

Chapter 4: Exploratory Data Analysis/Initial Results

4.1 Introduction

In chapter-4 of this research, the exploratory data analysis (EDA) as well as the initial findings of the study will be outlined. It is divided into three main parts: 4.2 Visualisations and descriptive statistics for data exploration, 4.3 Feature engineering, 4.4 Expected results. Section 4.2 employs varying kinds of visualization techniques. The study will reveal the pattern, trend, direction and / or relationship, if any, between the News sentiment and stock price fluctuation in the Malaysian market. For section 4.2, by calculating the mean, median, standard deviation, as well as frequency counts the research will be able to give a detailed description of the overall sentiment distribution and characteristics in the data set. Section 4.3 applies data preprocessing to cleanse the raw data, feature conversion for categorical variables and addition of other stock market features from the raw data into a format that is more perceivable to the upcoming predictive modelling section. In section 4. 4, the preliminary hypotheses of exploratory data analysis of the factors that influence stock price changes to determine the direction and focus for improvements in quantitative modelling and assessment.

4.2 Visualisations and descriptive statistics for data exploration

Visualizations and descriptive statistics play a crucial role in exploring and understanding the data (Qin et al., 2020). The duo will be employed in this research for effective presentation and analysis of results.

4.2.1 Visualisation tools

For this study, the first step will, therefore, involve plotting histogram graphs to analyze the distribution patterns of the sentiment scores. These visualizations will give information on the measures of central tendency, variability, and Bias/Skewness of the sentiments estimates generated from the machine learning algorithms and lexicon methods. Frequency distributions of sentiments they mean histograms will be used to depict how sentiments are distributed in the articles. The distribution of the sentiment scores will be presented by box plots, which will point out the median, the first and the third quartile, and if the case, potential outliers of the data, so providing a clean picture of the data's dispersion and its middle values. Such graphical

aids, as Ortis et al. (2021) has rightly pointed corporate decision makers can understand whether the sentiment distribution is skewed or not and if it is skewed then in what extent it is skewed towards positive, negative or neutral sentiments.

Another prominent method that this research will employed is time series analysis. Visual representations of the collected data in the form of line plots can help in capturing modulations and patterning of the sentiment score which can then be mapped with the stock prices. Indeed, by taking the sample on the monthly or weekly basis, the study can see how average sentiment score changes, and what periods are the most sensitive to sentiment shift. Further, scatter and Heat maps will work for correlation or association analysis. Scatter plots can be useful in establishing the connection between sentiment scores and stock prices and might reveal trends or pattern which will help to establish the availability of a connection between media sentiment and behavior in the market. Maps of correlation matrices can effectively and briefly represent the direction of dependencies between multiple variables (sentiment scores and diverse indicators of the stock exchange) as well as the degree of their bonds. These visualizations along with the rolling mean and standard deviation plots for the sentiment will give trends and the level of volatility over time and gives a dynamic view of how sentiment is influenced by stock market.

4.2.2 Descriptive statistics

For example, descriptive statistics work hand in hand with visualization tools because it presents the data numerically (Narechania, et al. , 2020). The numeric properties of mean, median, standard deviation and variance will give qualitative explanation of the sentiment scores which will be utilised for this research. Further to that, frequency counts of the subcategories of sentiment, that is positive, negative or neutral will bring another dimension of analysis that will aid in determining the general basal sentiment of the news articles. These descriptive statistics shall be complemented by the use of visual aids, this shall ensure that the research avails a vast opportunity to analyze the sentiment data as well as patterns that determines the media sentiment in relation to the stock price. This choice of the visualization and this type of descriptive statistics not only enrich the comprehension of the dataset but also serve as a proper groundwork for the subsequent predictive modelling and research in the field of financial markets.

