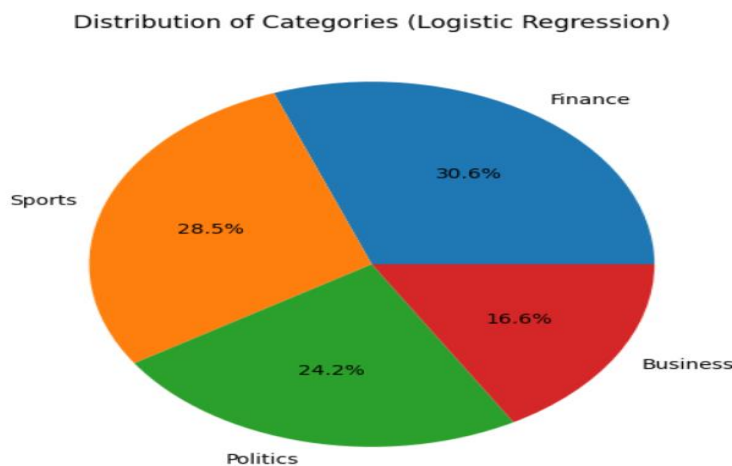


Figure 4.4-4. Logistic Regression Distribution of Categories.

And for better improvement we applied SVM.

4.4.5. SVM:

SVM is a machine learning algorithm that can be used for text classification, which is the task of assigning predefined categories or labels to a given text document. SVM is a popular algorithm for text classification due to its ability to handle high-dimensional feature spaces, which is often the case in text data. The basic idea behind SVM is to find a hyperplane that separates the data into different classes such that the margin between the hyperplane and the closest data points of each class is maximized. In text classification, each text document is represented as a vector of features, where each feature represents a term or a word in the document. The SVM algorithm then learns a decision boundary based on these features that separates the documents into different categories.

After applying SVM to classify text after NLP we achieved accuracy equals to **90.27%**.

Figure 4.4-5: shows the Percentage of classified tweets in each category.

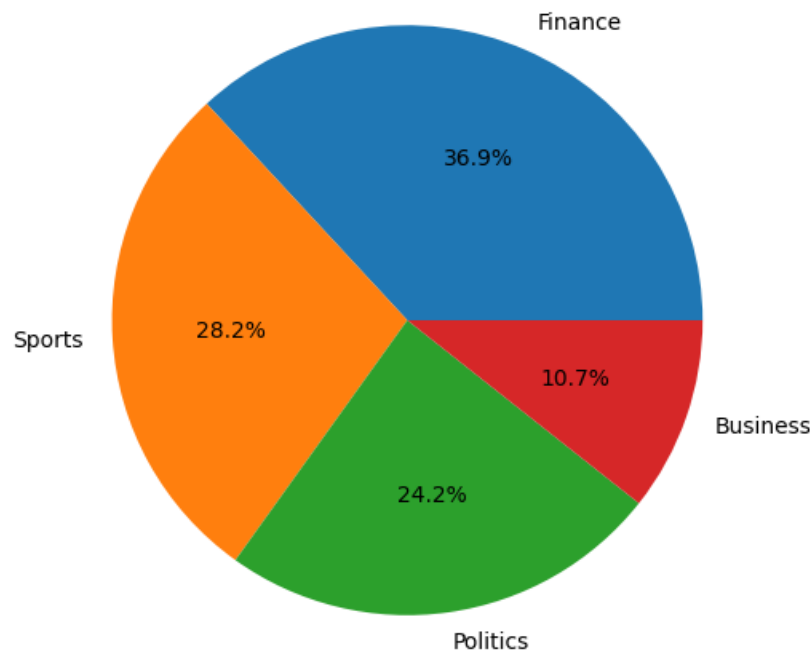


Figure 4.4-5. SVM Distribution of Categories.

Comparison between Algorithms According to their advantages:

Naïve Bayes	SVM	Logistic Regression	Random Forest Classifier
Simplicity.	Effective in high-dimensional spaces.	Simple and interpretable.	Ease to use and does not require much tuning of hyper parameters.
Fast training and prediction.	Robust to noise.	Fast training and prediction.	High accuracy due to the ensemble of multiple decision trees.
Robustness to irrelevant features.	Can handle non-linear data.	Robust to irrelevant features.	Robust to noisy data and outliers.
Efficient with high-dimensional data.	Good generalization performance.	Can handle non-linear relationships.	Can handle high-dimensional data with many features.
Performs well in practice.	Flexibility in choosing kernel functions.	Can handle large datasets.	
Online learning.	Can handle large datasets.	Good performance with small and moderated sized datasets.	

Table 4.4-1. Comparison between used Algorithms.

Comparison between Accuracy:

Algorithm	Accuracy
Naïve Bayes	83.22%
Random Forest Classifier	84.28%
Logistic Regression	85.55%
SVM	90.27%

Tabel 4.4.2 Comparison between accuracies.

Figure 4.4 illustrates the accuracy of each algorithm we used. As shown in the figure, (SVM) algorithm achieved the highest accuracy of 90%, indicating its effectiveness in our dataset.

