GROUP WORK PROJECT # ___ MScFE 600: FINANCIAL DATA
GROUP NUMBER: ____8049_____

FULL LEGAL NAME	LOCATION (COUNTRY)	EMAIL ADDRESS	MARK X FOR ANY NON-CONTRIBUTING MEMBER
Jiayang Li	United Kingdom	jli68666666@gmail.com	
James Asira	Kenya	jjasira2016@gmail.com	
Dev Sandipkumar Bodiwala	India	devexbodiwala@gmail.com	

Statement of integrity: By typing the names of all group members in the text boxes below, you confirm that the assignment submitted is original work produced by the group (excluding any non-contributing members identified with an "X" above).

Team member 1	Jiayang Li
Team member 2	James Asira
Team member 3	Dev Sandipkumar Bodiwala

Use the box below to explain any attempts to reach out to a non-contributing member. Type (N/A) if all members contributed.
Note: You may be required to provide proof of your outreach to non-contributing members upon request.

Part 1:

Q1:

The paper uses the two exchange-traded funds (ETFs) from emerging markets to predict stock market movements. Specifically, it uses the iShares MSCI Chile ETF (ECH) which Tracks Chilean equities and iShares MSCI Brazil ETF (EWZ) which tracks Brazilian equities. The data includes historical price data including the opening price, the highest price, the lowest price, the closing price, trading volume, the adjusted close price. Technical indicators are derived from these data using Pandas Technical Analysis Library (Pandas TA). With the use of Pandas TA, the feature set is expanded by an additional 210 indicators from the historical price data (Open, High, Low, Close, Volume and Adjusted Close) which only contains six features. Therefore, the dataset is composed of 216 daily features in total.

Technical indicators play a vital role in stock price forecasting as they provide insights into market behavior and help identify patterns and trends. For example, indicators like moving averages help identify prevailing trends (uptrend, downtrend, or sideways). Indicators such as relative strength index (RSI) and momentum oscillator assess the strength or weakness of price movements, helping to predict reversals or continuations of trends. In addition, technical indicators also contribute to data reduction and pattern recognition. For example, by focusing on meaningful patterns (e.g., crossovers, divergences, or breakout levels), indicators reduce noise and make price action more interpretable. In this paper, using technical indicators is important also because they are used as input features for a neural network model: it enhances the model's ability to learn from historical patterns and allows the neural network to focus on the most relevant market signals, increasing prediction accuracy and reducing overfitting.

Q2:

The iShares MSCI Chile ETF (ECH) is an exchange-traded fund (ETF) managed by BlackRock, designed to replicate the performance of the MSCI Chile IMI 25/50 Index, which encompasses a broad range of Chilean equities [1]. In terms of its asset type, ECH provides investors with exposure to Chile's equity market, holding a diversified portfolio of companies across various sectors. As of the latest data, the fund's top holdings include:

Sociedad Química y Minera de Chile S.A. (SQM): 13.14%

Banco de Chile: 12.94%

Banco Santander-Chile: 8.01%

Falabella S.A.: 7.93%

Cencosud S.A.: 4.91%

These top five holdings constitute approximately 46.93% of the total assets, indicating a concentration in the materials and financial sectors [2]. As of January 22, 2025, ECH's Net Asset Value (NAV) was $26.45, with a 52-week range between $24.22 and $29.44 [4]. The fund's performance over various periods is as follows:

The fund's performance over various periods is as follows:

- **Year-to-Date (YTD):** 4.98%
- **1-Year Return:** -7.63%
- **3-Year Annualized Return:** 7.76%
- **5-Year Annualized Return:** -1.32%
- **10-Year Annualized Return:** -1.58%

These figures reflect the fund's volatility, influenced by Chile's economic conditions and global commodity prices [1].

Additional Fund Details:

- **Expense Ratio:** 0.60%
- **Dividend Yield:** 3.03%
- **Inception Date:** November 12, 2007
- **Assets Under Management:** Approximately $474.10 million

ECH offers targeted access to Chilean stocks, making it a suitable option for investors seeking exposure to this emerging market. However, potential investors should be aware of the fund's concentration in specific sectors and the inherent volatility associated with emerging markets [3].

The authors decide to run a classification problem rather than a regression problem because their primary objective was to predict the direction of stock price movements rather than predicting the exact price values. Predicting the direction simplifies the problem by focusing on whether to buy, sell, or hold an asset, which is often more actionable for investors than predicting specific price levels. This simplifies decision-making and reduces the complexity associated with high variability in exact price predictions caused by market noise.

2 other examples of how they could have defined the classification variable:

1. Based on Daily Closing Price Movement:

$Y_t = 1$ if $\text{Close}(t) - \text{Close}(t-1) > 0$,

Yt = 0 otherwise

2. Based on Percentage Change:

Yt= 1 if (Open(t)−Open(t−1))/Open(t−1) > δ,

Yt = 0 otherwise

The second definition incorporates a threshold percentage δ to filter out minor price movements that could be attributed to market noise. For example, setting δ=0.5% ensures that only significant movements are classified as 1.