4.3 Feature engineering

Feature engineering is the process that enables formation of new variables (features) from raw data, which are supposed to enhance the learning models by applying prior knowledge about the application domain (Verdonck et al ., 2021). It means data selection and data preparation that act in the capacity of modifying and transforming data to create a better predictive model. In the case of Fan et al. (2019), the authors argue that good feature engineering boosts the credibility of a model as well as the quality of insights generated from the data because it directs the models' attention to what matters most in the data. As part of feature engineering in this work, raw text data from news articles are transformed into features that can be utilised for the determination of the movements in the stock prices. In this research, feature engineering will be applied in the following:

- **Text Preprocessing:** This involves **tokenization** which has to do with turning the collection of words in article content into individual tokens; **stop words removal** (remove the common words in the article content that do not contribute a lot of value or meaning), and **stemming and lemmatization** (reducing words to their root forms to ensure consistency e.g. "running" to "run").
- **Sentiment Scores:** This involves **overall sentiment score** (using NLP models to assign a sentiment score to each article, indicating whether the sentiment is positive, neutral, or negative, and **lexicon-based sentiment score** (calculating sentiment based on predefined lexicons to provide an alternative measure of sentiment).
- **Textual Features:** This involves **the term frequency-inverse document frequency (TF-IDF)**, i. e. Measuring the importance of words in an article relative to a corpus of articles, helping to highlight unique and significant terms, and **N-grams** (extracting contiguous sequences of n words e.g., bigrams, trigrams, to capture context and common phrases).
- **Temporal Features:** This has to do with **publication date and time** (incorporating the date and time the specific article was published in order to analyse temporal patterns and their impact on stock prices) and **lag features** (i.e. creating lagged versions of sentiment scores to capture delayed effects of news sentiment on stock prices, e.g., sentiment score from the previous day or week).

- **Categorical Features:** Such features include **source and author** (encoding the source and author of the article to account for potential biases or credibility variations).
- **Stock Market Features:** This includes **previous day's stock price** (including the stock price from the previous trading day to account for existing market trends) and **trading volume** (incorporating trading volume to understand market activity and sentiment impact).

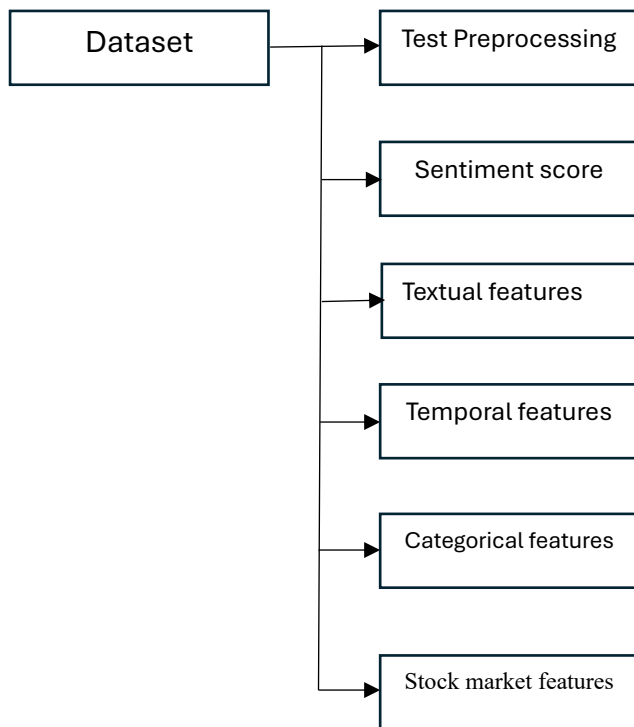


Figure 4.1: Feature engineering applications on the datasets

4.4 Expected Results

S/No	Research objectives	Expected results
1	To analyze the nuanced impact of specific sentiments expressed in financial news headlines on stock price movements within the Malaysian stock market context.	<ul style="list-style-type: none"> • Detailed Sentiment Categories: Identification and classification of specific sentiments (optimism, pessimism, fear, confidence) expressed in financial news headlines.

S/No	Research objectives	Expected results
		<ul style="list-style-type: none"> • Impact on Stock Price Movements: Quantitative analysis showing how these specific sentiments impact stock price movements. For example, headlines that expressed strong confidence might correlate with an increase in stock prices, while negative headlines might lead to a decrease. • Sector-Specific Impact: Insights to check how different sectors (technology, finance, consumer goods) are affected by various sentiment types, separately. Certain sectors may react more strongly to specific sentiments than others. • Sentiment Intensity: Analysis to check how the intensity of sentiment (strongly positive/negative vs. mildly positive/negative) affects the magnitude of stock price movements.
2	The aim is to recognize and assess the main obstacles in accurately forecasting stock prices in the Malaysian market through sentiment analysis methods, and improve advanced models such as LSTM networks to boost prediction accuracy by tackling these obstacles.	<ul style="list-style-type: none"> • Performance Metrics: Compare the performance metrics (precision, accuracy, F1-score, recall, etc.) for traditional sentiment analysis techniques and advanced machine learning models in predicting stock price movements.