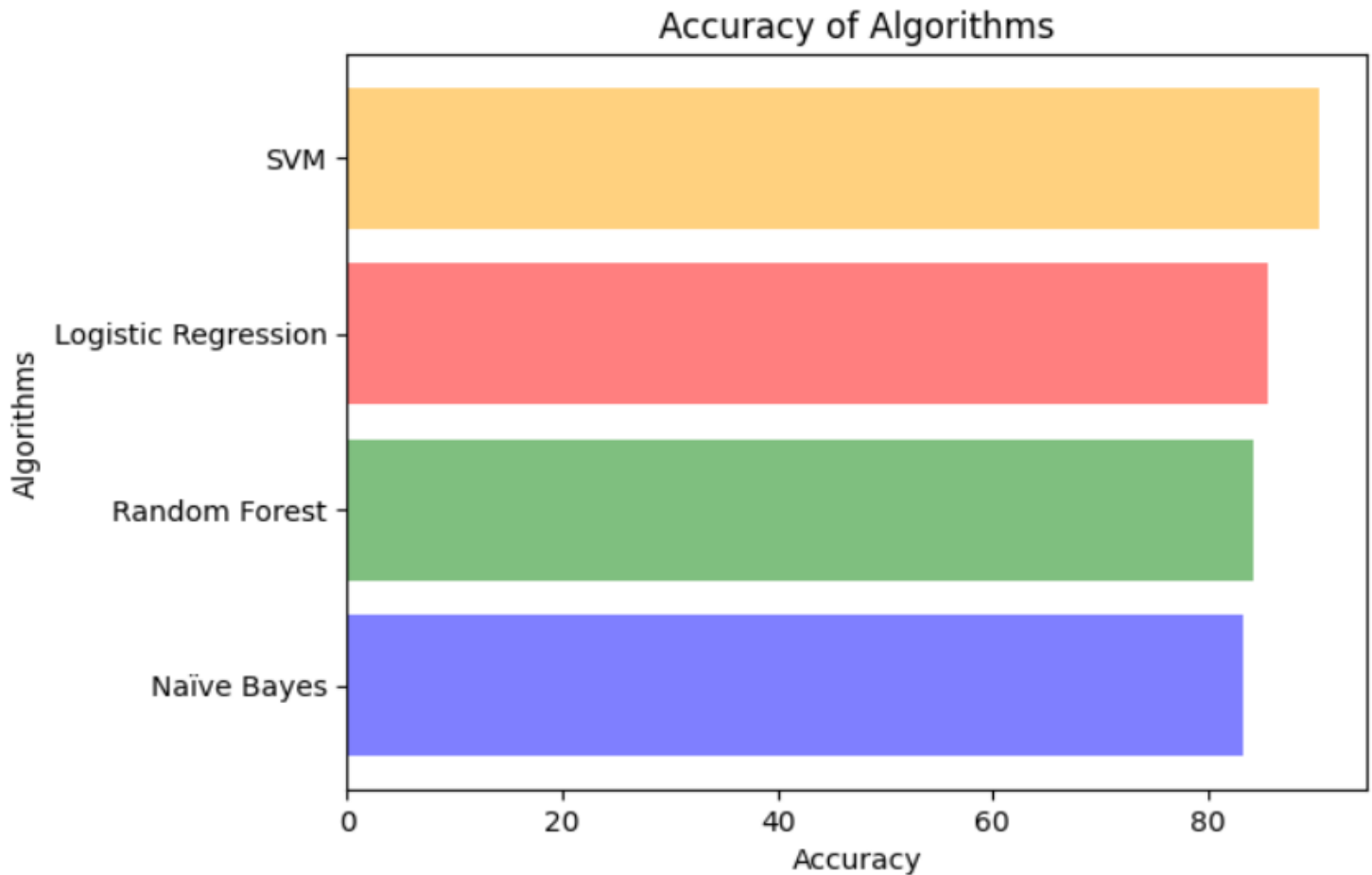


Figure 4.4. Visualization of Algorithms accuracies.

Chapter Five

Run the Application

- ❑ Firstly when user opens the app a splash screen that contains the app name will appears.



Figure 5-1. Splash Screen.

- ❑ Secondly, the login page user could login with a valid email and password.