Q3:

If Section 2, "Materials and Methods," is to be separated into a new Section 2 called "Data," the subcategories within this section could align with the process of handling and preparing data for the study. It can include the follow subcategories:

1. Data Sources: details about the ETFs studied (ECH, EWZ, etc.) and the specific time periods considered.

2. Data Description: explanation of the raw financial data collected, including: opening, closing, high, and low prices, trading volume, adjusted close.

3. Data Processing: steps taken to clean and preprocess the data, including using min-max approach for data normalization, dropping days containing unavailable data (choose date between 01/01/2009 and 01/01/2020).

4. Technical Indicators: list and explain the technical indicators derived from the raw data, such as moving averages.

5. Feature Selection: discussion of the statistical or algorithmic methods used to select the most relevant indicators and criteria for eliminating irrelevant or redundant features.

If Section 3 is titled "Methodology," it should focus on the techniques, models, and algorithms applied in the paper. It can include the follow subcategories:

1. Technique overview: brief description of the overall methodology used in the study, including data preprocessing, feature selection, and predictive modeling.

2. LASSO for Feature Selection: explanation of the Least Absolute Shrinkage and Selection Operator (LASSO) method and how LASSO was applied to address the issue of overfitting.

3. Neural Network Architecture: description of the multilayer perceptron model used for classification and details of the architecture including number of layers and nodes, activation functions and loss functions.

4. Model evaluation: this should include the cross-validation used in the study to evaluate how well the model generalizes to new datasets. It should include details such as how the dataset is splitted into k partitions and alternates them with training and testing the model.

Descriptive Statistics involve summarizing and understanding the characteristics of the data. They focus on relationships, distributions, and patterns in the dataset before applying predictive models. For example, Pearson Correlation is used to measure linear relationships between variables but does not make predictions.

Models refer to predictive frameworks or algorithms designed to learn from data and generate forecasts or classifications and they are used to predict outcomes based on input features. For example, LASSO is a feature selection method embedded in predictive modeling.

To divide descriptive statistics from models, we can include the descriptive statistics section and the models section separately. For the descriptive statistics section, we can include Chi-Squared, variance, principal feature analysis, pearson correlation, Mean Absolute Difference (MAD) and Dispersion Ratio (DR). For models, we can include Least Absolute Shrinkage and Selection Operator (LASSO), Tree-based Feature Selection, multilayer perceptron model (MLP) and Cross-validation.

The new section 3 can be outlined as follows:

Section 3: Descriptive Statistics

- 3.1 Exploratory Data Analysis
    - Summary of key data characteristics (e.g., averages, ranges, and distributions).
- 3.2 Correlation Analysis
    - Analysis of Pearson correlations between variables to understand feature relationships.
- 3.3 Feature Importance (Descriptive)
    - Initial insights into variable relevance based on statistical analysis.

Models

- 3.4 Feature Selection with LASSO
    - Description of how LASSO was used to refine the feature set.
- 3.5 Predictive Model: Neural Networks
    - Details on the neural network structure and implementation.
- 3.6 Model Training and Evaluation
    - Explanation of the training process and evaluation metrics.

optimization process of technical indicators used in the paper includes the following:

1.Normalization and Scaling**:** Ensures that all indicators are on the same scale, preventing dominance by large-scale features.

2.Feature Engineering: Creation of new, meaningful features by combining or transforming existing indicators (e.g using Technical Analysis Library to derive additional features).

3. Cross-Validation: The selected indicators were tested in different subsets of data to ensure their robustness and stability.

The authors improve the predictive power of these indicators in the following ways:

1. Selecting relevant features: By focusing on indicators with the highest predictive value, the noise in the model is reduced and the model learns meaningful patterns more effectively.

2. Reducing computational costs: A smaller, optimized feature set requires less computational power, enabling faster training and prediction without sacrificing accuracy.

3. Neural networks perform better with high-quality inputs; optimizing indicators ensures the model captures key market signals and avoids overfitting to irrelevant data.

It is important to optimize them for the neural network model because optimized features reduce the risk of overfitting, which is a common issue with neural networks due to high capacity to memorize data. In addition, training neural networks with fewer, higher-quality inputs improves speed and convergence.

Q4:

In the paper, a feature refers to a technical indicator derived from financial time series data, such as opening prices, closing prices, high prices, low prices, and trading volume. These features serve as the input for the multilayer perceptron (MLP) model and represent various aspects of market behavior.

A feature is an input variable or attribute used to describe the data and predict the target variable. Features are derived from raw data. A method is a procedure or an algorithm used to process data, select features, or train models. A model is a framework that uses features to make predictions or classifications. In this paper, features are technical indicators that represent market trends, momentum and volatility derived from historical price data (Open, High, Low, Close, Volume and Adjusted Close). Methods include LASSO (for feature selection), normalization (to scale data), cross-validation (to evaluate models). The multilayer perceptron (MLP) is the primary model used to predict the direction of stock price movements based on optimized technical indicators.

The categories of features I have learned include price-based features such as Simple Moving Averages (e.g. 50-day, 200-day) volume-based features such as Volume-weighted average price (VWAP), momentum features such as relative strength index (RSI) and momentum oscillator.