S/No	Research objectives	Expected results
		<ul style="list-style-type: none"> • Model Effectiveness: Findings that indicate which models (e.g., Naive Bayes, Lexicon-based, LSTM networks) are more effective in capturing the sentiment from news headlines and predicting stock price movements. • Feature Importance: Insights into which features (specific words, phrases, or sentiment scores) are most predictive of stock price movements for each model. • Time Sensitivity: Evaluation of how different models can handle the temporal aspect of news sentiment and stock price prediction. Possibly showing that LSTM networks can account for sequential data and outperform the traditional methods.
3	To analyze the effects of various sentiment analysis algorithms, like Hybrid Naive Bayes and Opinion Lexicon-based methods, on forecasting stock price changes in Malaysia, and enhancing these algorithms to enhance prediction accuracy.	<ul style="list-style-type: none"> • Temporal Patterns: Identification of temporal patterns showing the correlation of sentiment data from news headlines with historical stock price movements. For example, a lagged effect where sentiment influences stock prices after a certain time delay. • Causality and Correlation: Analysis of causality and correlation between sentiment changes and

S/No	Research objectives	Expected results
		<p>stock price movements over different time frames (either immediate, short-term, or long-term).</p> <ul style="list-style-type: none"> • Volatility Analysis: Insights to check sentiment-driven news impacts stock price volatility over time. Periods of high sentiment activity might correspond with higher volatility. • Event-Based Analysis: Case studies of significant events (e.g., earnings announcements, political changes) and their temporal impact on stock prices, revealing specific patterns during these events. • Historical Trends: Uncovering long-term trends where certain types of sentiment consistently lead or lag behind stock price movements, providing a historical perspective on sentiment's influence on the market.

Table 4.1: Research objectives and expected results

4.4.1 Machine Learning (Initial Result)

The process and steps to get initial result of machine learning by using Scikit-learn (sklearn) at Colab as shown at below:

```
!pip install openpyxl
!pip install nltk scikit-learn pandas
```

```
import pandas as pd
import os

# Initialize a list to store DataFrames
dfs = []

# Loop through files and read each one
for i in range(1, 30):
    file_path = f'/content/filtered_sentiment_analyzed_{i}.csv'
    if os.path.exists(file_path):
        df = pd.read_csv(file_path)
        dfs.append(df)

# Concatenate all DataFrames
merged_df = pd.concat(dfs, ignore_index=True)

# Save to an Excel file
merged_excel_path = '/content/merged_sentiment_analyzed.xlsx'
merged_df.to_excel(merged_excel_path, index=False)

print(f"Merged DataFrame saved to: {merged_excel_path}")
```

```
# Load the merged sentiment data and stock price data
sentiment_df = pd.read_excel(merged_excel_path)
stock_path = '/content/financial_services.csv'
stock_df = pd.read_csv(stock_path)

print("Sentiment DataFrame:")
print(sentiment_df.head())

print("Stock Prices DataFrame:")
print(stock_df.head())
```



```

import string
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from sklearn.feature_extraction.text import CountVectorizer
import nltk

nltk.download('stopwords')
nltk.download('punkt')

# Preprocessing function
def preprocess_text(text):
    text = text.lower()
    text = text.translate(str.maketrans("", "", string.punctuation))
    tokens = word_tokenize(text)
    tokens = [word for word in tokens if word not in stopwords.words('english')]
    return " ".join(tokens)

# Apply preprocessing to the title column
sentiment_df['Processed Title'] = sentiment_df['Title'].apply(preprocess_text)

# Convert 'Published Date' to datetime format for merging
# Force both date columns to have the same format without timezone
sentiment_df['Published Date'] = pd.to_datetime(sentiment_df['Published Date']).dt.tz_localize(None)
stock_df['Date'] = pd.to_datetime(stock_df['Date'])

# Merge the news sentiment scores with stock prices based on the date
merged_data = pd.merge(sentiment_df[['Published Date', 'Sentiment Score']],
                        stock_df,
                        left_on='Published Date',
                        right_on='Date',
                        how='inner')

print("Merged DataFrame:")
print(merged_data.head())