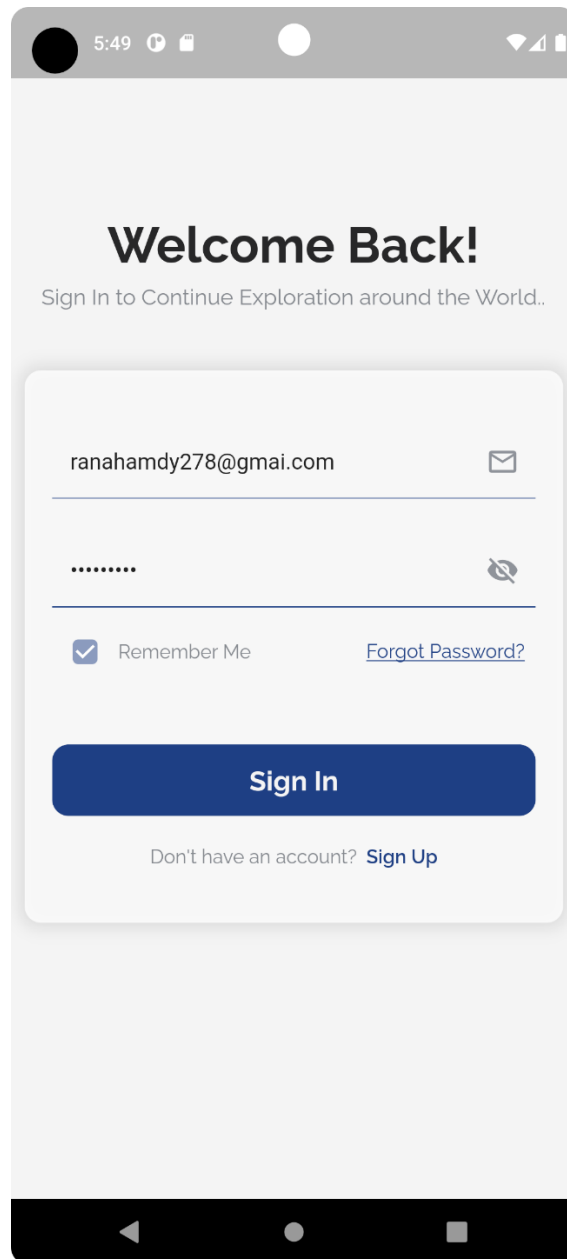


Figure 5-2. Login Screen with valid data.

- ❑ User could reset his password with entering his email.

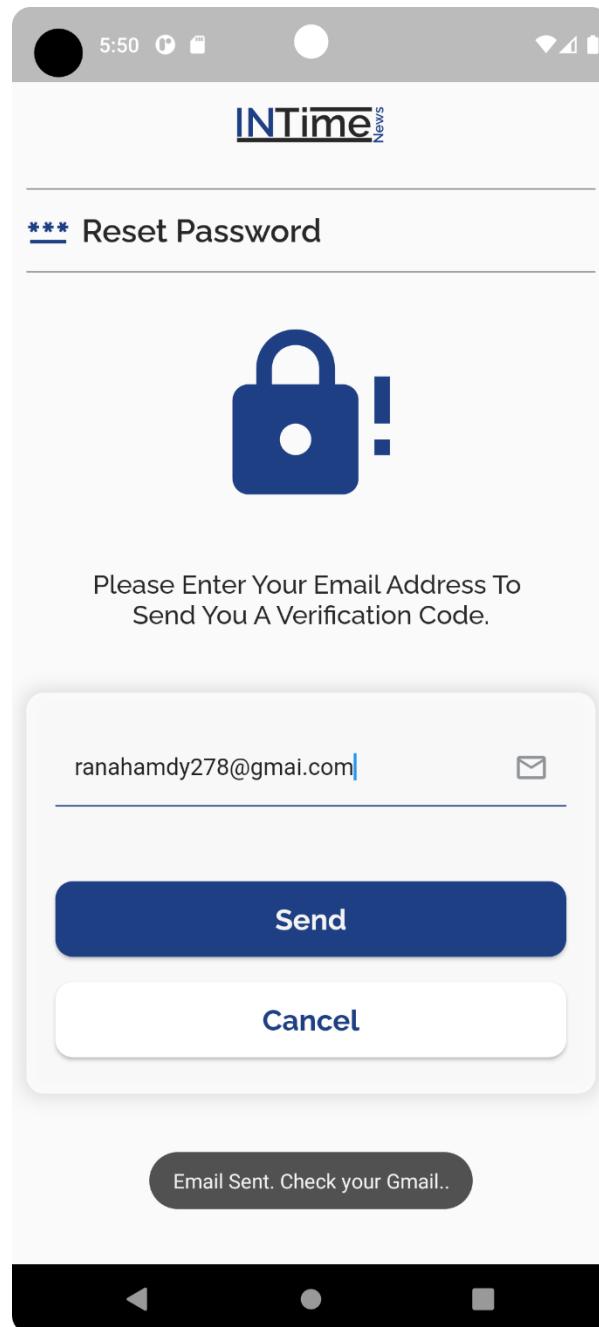



Figure 5-3. Reset password Screen.

- ❑ If this is the user's first time in the application he will have to sign up with entering valid data in all the fields and the password must match the confirm password.


5:53


Sign Up


Sign Up to enjoy new experience exploring news around the World.

Enter full name 

This Field can't be empty

ranahamdy278@gmail.com 

..... 

11111 

Password Mismatch

☒ I agree to the [Terms & Conditions](#)

Sign Up

Already have an account? [Sign In](#)

Figure 5-4. Sign Up Screen.

- ❑ In the home page it appears events randomly in all categories and if the user wants a specific category he could choose it and the events will be filtered.