This question has been answered in the last part of Q3.

Q5:

Cross-validation is a statistical method used in machine learning and data analysis to evaluate the performance of a model.It helps determine how well the model generalizes to unseen data by splitting the dataset into smaller subsets for training and testing.

K-fold cross validation splits the dataset into k partitions. It chooses one partition for testing and the remaining nine partitions for training. It then proceeds to the next partition for testing and the other nine partitions for training, repeating this process for all partitions and averaging the accuracy.

Jaccard distance is a measure of dissimilarity between two sets. It quantifies how different two sets are by comparing the size of their intersection (common elements) to the size of their union (all unique elements combined). It ranges between 0 and 1 with 0 representing complete similarity and 1 representing complete dissimilarity. Jaccard distance is also normalized, which means its value only depends on their relative intersection sizes and the overall set union.

We compare Jaccard Distance with Euclidean Distance and Manhattan Distance. Euclidean Distance measures the straight-line distance between two points in Euclidean space and Manhattan Distance measures the sum of absolute differences between coordinates of two points. Different from Jaccard Distance which is used for binary or categorical data, Euclidean space and Manhattan Distance are applicable for numerical (continuous) data. Also, Jaccard Distance is not scale-sensitive while Euclidean Distance is sensitive to magnitude and Manhattan Distance is less sensitive to scale than Euclidean Distance. Jaccard Distance is ranged from 0 to 1 while Euclidean Distance and Manhattan Distance are ranged from 0 to infinity. Jaccard Distance is mainly used for comparing documents based on shared terms while Euclidean Distance is used for clustering (e.g., k-means clustering) and Manhattan Distance is used for feature selection and recommender systems.

The author defines an optimal solution as a configuration of technical indicators and model parameters that maximizes the predictive accuracy of the neural network while minimizing computational complexity and overfitting. This can be seen from “employing the strategy described in Algorithm 2, selecting a subset of features that can provide better results using less computational resources is possible” in section 3 of the paper [10]. The optimal solution reflects the best trade-off between model performance and efficiency.

Step 1:

The financial problem the author aims to solve is to address the challenge of predicting stock market movements in emerging markets, which are characterized by higher volatility, lower liquidity, and less predictable behaviors compared to developed markets. This can be seen from "Our focal point is the selection of features to predict the trend of two ETFs of the emerging markets: iShares MSCI Chile ETF (ECH) and iShares MSCI Brazil ETF (EWZ), using technical and statistical analysis to later compare them against iShares Core S&P 500 ETF (IVV) [10]." The author focuses on providing an innovative, data-driven approach to improve investment decision-making, enhance prediction accuracy, reduce computational cost by selecting most salient features, and mitigate volatility risks.

Emerging and developed markets exhibit significant structural and behavioral differences that influence how predictive models are designed. These differences directly impact the choice of data, features, and methodologies for stock market predictions.

1. Emerging markets tend to have higher volatility due to political instability, economic uncertainty, and lower market liquidity while developed markets are relatively stable with lower volatility due to mature economic systems and regulatory frameworks. This requires models for emerging markets to incorporate features that can adapt to sudden changes and have preprocessing methods to manage noise and outliers.

2. Emerging markets also lack comprehensive, high-quality data and historical records may be incomplete while developed markets have abundant and reliable historical data. This requires models for emerging markets to handle sparse or noisy data and to use simplified feature sets or enhanced data preprocessing techniques.

3. Emerging markets tend to have high retail investor participation, leading to sentiment-driven and less predictable movements while developed markets are dominated by institutional investors which result in more rational and data-driven price movements. This makes features related to trading volume and news sentiment more important in emerging markets.

4. Emerging markets are often less efficient, meaning that prices may not fully reflect all available information while developed markets are generally more efficient, with prices reflecting most publicly available information. This is significant to model design since technical indicators and machine learning models may perform better as they can capture inefficiencies.

5. Emerging markets tend to have lower liquidity that leads to larger bid-ask spreads and increased price sensitivity to trades while developed markets tend to have high liquidity. This requires models in emerging markets to include liquidity indicators, such as trading volume and volatility.

These distinctions are significant for the model's design because emerging markets require indicators tailored to volatility, liquidity, and sentiment. Predictive models in emerging markets must handle noisy, incomplete, and volatile data. Robust methods like LASSO (used in the paper) help optimize feature sets for these conditions. Models for emerging markets must account for diverse and less predictable behaviors, making cross-validation and generalization techniques critical. In addition, predicting in emerging markets has higher potential for actionable insights because inefficiencies can be exploited for profit.

Step 2:

The main takeaways of the results include the following:

The optimized set of technical indicators, selected using LASSO, enhanced the neural network's predictive accuracy. The model achieved higher accuracy in classifying stock price movements compared to models using unoptimized or larger sets of indicators. By selecting approximately 5% of the total technical indicators, the study greatly reduced computational costs. It also shows that a smaller, more relevant feature set allowed the neural network to train faster without sacrificing accuracy. The method effectively captured market inefficiencies and patterns unique to emerging markets, characterized by higher volatility and lower liquidity. Investors can use this approach to make more informed decisions, potentially improving portfolio performance in these high-risk markets.The optimized approach resulted in a 2% improvement in prediction accuracy compared to using all available indicators. The authors also acknowledged that their model's performance is specific to the chosen ETFs (e.g., ECH and EWZ) and may need adjustment for other assets or markets.