```

```

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report

# Vectorize the text data
vectorizer = CountVectorizer(max_features=1000)
X = vectorizer.fit_transform(sentiment_df['Processed Title'])

# Define target variable for binary classification
y = sentiment_df['Sentiment Score'].apply(lambda x: 1 if x > 0 else 0) # Example binary classification

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train model
model = LogisticRegression()
model.fit(X_train, y_train)

# Predict and evaluate
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred, output_dict=True)

print("Accuracy:", accuracy)
print("Classification Report:")
print(classification_report(y_test, y_pred))

```

Accuracy: 0.7046435516767063

Classification Report:

	precision	recall	f1-score	support
0	0.70	0.48	0.57	8622
1	0.70	0.86	0.77	12461
accuracy			0.70	21083
macro avg	0.70	0.67	0.67	21083
weighted avg	0.70	0.70	0.69	21083

```
# Convert classification report to DataFrame and save to Excel file
report_df = pd.DataFrame(report).transpose()
report_excel_path = '/content/classification_report.xlsx'
report_df.to_excel(report_excel_path, index=True)
```

```
print(f"Classification report saved to: {report_excel_path}")
```

```
Classification report saved to: /content/classification_report.xlsx
```

Figure 4.2: Initial result of machine learning by Scikit-learn

```

!pip install newspaper3k
import pandas as pd
from newspaper import Article
import logging

def fetch_articles(start, end):
    # Create an empty list to store the article data
    data = []

    # Loop through the article range and extract the data
    for i in range(start, end + 1):
        url = f'https://www.malaysiakini.com/news/{i}'
        try:
            article = Article(url)
            article.download()
            article.parse()

            # Extract the relevant data
            title = article.title
            author = article.authors
            published_date = article.publish_date
            content = article.text

            # Filter articles between 2018 and 2024
            if published_date and 2018 <= published_date.year <= 2024:
                data.append({
                    'Title': title,
                    'Author': author,
                    'Published Date': published_date,
                    'Content': content
                })
        except Exception as e:
            # Log the error and skip the invalid URL
            logging.error(f"Error fetching article from URL: {url} - {e}")
            continue

    return data

# Set the range for article URLs
start_id = 710000
end_id = 720000

# Fetch articles within the date range and URL range
articles = fetch_articles(start_id, end_id)

# Create a DataFrame from the extracted data
df = pd.DataFrame(articles)

# Save the data to a CSV file with UTF-8 encoding
df.to_csv('malaysiakini_news_filtered_40to450000.csv', index=False, encoding='utf-8')

# Display the DataFrame
print(df)

```

Figure 4.3: Coding for Data Collection of News title

```
!pip install vaderSentiment
```

```
import pandas as pd
import string
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
import nltk
```

```
# To ensure have the necessary NLTK corpora
nltk.download('stopwords', quiet=True)
nltk.download('punkt', quiet=True)
```

```

# Preprocessing function
def preprocess_text(text):
    if not isinstance(text, str):
        text = str(text)
    text = text.lower()
    text = text.translate(str.maketrans("", "", string.punctuation))
    tokens = word_tokenize(text)
    tokens = [word for word in tokens if word not in stopwords.words('english')]
    return " ".join(tokens)

# Initialize VADER sentiment analyzer
analyzer = SentimentIntensityAnalyzer()

# Function to get sentiment score using VADER
def get_sentiment_score(text):
    sentiment_dict = analyzer.polarity_scores(text)
    return sentiment_dict['compound']

# Loop through files from cleaned_1.csv to cleaned_29.csv
for i in range(1, 30):
    filename = f"cleaned_{i}.csv"
    print(f"\nProcessing file: {filename}")

    # Read file
    try:
        news_df = pd.read_csv(filename)
        print(f"Original shape: {news_df.shape}")

        # Apply preprocessing to the content column
        news_df['Processed Content'] = news_df['Content'].apply(preprocess_text)

        # Apply sentiment analysis to the processed content
        news_df['Sentiment Score'] = news_df['Processed Content'].apply(get_sentiment_score)

        # Save the results
        output_filename = f"sentiment_analyzed_{i}.csv"
        news_df.to_csv(output_filename, index=False)
        print(f"Sentiment analysis completed. Results saved as: {output_filename}")
        print(f"Final shape: {news_df.shape}")
        print(news_df[['Title', 'Sentiment Score']].head())

    except FileNotFoundError:
        print(f"File {filename} not found. Skipping...")
    except Exception as e:
        print(f"An error occurred while processing {filename}: {str(e)}")

print("\nSentiment analysis completed for all files.")