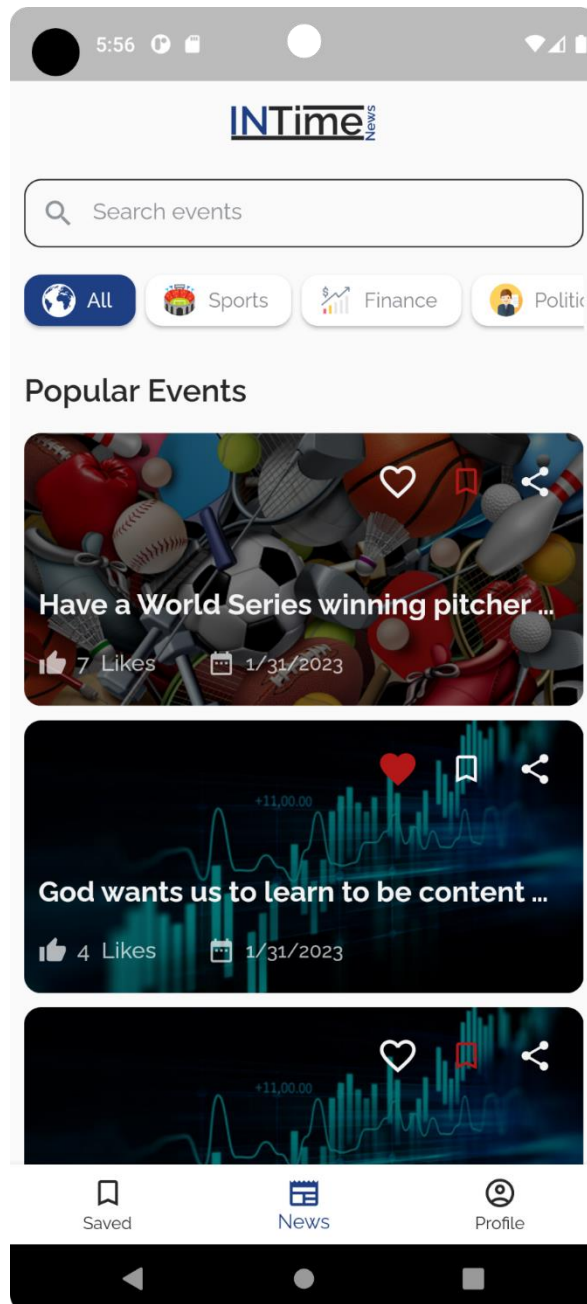


Figure 5-5.Home page.

- ❑ User could search an event by entering a vaild event name.

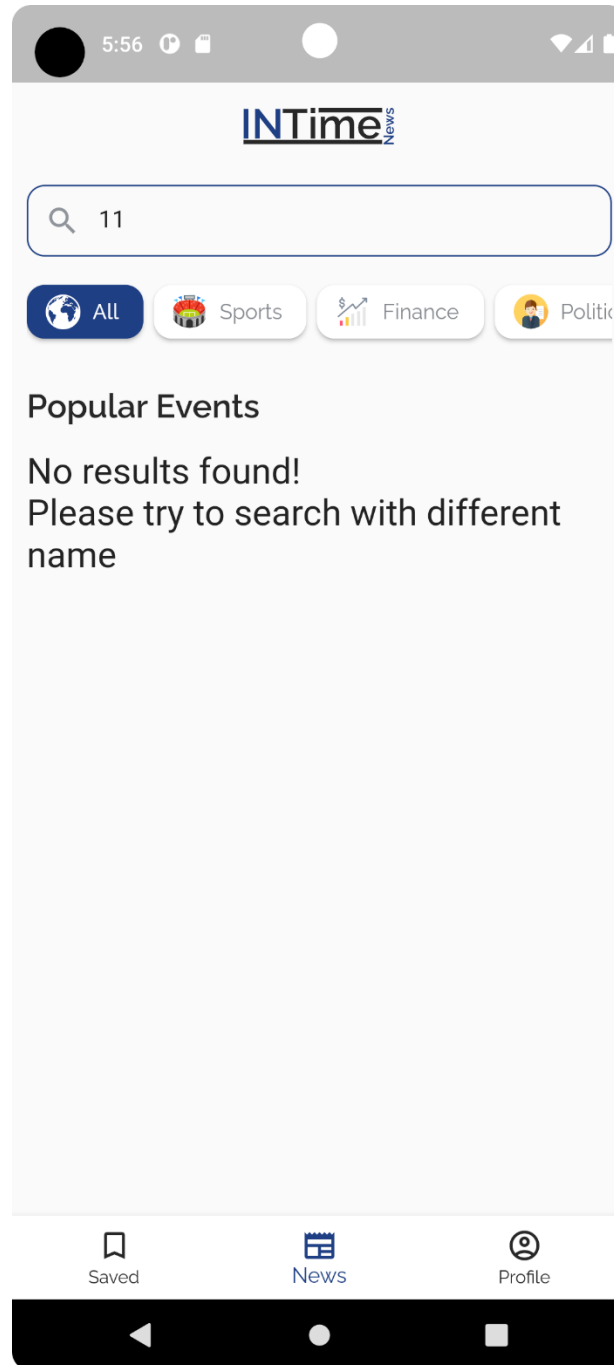


Figure 5-6. Search with invalid event.

❑ Search with valid event name.