The following features seem useful from the studies:

1. BBP, Bollinger band percent: this feature appears in *Selected(5)* sets of all three ETFs (ECH, EWZ, IVV)

2. BOP: Balance of power: this feature appears in *Selected(5)* sets of all three ETFs (ECH, EWZ, IVV)

3. DEC: decreasing: this feature appears in *Selected(5)* sets of all three ETFs (ECH, EWZ, IVV)

4. INC: increasing: this feature appears in *Selected(5)* sets of all three ETFs (ECH, EWZ, IVV)

5. AOBV, Archer’s on balance volume: this feature appears in Selected(5) sets of ECH and EWZ

6. CTI, correlation trend indicator: this feature appears in Selected(5) sets of ECH and EWZ

7. EBSW, even better SineWave: this feature appears in Selected(5) sets of ECH and EWZ.

8. STOCHRSI, stochastic relative strength index: this feature appears in IVV

9. WILLR, Williams % R: this feature appears in EWZ and IVV

Step 3:

We pick ECH, download its data and pick correlation as the metric. We then implement the k-cross-fold validation.

[20]:
```
import pandas as pd
import yfinance as yf
from sklearn.model_selection import KFold
import matplotlib.pyplot as plt

# Step 1: Download data for ECH
fund_ticker = "ECH"
data = yf.download(fund_ticker, start="2015-01-01", end="2025-01-01")

# Check if data was retrieved
if data.empty:
    raise ValueError("No data was retrieved for the given ticker and date range.")

# Step 2: Ensure data includes relevant columns and reset index
data.reset_index(inplace=True)
data
```

```
[**********************100%**********************]  1 of 1 completed
```

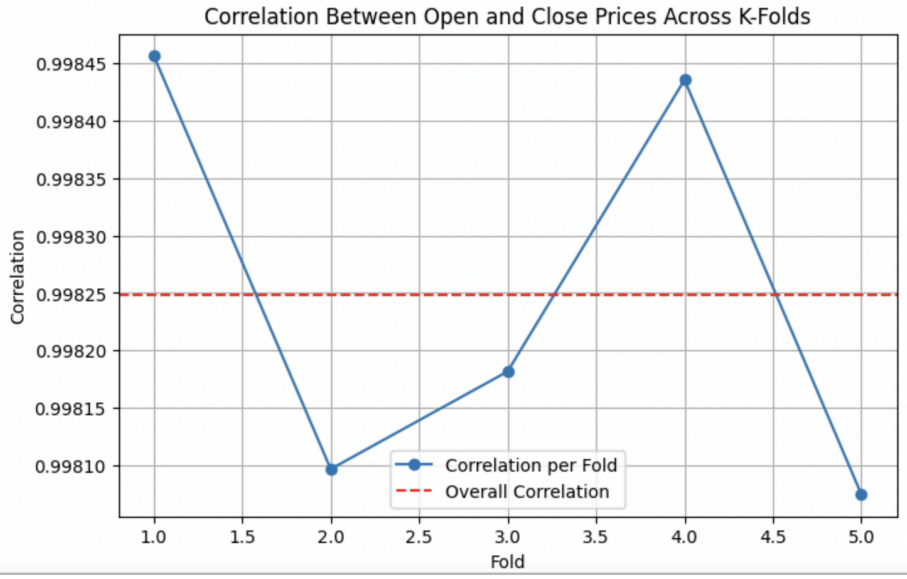
[20]:

Price	Date	Close	High	Low	Open	Volume
Ticker		ECH	ECH	ECH	ECH	ECH
0	2015-01-02	29.007072	29.360368	28.911387	29.308846	257000
1	2015-01-05	28.484493	28.948195	28.469772	28.815709	355800
2	2015-01-06	28.403524	28.712657	28.359361	28.521290	262300
3	2015-01-07	28.543371	28.727380	28.521291	28.609614	373700
4	2015-01-08	28.734745	28.852510	28.646419	28.778905	139900
...	...	...	...	...	...	...
2511	2024-12-24	25.129999	25.209999	25.020000	25.020000	58400
2512	2024-12-26	25.090000	25.240000	25.030001	25.049999	84000
2513	2024-12-27	25.139999	25.250000	25.030001	25.070000	153600