```

```

Processing file: cleaned_1.csv
Original shape: (1671, 4)
Sentiment analysis completed. Results saved as: sentiment_analyzed_1.csv
Final shape: (1671, 6)

```

	Title	Sentiment Score	
0	Happy New Year from Malaysiakini	0.9451	
1	500人赴双下集会, 跨年喊“油价纳吉都要下”		0.0000
2	Seorang diplomat di Kota Darul Naim	0.8555	
3	Yoursay: Non-Muslims pay taxes, but can't be i...	0.9960	
4	Najib praises Hadi's 'better way'; 500 rally o...	0.5267	

```

Processing file: cleaned_2.csv
Original shape: (3039, 4)
Sentiment analysis completed. Results saved as: sentiment_analyzed_2.csv
Final shape: (3039, 6)

```

	Title	Sentiment Score	
0	Say goodbye to elections should Najib win, Mah...	-0.1082	
1	Maria's candidacy breathes hope for GE14	0.9985	
2	“雪槟即大马未来”, 阿兹敏冠英化身CEO卖政绩		0.0000
3	Trump ready to meet N Korea's Kim Jong-un by May	0.9678	
4	马哈迪促澳洲总理, 趁东盟峰会向纳吉提一马案		0.0000

```

Processing file: cleaned_3.csv
Original shape: (2575, 4)
Sentiment analysis completed. Results saved as: sentiment_analyzed_3.csv
Final shape: (2575, 6)

```

	Title	Sentiment Score	
0	Nurul Izzah: Nation may witness 'dirtiest poll...	0.0258	
1	TV3记者“倒戈”坦承, 报道黑函前没向安华求证		0.0000
2	PM's 'chaos if gov't changed' remark undemocra...	0.9806	
3	SAMM gives MACC documents on S'gor sand mining...	0.2023	
4	到底为什么要投废票?		0.0000

Figure 4.4: Coding to get sentiment score

4.5 Future Work

Potential avenues for future research include:

<i>Potential avenue for future research</i>	<i>Description</i>
1. Expansion and extension of even more News data sources	For instance, news aggregator such Reddit, social networks including Facebook or Instagram, a financial blog such as the Malaysian reserve, Invest Safely, and international newspapers or magazines that operate within the country and deal with the Malaysian market such as Bloomberg. Besides, try to get the possible differences of the sentiment and the impact of it coming from other news sources for the enhanced information acquisition.
Advancement of sentiment analysis by develop more sophisticated sentiment analysis techniques	The presence of the domain-specific lexicons and the fine-tuned friendly language models that are ideal for Malaysia market. Plus, examine more for finer polarity tags' differentiation like confidence & fear, optimism & pessimism to discover richer information.
Incorporating Additional Data by integrating other sources	For instance, industry-specific news, macroeconomic indicators, or social media sentiments. The purpose is to improve current methodology for the creation or discovery of the factors that potentially impacts the stock price movements and for the improvement of the effectiveness of current theories on stock price prediction.
Conduct deeper analysis of the temporal relationships between news sentiment and stock price movements	For example, lagged effects and lead-lag relationships. Moreover, it is also recommended to use more sophisticated methods such as Granger causality tests and vector autoregressive models to get better understanding concerning the mechanisms through which sentiment impacts the variation in stock prices.
Investigate impact of news sentiment	Varying the impact based on sensitivity levels to news. It will benefit the stakeholders such as investors and the financial analysts to get more

on stock price movements at the sector or industry level	insights. Additionally, application of the developed sentiment analysis and stock price prediction models into the real world trading or decision making for investment. Refining of the models in collaboration with the representatives from sector or industry, so a latest layer of strength and flexibility can be added to its application in the Malaysian financial environment.
--	--

Table 4.2: Potential Avenues for future works