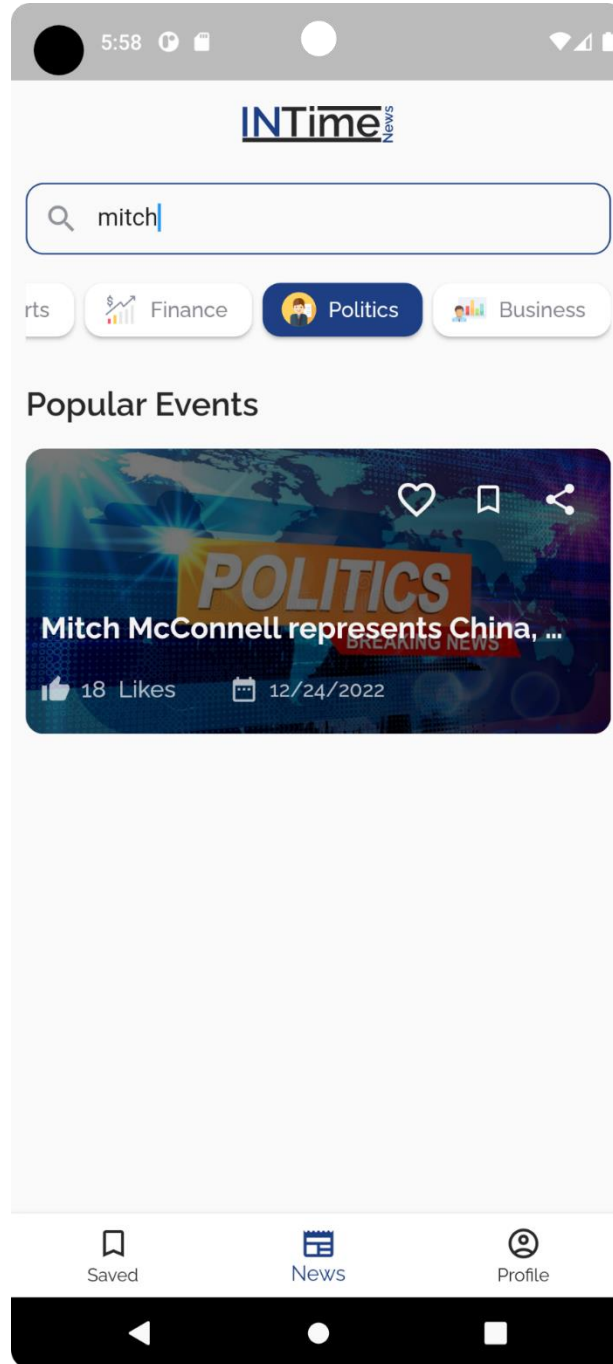


Figure 5-7. Search with valid event.

❑ User could see his save list.

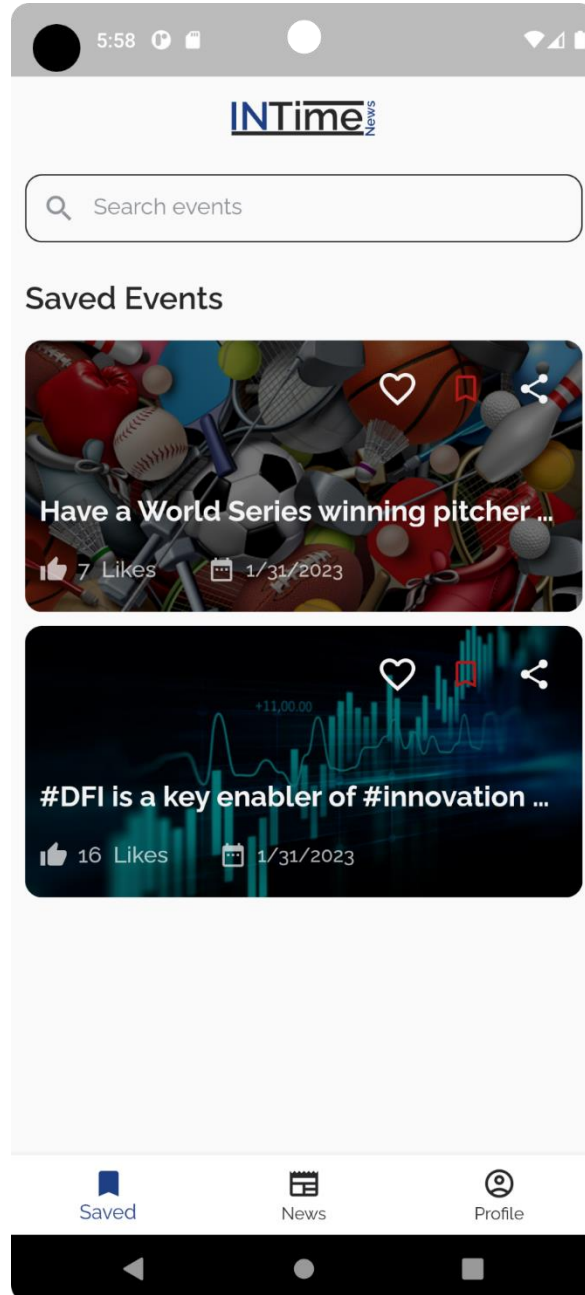


Figure 5-8. Saved Events.

- ❑ User could see the tweet whole content by clicking on it the app will transfer him to another page with the full content of the tweet.



Figure 5-9. Tweet's full content.

- ❑ User could share the tweet content by clicking of share to copy it.