[22]:
```python
# Flatten MultiIndex if necessary
if isinstance(data.columns, pd.MultiIndex):
    data.columns = [col[0] for col in data.columns]  # Extract first level of MultiIndex

# Select relevant columns
data = data[["Date", "Open", "High", "Low", "Close", "Volume"]]

# Ensure 'Open' and 'Close' are valid Series
if not isinstance(data["Open"], pd.Series) or not isinstance(data["Close"], pd.Series):
    raise TypeError("Columns 'Open' and 'Close' are not valid Pandas Series.")

# Convert to numeric (handling any potential string or NaN values)
data["Open"] = pd.to_numeric(data["Open"], errors="coerce")
data["Close"] = pd.to_numeric(data["Close"], errors="coerce")

# Step 3: Drop rows with missing values (if any)
data.dropna(inplace=True)

# Step 4: Calculate overall correlation (Open vs Close)
overall_correlation = data["Open"].corr(data["Close"])
print(f"Overall Correlation (Open vs Close): {overall_correlation:.2f}")

# Step 5: Implement k-fold cross-validation
kf = KFold(n_splits=5, shuffle=True, random_state=42)  # 5-fold cross-validation
correlation_results = []

# Perform cross-validation
for train_index, test_index in kf.split(data):
    test_data = data.iloc[test_index]

    # Ensure the test data has enough rows
    if len(test_data) > 1:
        corr = test_data["Open"].corr(test_data["Close"])
        correlation_results.append(corr)
    else:
        correlation_results.append(None)
```

```python
# Step 6: Create a results table
results_table = pd.DataFrame({
    "Fold": range(1, len(correlation_results) + 1),
    "Correlation (Open vs Close)": correlation_results
})

# Print the table
print("\nK-Fold Correlation Results:")
print(results_table)

# Step 7: Visualize the results
plt.figure(figsize=(8, 5))
plt.plot(results_table["Fold"], results_table["Correlation (Open vs Close)"], marker='o', label='Correlation per Fold')
plt.axhline(y=overall_correlation, color='r', linestyle='--', label='Overall Correlation')
plt.title('Correlation Between Open and Close Prices Across K-Folds')
plt.xlabel('Fold')
plt.ylabel('Correlation')
plt.legend()
plt.grid(True)
plt.show()
```

```
Overall Correlation (Open vs Close): 1.00

K-Fold Correlation Results:
   Fold  Correlation (Open vs Close)
0     1                    0.998456
1     2                    0.998097
2     3                    0.998182
3     4                    0.998435
4     5                    0.998075
```



Part 2 :

Pick subcategory Social Media.

1. Sources of data

Social media data is derived from various platforms where users interact and share content. Key sources include:

1. Social Media Platforms: Twitter, Facebook, Instagram, LinkedIn, and TikTok

2. Microblogging Sites: Twitter for short text-based updates.

3. Video Platforms: YouTube, TikTok, and Twitch.

4. Image Platforms: Instagram and Pinterest.

5. Discussion Forums: Reddit and Quora.

6. Third-Party APIs: Platforms like Twitter API, Facebook Graph API, and YouTube Data API.

2. Types of data

According to the paper, there are three most used types of social media data: connection-based, interest-based, and review-based [11].

Connection-based networks are platforms designed for users to freely share their opinions or thoughts and build online connections with each other. One common use of these networks is to predict stock prices based on social media posts regarding different companies with the use of sentiment analysis. In addition, connection-based networks can also be used to forecast a company's profitability and understand connections between stakeholders and companies from their interactions.

Interest-based networks, which are focused on specific topics such as games, music, or finance, attract users who are passionate and knowledgeable about their specific fields,leading to conversations that are more concentrated and valuable than those on common social media [11]. For example, people with similar interests on stocks and options can discuss their opinions on Stocktwits.com.

Review-based networks are platforms for users to rank products or services. For example, Yelp is a popular website for users to rank restaurants, bars, salons based on their experience. On Airbnb, people can rank their short and long term homestay experience and experiences in various countries and regions. Review-based networks can influence sales volume and customer attention [11]. They can also forecast the future returns and risks for a company.

3. Quality of data

Key criteria for data quality includes:

1. Relevance: Data should align with the research or business objectives.
2. Accuracy: Posts should reflect the authentic thoughts and actions of users and should not be fabricated.
3. Volume: Social media platforms generate massive data volumes; there should be appropriate filtering.
4. Timeliness: Data should be current for real-time insights.
5. Noise: Removing spam, bots, and irrelevant content enhances data utility.
6. Consistency: Formatting, language variations, and duplicates can impact consistency.

4. Ethical issues

Handling social media data comes with ethical responsibilities:

1. Privacy: Avoid collecting personal data without consent. Anonymize data when possible.
2. Consent: Adhere to platform policies and seek user permissions where necessary.
3. Bias: Be cautious of cultural, linguistic, and algorithmic biases in the data.
4. Security: Protect sensitive information to prevent breaches or misuse.
5. Misuse: Ensure that data isn't used for malicious purposes, such as disinformation campaigns.

5. Python code to import and structure into useful data structures

We choose Twitter data from Stock Market Tweets Data from IEEE Dataport. The dataset covers tweets from April 9th, 2020 to July 16th, 2020 and it was collected using the S&P 500 tag which is the reference to the top 25 companies in the S&P 500 index and the Bloomberg tag. The data is stored in a ZIP archive containing two CSV files: tweets_labelled_09042020_16072020.csv and tweets_remaining_09042020_16072020.csv. We use the following python code in Module 4 lecture 3 to extract the ZIP file containing Twitter data, iterate through the CSV files, read each file into a DataFrame and combine them into a single DataFrame.

