CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

This chapter introduces the research methodology used to analyze the trends in public reactions to the topic of the 2024 U.S. election, as posted on social media platforms. The digital influence is derived from the X (formerly Twitter) platform, which is one of the most popular and reliable social media platforms in North America. The use of deep learning techniques is the most common approach in this research field. RNNs and LSTMs can parse and recognize complex data from social media platforms. This chapter outlines the problem identification and introduces the research framework, which includes the processes of data collection, data pre-processing, model training, and model development. This study provides guidelines for the project, aiming to gain insights into trends from the complex but real social media environment.

3.2 Research Framwork

The research framework outlines the steps in the lifecycle of a data science project, which are exhibited in Figure 3.1.

(a) **Problem Identification:** To identify the real trends in public reactions, it is necessary to define efficient keywords related to the topic of the study. These include the election and the main participants of the election. This ensures high data quality and scope for the project. Pre-processing is used to verify machine learning model parameters and help develop the analysis system.

(b) **Data Collection:** Get Twitter content data from the X platform using the Twitter API. It is based on specific keywords and hashtags.

(c) **Data pre-processing:** Clean the data from the previous step and transform it into a structured dataset for Exploratory Data Analysis (EDA) to gain insights and identify potential associations.

(d) **Model Training:** Compare the target label with the real data analysis results to adjust the input parameters and address the high quality of training data and the training algorithm using the support vector machine (SVM).

(e) **Model Evaluation:** Compare and test algorithm and deep learning to find the balance between interpretability, performance, and complexity.).

(f) **Model Presentation:** Visualize the results generated by the model.