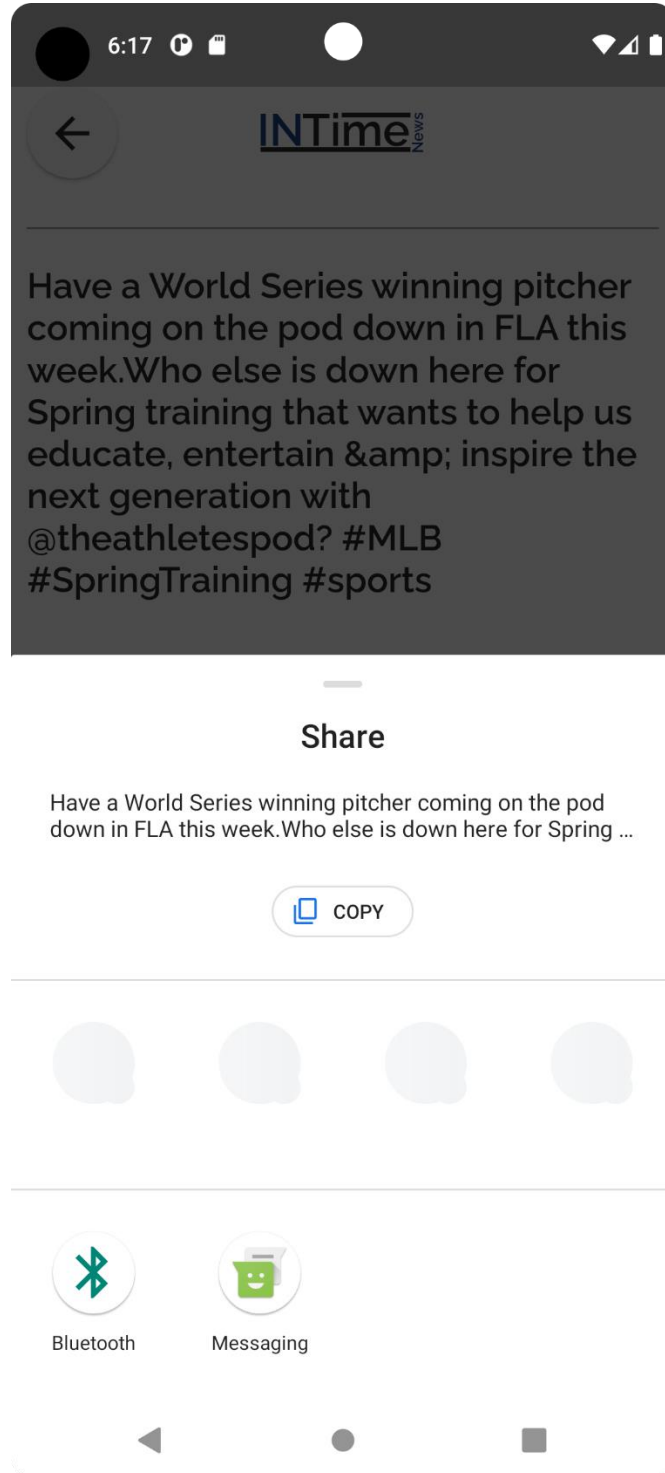


Figure 5-10. Share Event.

- ❑ User could manage his profile by changing name, email, and password.

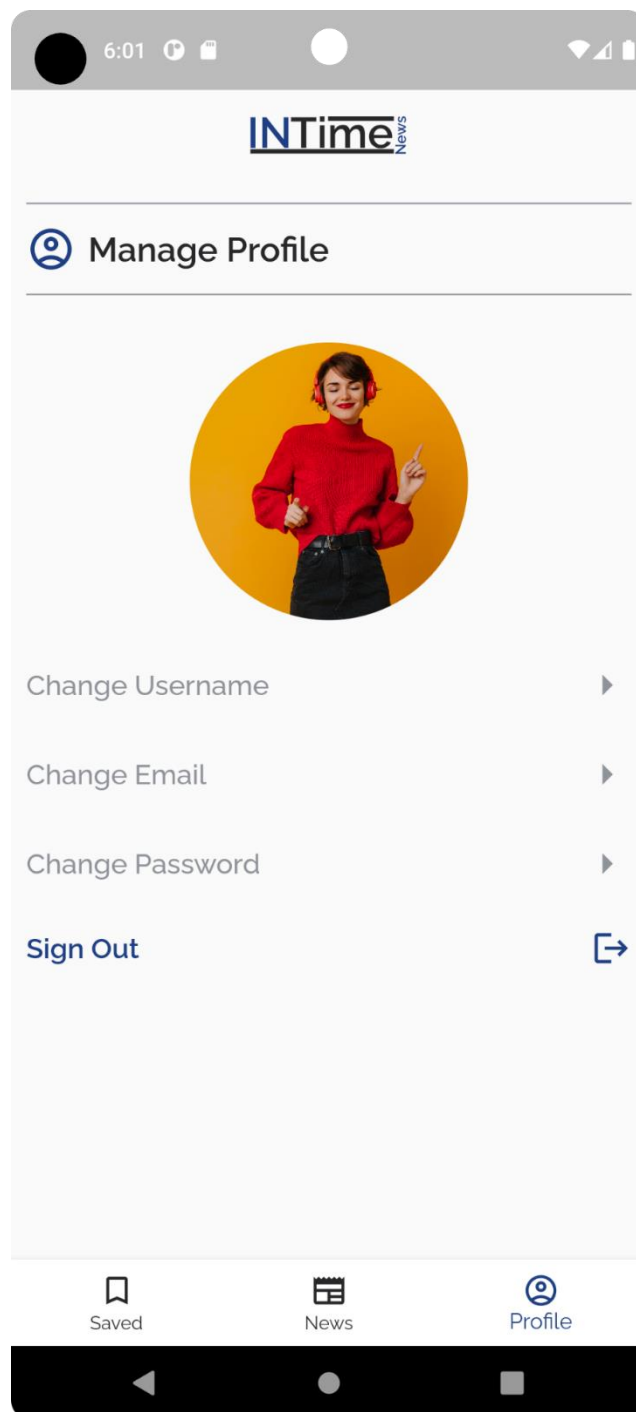


Figure 5-11. Manage Profile.

❑ Entering new name.