[1]:
```
import datetime
import matplotlib.pyplot as plt
import os
import pandas as pd
import plotly.express as px
import re
import requests
import zipfile

from datetime import datetime, timedelta
```

[4]:
```
with zipfile.ZipFile('tweets.zip', 'r') as zip_ref:
    zip_ref.extractall('extracted_tweets')

tweets_folder = 'extracted_tweets/tweets'
csv_files = [f for f in os.listdir(tweets_folder) if f.endswith('.csv')]

# Loop through CSV files and read them into DataFrames
dfs = []
for csv_file in csv_files:
    file_path = os.path.join(tweets_folder, csv_file)
    df = pd.read_csv(file_path, sep=';')

    if 'full_text' in df.columns:
        df = df.rename(columns={'full_text': 'tweet_text'})
    elif 'text' in df.columns:
        df = df.rename(columns={'text': 'tweet_text'})

    dfs.append(df)

# Concatenate all DataFrames into a single DataFrame and display it
twitter_df = pd.concat(dfs, ignore_index=True)
twitter_df
```

6. exploratory data analysis of sample data

Firstly, we discover retweets in Twitter data. If a user wants to share someone's tweet with their followers, they can retweet it and "RT" is the indicator of retweets in Twitter. We take our DataFrame and split it into two parts: one contains all retweets and the other one is the remainder.

[5]:
```
twitter_df['created_at'] = pd.to_datetime(twitter_df['created_at']).dt.tz_convert('America/New_York')
# Create a DataFrame containing only retweets
retweets_df = twitter_df[twitter_df['tweet_text'].str.startswith('RT')]

# Create a DataFrame containing tweets that are not retweets
remainder_df = twitter_df[~twitter_df['tweet_text'].str.startswith('RT')]

# Display the DataFrames
print("## Retweets DataFrame:")
print("This DataFrame contains only retweets from the Twitter data.")
display(retweets_df)

print("\n## Remainder DataFrame:")
print("This DataFrame contains tweets that are not retweets.")
display(remainder_df)
```

```
## Retweets DataFrame:
This DataFrame contains only retweets from the Twitter data.
```

	id	created_at	tweet_text	sentiment
0	77522	2020-04-14 21:03:46-04:00	RT @RobertBeadles: Yo💥\nEnter to WIN 1,000 Mon...	positive
3	760262	2020-07-03 15:39:35-04:00	RT @bentboolean: How much of Amazon's traffic ...	positive
5	27027	2020-04-12 17:52:56-04:00	RT @QuantTrend: Reduce your portfolio RISK! GO...	positive
7	392845	2020-06-01 21:12:29-04:00	RT @ArjunKharpal: #Apple has cut the prices of...	negative
8	313771	2020-05-07 00:58:41-04:00	RT @SMA_alpha: The #CDC U.S. New Case data has...	negative
...	...	...	...	...
928659	938659	2020-07-15 20:01:05-04:00	RT @tradewithdough: #India is adding millions ...	NaN
928662	938662	2020-07-15 20:00:57-04:00	RT @WarlusTrades: $SPX SPY #ES_F #AMD\n\n⚠️ Co...	NaN
928664	938664	2020-07-15 20:00:42-04:00	RT @BerkshireCapGrp: MEDH GAP FILLED. TIME TO ...	NaN

350752 rows × 4 columns

Remainder DataFrame:
This DataFrame contains tweets that are not retweets.

	id	created_at	tweet_text	sentiment
1	661634	2020-06-25 02:20:06-04:00	#SriLanka surcharge on fuel removed!\n⛽📉\nThe ...	negative
2	413231	2020-06-04 11:41:45-04:00	Net issuance increases to fund fiscal programs...	positive
4	830153	2020-07-09 10:39:14-04:00	$AMD Ryzen 4000 desktop CPUs looking 'great' a...	positive
6	472959	2020-06-09 01:23:06-04:00	$863.69 Million in Sales Expected for Spirit A...	positive
9	267894	2020-05-04 11:16:29-04:00	Where to Look for Dependable Dividends\nRead M...	neutral
...	...	...	...	...
928666	938666	2020-07-15 20:00:32-04:00	Real-Time Data Recovery Stalled Amid COVID-19 ...	NaN
928669	938669	2020-07-15 20:00:23-04:00	You \n\n$SPX SPY #ES_F	NaN
928670	938670	2020-07-15 20:00:23-04:00	$KO Coca-Cola #Options #maxpain Chart, Open In...	NaN
928671	938671	2020-07-15 20:00:06-04:00	Here's a dividends watchlist \nfor the 01/04/0...	NaN
928672	938672	2020-07-15 20:00:00-04:00	$AAL TWTR $SPCE AZN $ERI RCL $JNJ AA $SNBR...	NaN