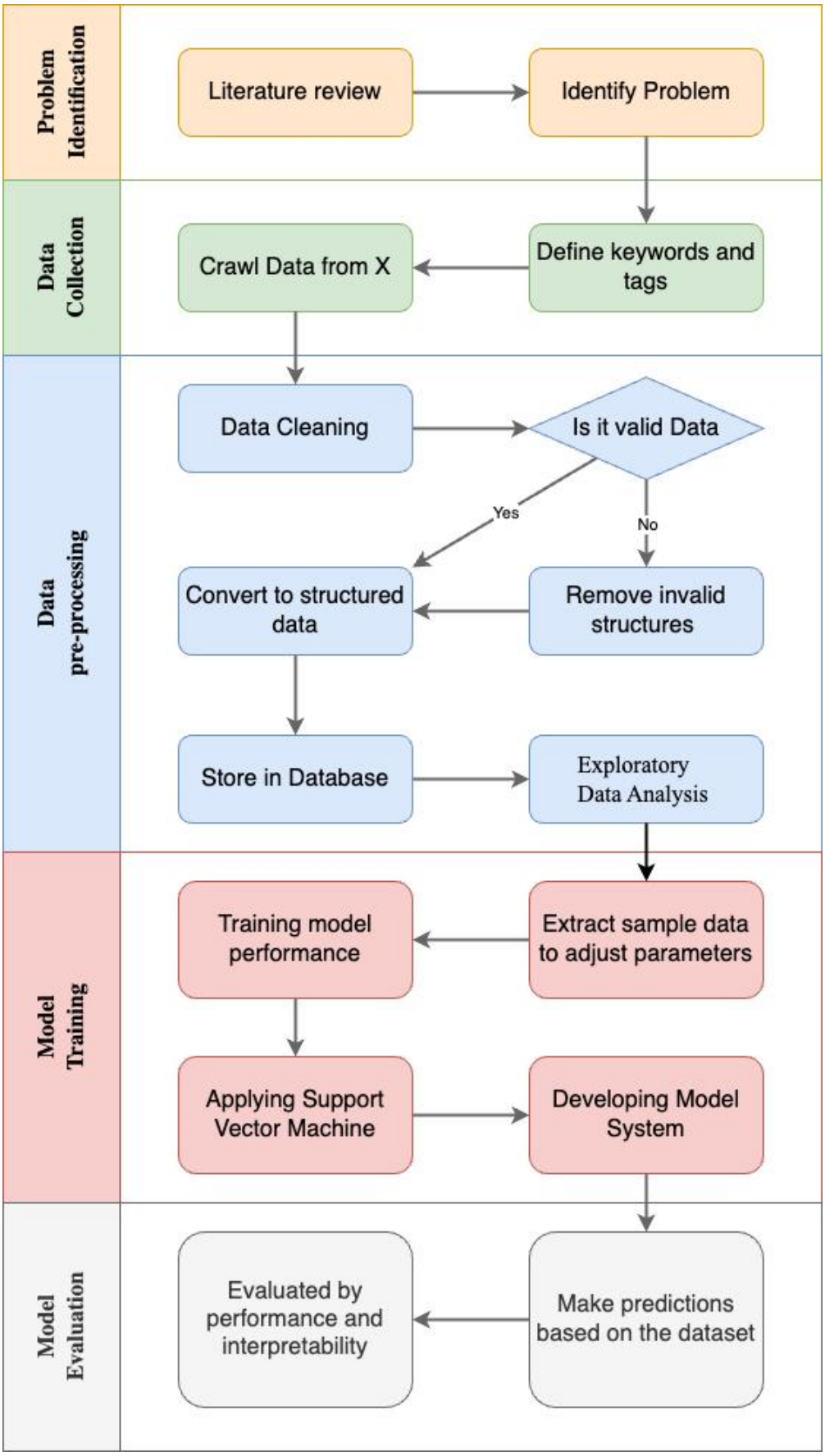

Figure 3.1 Research framework

3.3 Problem Identification

The essential component of analyzing the trends of the 2024 U.S. election is extracting the sentiment from the post content. To correct the biased analysis of traditional media and avoid drawing incorrect conclusions, this project needs to solve the problems using support vector machine (SVM) algorithms and LSTM, and identify the metrics of the analysis results.

3.4 Data Collection

The collected data is split into two methods: one involves scraping data using crawling techniques, and the other uses the Twitter API framework. To filter the data sources and ensure high-quality data scraping, the project first extracted the most frequent relevant hashtags as search keywords, including: U.S. election, election 2024, Trump, Harris, Biden, and etc.

As X has enabled new technology to prevent users from scraping data. The upper limit of each data scraping is set to 100, and it runs every 15 minutes. And the time span of data collection is from January 1, 2024, to November 11, 2024.

```
# Data Collection

filename = 'america_election_2024.csv'
search_keyword = 'election since:2024-01-01 until:2024-11-11 lang:id'
limit = 100

!npx -y tweet-harvest@2.6.1 -o "{filename}" -s "{search_keyword}" --tab "LATEST" -l {limit} --token {twitter_auth_token}
```

Figure 3.2 Data Collection

3.5 Data Pre-Processing

The original dataset contains a total of 5,000 entries and 15 columns, as displayed in Figure 3.3. The data cleaning process involves removing duplicated rows and irrelevant columns. Next, handle the missing data values by dropping unnecessary information, ignoring null or NaN values, and replacing NaN with 0 in calculated columns. Finally, validate the dataset before providing it to the model.

	conversation_id_str	created_at	favorite_count	full_text	id_str	image_url	in_reply_to_screen_name	lang	location	quote_count	reply_count	retweet_count	
0	1814232945538007295	Mon Sun 10 11:48:23 +0000 2024	810	Congratulations President @realDonaldTrump, no...	7a654439-bf23-4c27-ac6f-a463170ec9e0	NaN	rafael	en	United States	0	0	0	https:/
1	1814232671591276810	Mon Sun 10 11:48:10 +0000 2024	556	We have a convicted criminal president!!!!!! ...	547f859e-8434-4be9-a597-7a7498a8a1d1	NaN	americaelige	en	United States	15	2	8	https://
2	1814156048871379202	Mon Sun 10 11:47:58 +0000 2024	643	@Elon8558 you have no right to disconnect me f...	fa017152-8d2a-4c55-a020-1117e0679d89	NaN	slice_711	en	NaN	1	0	1	https://
3	1814232124473692517	Mon Sun 10 11:47:53 +0000 2024	586	#cryptocurrency represents a transformative an...	643fdeaf-df1c-4499-b017-f066a6f4a5df	https://pbs.twimg.com/media/GbmxAZMX0AEs0Gq?fo...	am_in_am_out	en	NaN	0	0	0	https://x.
4	1810548803512217768	Mon Sun 10 11:47:51 +0000 2024	783	@HillaryClinton hey Gurl I hope u wi the elect...	2451f5af-ce33-4dac-988b-94a2ef290d59	NaN	himmyontop	en	United States	0	0	0	https://x.
...	...	...	...	...	...	...	...	...	...	...	...	...	...

Figure 3.3 Dataset

Data cleaning is an important preprocessing step before using the dataset for training the model, ensuring high-quality data and optimal system performance. Program the cleaning function to handle the Twitter content. The following step involves removing irrelevant information from the tweet content.

As shown in Figure 3.4, the function removes URLs, mentions, hashtags, numbers, and special characters or punctuation that have no useful information for the analysis model. It also converts the text to lowercase to ensure higher performance for the analysis, as the concepts are treated equally in the pre-processing step.