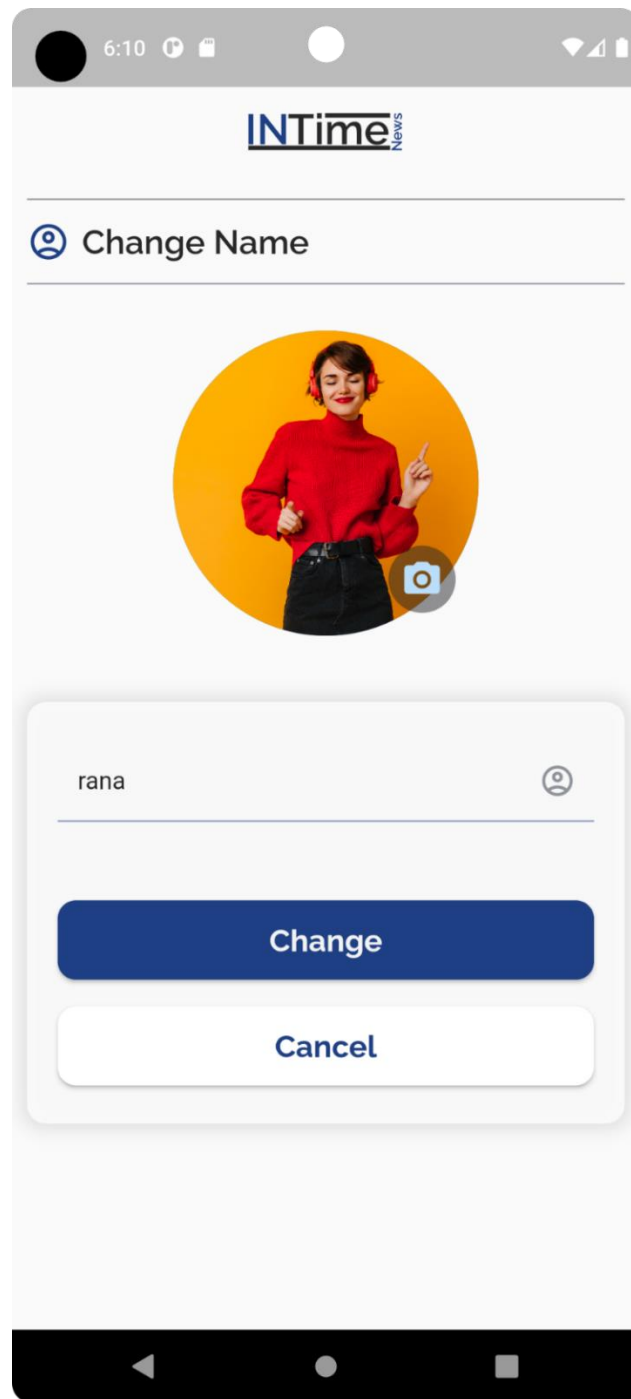


Figure5-12. Change Name.

❑ Entering new email.

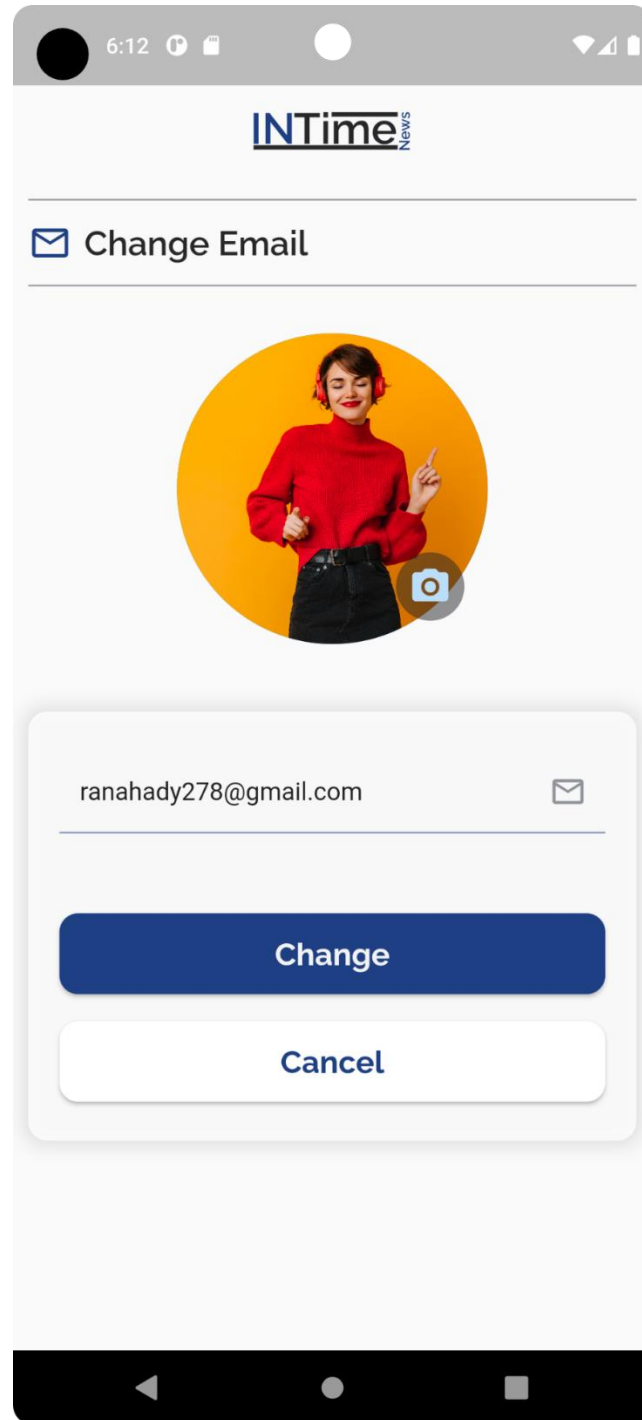


Figure5-13. Change Email.