577921 rows × 4 columns

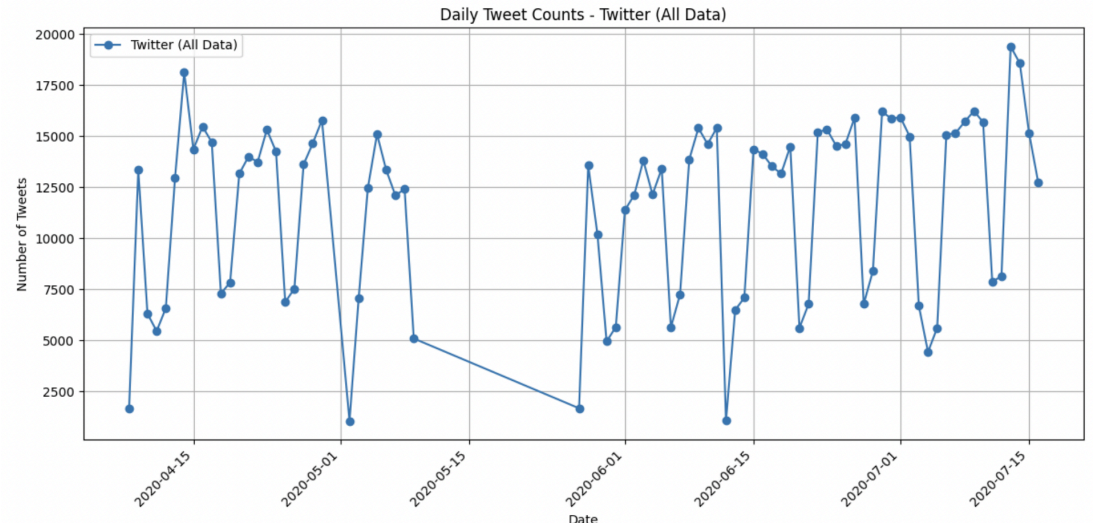
To better understand the distribution of tweets, we resample the data weekly and then count the number of tweets within each weekly interval. We can see that the tweet counts fluctuate between May and July but there are two weeks with zero tweets recorded and this may be due to the data collection error.

[5]:
```
# Get weekly tweet counts
weekly_counts = twitter_df.set_index('created_at').resample('W-MON', label='left', closed='left').count()
weekly_counts = weekly_counts.rename_axis(index="Week Starting")
display(weekly_counts)
```

Week Starting	id	tweet_text	sentiment
2020-04-06 00:00:00-04:00	33291	33291	39
2020-04-13 00:00:00-04:00	90678	90678	143
2020-04-20 00:00:00-04:00	84788	84788	106
2020-04-27 00:00:00-04:00	52106	52106	71
2020-05-04 00:00:00-04:00	70438	70438	90
2020-05-11 00:00:00-04:00	0	0	0
2020-05-18 00:00:00-04:00	0	0	0
2020-05-25 00:00:00-04:00	35982	35982	53
2020-06-01 00:00:00-04:00	75666	75666	100
2020-06-08 00:00:00-04:00	73845	73845	98
2020-06-15 00:00:00-04:00	81982	81982	124
2020-06-22 00:00:00-04:00	90650	90650	122
2020-06-29 00:00:00-04:00	79637	79637	121
2020-07-06 00:00:00-04:00	93789	93789	129
2020-07-13 00:00:00-04:00	65821	65821	104

We can also visualize the twitter data by plotting the daily tweet counts. We can do this by grouping the data by date and plot the daily counts for the twitter_df DataFrame.

```
[6]: daily_tweet_counts_twitter_all = twitter_df.groupby(twitter_df['created_at'].dt.date)['tweet_text'].count()

# Plotting
plt.figure(figsize=(12, 6))

plt.plot(daily_tweet_counts_twitter_all.index, daily_tweet_counts_twitter_all.values, label='Twitter (All Data)', marker=

plt.xlabel('Date')
plt.ylabel('Number of Tweets')
plt.title('Daily Tweet Counts - Twitter (All Data)')
plt.legend()
plt.grid(True)

plt.xticks(rotation=45, ha='right')  # Rotate x-axis labels for better readability

plt.tight_layout()  # Adjust layout to prevent labels from overlapping
plt.show()
```



In Twitter text, stock symbols can appear as cashtags or hashtags which represent the ticker symbol of a company. We can then discover hashtags in our Twitter data to get some insights into which stocks are being discussed most frequently.

```
[6]: # Discover stock symbols and hashtags
twitter_df['cashtags_hashtags'] = twitter_df['tweet_text'].str.findall(r'(?:\$|#)([a-zA-Z]+\.*[a-zA-Z]+)')
twitter_df.explode('cashtags_hashtags')['cashtags_hashtags'].str.lower().value_counts().sort_values(ascending=False).head(10)
```

```
[6]: cashtags_hashtags
stocks         262381
spx            190857
aapl           123629
spy            115031
amzn           108484
fb              85685
es              66068
stockmarket     65091
msft            62018
trading         60261
Name: count, dtype: int64
```

From the results, we can see that some mostly discussed stocks are Apple (aapl), Amazon (amzn), FB (fb), and Microsoft (msft). The S&P 500 index (spx) and SPDR S&P 500 ETF Trust (spy) are also frequently mentioned. We can then try to filter tweets with any mention of the SPY ETF (SPDR S&P 500 ETF Trust). We search the entire tweet text where the SPY ETF might be

mentioned using regular expressions. The resulting DataFrame contains a subset of the original data, specifically focusing on tweets relevant to the SPY ETF.