Using the tokenization function, the content is split into individual words as input parameters to ensure the system understands the information better. Next, remove the stopwords, which are a special case for this project, and then join the split fragments to form new, more understandable sentences.

```python
def clean_tweet(tweet):
    # Remove URLs
    tweet = re.sub(r'http\S+|www\S+|https\S+', '', tweet, flags=re.MULTILINE)

    # Remove mentions (@username)
    tweet = re.sub(r'@\w+', '', tweet)

    # Remove hashtags (#hashtag) but keep the text
    tweet = re.sub(r'#\w+', '', tweet)

    # Remove numbers
    tweet = re.sub(r'\d+', '', tweet)

    # Remove special characters and punctuations
    tweet = re.sub(r'[^\w\s]', '', tweet)

    # Convert to lowercase
    tweet = tweet.lower()

    # Remove extra spaces
    tweet = re.sub(r'\s+', ' ', tweet).strip()

    # Tokenize the tweet
    words = word_tokenize(tweet)

    # Remove stopwords (optional, depending on your use case)
    stop_words = set(stopwords.words('english'))
    words = [word for word in words if word not in stop_words]

    # Rejoin words back into a sentence
    cleaned_tweet = ' '.join(words)

    return cleaned_tweet
```

Figure 3.4 Data Cleaning

The dataset from the previous step is derived from the original collected dataset, and it is now prepared for training. Comparing the processed dataset with the original dataset, there are 4,762 valid rows and 238 invalid entries. The rate of the full dataset is shown in Figure 3.5.

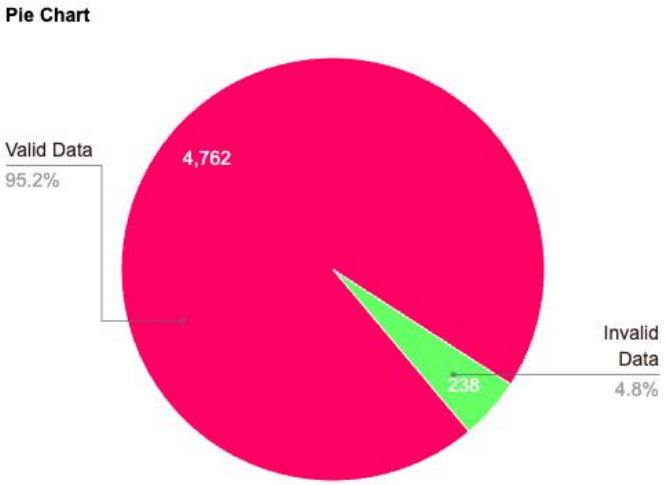

Figure 3.5 Pie Chart

3.6 Model Tranining

To train the model using the structured dataset, this project classifies the text row data into metadata. The most popular and powerful algorithm for text classification tasks is the Support Vector Machine (SVM) algorithm. The project combines it with the text vectorization technique, which is Term Frequency - Inverse Document Frequency (TF-IDF), to analyze the sentiment of tweets.

Figure 3.6 shows the code that converts the tweet text into numerical features using TF-IDF (Term Frequency-Inverse Document Frequency). It transforms the raw text into a sparse feature matrix that the SVM model can utilize.

```
# Initialize TfidfVectorizer
vectorizer = TfidfVectorizer(max_features=5000, stop_words='english')

# Convert tweet text to TF-IDF features
X = vectorizer.fit_transform(data['tweet_text'])  # X is the feature matrix (tweet text converted to numbers)
y = data['sentiment']  # y is the target labels (sentiment)

# Split the data into training (70%) and testing (30%) sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

Figure 3.6 Convert Content

Figure 3.7 shows split data into the model as input, import the SVM kernels whitch is Support Vector Classifier (SVC) to training. Set the praramters as 0.7 and 0.3 to make balance on the perfromance and accurate.

```
# Split the data into training (70%) and testing (30%) sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Initialize the Support Vector Classifier (SVM) with a linear kernel
svm_model = SVC(kernel='linear')

# Train the SVM model using the training data
svm_model.fit(X_train, y_train)
```

Figure 3.7 Model Training

3.7 Model Evaluation

From the previous step, obtain the final report, which includes the matrix used to measure the evaluation of the model's predicted results. These results are compared with the testing data, as shown in Figure 3.8.

```python
# Predict sentiment for the test set
y_pred = svm_model.predict(X_test)

# Evaluate the model's performance
accuracy = accuracy_score(y_test, y_pred)
print(f"Model Accuracy: {accuracy * 100:.2f}%")

# Detailed classification report (precision, recall, F1-score)
print(classification_report(y_test, y_pred))

# Fetch new tweets (replace with your own query)
new_tweets = fetch_tweets('Python programming', count=5)

# Extract text from the new tweets
new_tweets_text = [tweet[0] for tweet in new_tweets]

# Convert the text into the same TF-IDF format as the training data
new_tweets_vec = vectorizer.transform(new_tweets_text)

# Predict the sentiment of the new tweets
new_predictions = svm_model.predict(new_tweets_vec)

# Display the results
for tweet, sentiment in zip(new_tweets_text, new_predictions):
    sentiment_label = "Positive" if sentiment == 1 else "Negative"
    print(f"Tweet: {tweet}\nPredicted Sentiment: {sentiment_label}\n")
```

Figure 3.8 Model Evaluation

3.8 Chatper Summary

This chapter explains the methodology of this research and how the project achieves its goals using deep machine learning techniques. With the powerful and functional open-source libraries in Python, this chapter demonstrates how to collect data, process data, train the model, and evaluate the model. It provides essential technical support for generating the analysis report of the U.S. election from social media.