❑ Entering new password.

6:14

INTime News

*** Change Password

Enter New Password

This Field can't be empty

Confirm Password

Please entre your password

Change

Cancel

Figure 5-14. Change Password.

Chapter Six

Conclusion and Future Work

6.1 Conclusion

Event detection on social media is an important area that has the potential to provide information about events as they happen. The use of machine learning algorithms has made it possible to automate this process, but there are still many challenges that need to be addressed. These challenges include dealing with noisy and unstructured data, identifying relevant features and patterns, and handling the dynamic and evolving nature of social media content.

SVM is an effective machine learning algorithm for event detection on social media. The fact that it achieved an accuracy of 90% suggests that it is capable of accurately identifying events based on patterns and trends in social media data.

6.2 Future Work

We will focus on improving the accuracy and scalability of the logistic regression model. Additionally, other machine learning algorithms could be explored and compared to logistic regression to determine the most effective approach for event detection on social media. Also we can use different data source as images and videos. As social media platforms are increasingly used to share multimedia content, and incorporating this data into event detection algorithms could improve accuracy and provide more comprehensive insights into events.

REFERENCES

1. Beccari, B, A Comparative Analysis of Disaster Risk, Vulnerability and Resilience Composite Indicators. PLoS Currents (2016).
2. Boettcher, A. and Lee, D. Eventradar: A real-time local event detection scheme using twitter stream. In Green Computing and Communications (GreenCom), 2012 IEEE International Conference (2012).
3. Ateş, M.A., Purchasing and Supply Management at the Purchase Category Level: strategy, structure and performance, Promoter(s): Prof.dr. J.Y.F. Wynstra & Dr. E.M. van Raaij.
4. Atefeh, F., Khreich, W.: A survey of techniques for event detection in twitter. Comput. Intell., 132–164 (2013).
5. G. Cookson, “World Health Organization: Road traffic injuries.” [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries>.
6. C. Aggarwal, and P. Yu, A Framework for Clustering Massive Text and Categorical Data Streams, SDM, (2006).
7. T. Yang, R. Jin, Y. Chi, and S. Zhu, Combining link and content for community detection: a discriminative approach, KDD Conf., (2009).
8. H. Sayyadi, M. Hurst, and A. Maykov, Event Detection in Social Streams, AAAI Conf., (2009).
9. H. Schutze, and C. Silverstein, Projections for Efficient Document Clustering, SIGIR Conf., (1997).
10. N. Panagiotou, I. Katakis, and D. Gunopulos, “Detecting events in online social networks: Definitions, trends and challenges,” vol. 9580, Springer, (2016).

11. S. B. Kaleel, M. AlMeshary, and A. Abhari, “Event detection and trending in multiple social networking sites,” (2013).
12. W. Dou, X. Wang, W. Ribarsky, and M. Zhou, “Event detection in social media data,” in IEEE, (2012).
13. Allan, J., Papka, R., Lavrenko, V.: On-line new event detection and tracking. In: SIGIR, pp. 37–45 (1998).
14. AlSumait, L., Barbará, D., Domeniconi, C.: On-line lda: adaptive topic models for mining text streams with applications to topic detection and tracking. In: ICDM, pp. 3–12 (2008).
15. Beckmann, N., Kriegel, H.-P., Schneider, R., Seeger, B.: The r*-tree: an efficient and robust access method for points and rectangles. In: SIGMOD, pp. 322–331 (1990).
16. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. 3, 993–1022 (2003).
17. Chang, Y.-L., Chien, J.-T.: Latent dirichlet learning for document summarization. In: ICASSP, pp. 1689–1692 (2009).
18. Cheng, Z., Caverlee, J., Lee, K.: You are where you tweet: a contentbased approach to geo-locating twitter users. In: CIKM, pp. 759–768 (2010).
19. Zhang, K., Zi, J., Wu, L.G.: New event detection based on indexingtree and named entity. In: SIGIR, pp. 215–222 (2007).
20. Zhao, Q., Mitra, P.: Event detection and visualization for social text streams. In: ICWSM (2007).
21. Zhao, Q., Mitra, P., Chen, B.: Temporal and information flow based event detection from social text streams. In: AAAI, pp. 1501–1506 (2007).
22. Zhou, X., Zhou, X., Chen, L., Bouguettaya, A.: Efficient subsequence matching over large video databases. VLDB J. 21(4), 489–508 (2012).

23. Zhou, X., Zhou, X., Chen, L., Shu, Y., Bouguettaya, A., Taylor, J.A.: Adaptive subspace symbolization for content-based video detection. *IEEE Trans. Knowl. Data Eng.* 22(10), 1372–1387 (2010).
24. Zunjarwad, A., Sundaram, H., Xie, L.: Contextual wisdom: social relations and correlations for multimedia event annotation. In: *ACM Multimedia*, pp. 615–624 (2007).
25. Panagiotou N, Katakis I, Gunopulos D. Detecting events in online social networks: definitions, trends and challenges. In: Michaelis S, editor. *Solving large scale learning tasks: challenges and algorithms*. Cham: Springer;. p. 42–84 ,(2016).