[7]:
```python
# Define a function to check for S&P 500 mentions with variations
def contains_spy(text):
    # Handle variations in spacing, "&", and case
    text = text.lower()  # Convert to lowercase for case-insensitive matching
    text = re.sub(r"[^a-zA-Z0-9 ]", "", text)  # Remove special characters except spaces

    # Check for different patterns
    patterns = [
        r"\bspy\b",  # Using word boundaries (\b) to avoid capturing part of word (e.g. "spying")
        r"\$spy",  # Cashtag
        r"#spy",  # Hashtag
        r"\b\$spy\b",  # Cashtag with word boundaries
        r"\b#spy\b",  # Hashtag with word boundaries
        r"\bspdr\b",  # Full name part using word boundaries
        r"\bs\&p 500 etf\b",  # Full name part with variations using word boundaries
        r"\bs\&p500 etf\b",  # Full name part with variations using word boundaries
        r"\bsp500 etf\b"  # Full name part with variations using word boundaries
        r"\bspy etf\b"  # Full name part with variations using word boundaries
        # ...add more variations as needed...
    ]

    return any(re.search(pattern, text) for pattern in patterns)

# Apply the function to filter the DataFrame and display it
filtered_twitter_df = twitter_df[twitter_df['tweet_text'].apply(contains_spy)]
filtered_twitter_df
```

[7]:

	id	created_at	tweet_text	sentiment	cashtags_hashtags
1	2	2020-04-09 19:58:55-04:00	#ES_F achieved Target 2780 closing above 50% #...	NaN	[ES, Fibonacci, SPX, SPY, tradign, futures]
10	14	2020-04-09 19:56:51-04:00	$UMRX bouncing. EXTREMELY OVERSOLD #Coronaviru...	NaN	[UMRX, Coronavirus, DECN, OPGN, CODX, HTBX, TN...
30	34	2020-04-09 19:54:28-04:00	$SPY QQQ $VXX AAPL $BA MSFT\n\nGuys, I figu...	NaN	[SPY, QQQ, VXX, AAPL, BA, MSFT]
35	39	2020-04-09 19:54:01-04:00	$AAPL SPY retest highs before retesting lows....	NaN	[AAPL, SPY]
55	59	2020-04-09 19:48:56-04:00	Traders, did you secure the 🍯 this week? $SPY ...	NaN	[SPY, ASTC, ICD, CLMT, ACY, TLSA, NLS, TSLA, B...
...	...	...	...	...	...
928644	644942	2020-06-24 08:15:40-04:00	$SPY QQQ $IWM AAPL <smh> Gonna go gre...	NaN	[SPY, QQQ, IWM, AAPL]
928645	785568	2020-07-06 09:33:50-04:00	RT @hyumialert: [$SPY QQQ $IWM SPX $NDX RU...	NaN	[SPY, QQQ, IWM, SPX, NDX, RUT, AMZN, NFLX, AAP...
928653	71944	2020-04-15 09:05:27-04:00	All these puts printing \n\n$spy nugt $jpm c...	NaN	[spy, nugt, jpm, cvna]
928667	592492	2020-06-20 16:34:07-04:00	RT @smtraderCA: "Is A Big Moving Coming?" for ...	NaN	[SPX, NDX, SPY, QQQ]
928670	627230	2020-06-23 10:08:15-04:00	$ITOX working on a contract with a fortune 500...	NaN	[ITOX, xrp, btc, spy, tsla, msft, goog, ba, fb...

99246 rows × 5 columns

7. short literature search that links to papers citing research

Social media analytics (SMA) has been extensively applied across various sectors, including healthcare, tourism, and disaster management. In healthcare, SMA is utilized to monitor public health trends and patient sentiments, providing real-time insights into disease outbreaks and treatment efficacies [5]. The tourism industry leverages SMA to understand traveler preferences and enhance destination marketing strategies [6]. Additionally, during natural disasters, analyzing social media data aids in real-time crisis management by identifying affected areas and coordinating relief efforts [7]. SMA has also been widely adopted in finance, where researchers have demonstrated how sentiment analysis of Twitter data can predict stock market fluctuations [8]. Moreover, studies on online communities show that social resilience can be examined through SMA techniques by analyzing platform behavior and engagement [9]. These applications underscore the multifaceted utility of SMA in addressing complex challenges across different fields.

Works Cited

1. BlackRock. *iShares MSCI Chile ETF (ECH) - Fund Fact Sheet.* iShares by BlackRock, www.ishares.com/us/literature/fact-sheet/ech-ishares-msci-chile-etf-fund-fact-sheet-en-us.pdf. Accessed 23 Jan. 2025.

2. Yahoo Finance. "iShares MSCI Chile ETF (ECH) Stock Price, News, Quote & History." *Yahoo Finance*, finance.yahoo.com/quote/ECH/. Accessed 23 Jan. 2025.

3. Stock Analysis. "iShares MSCI Chile ETF (ECH) Overview." *StockAnalysis*, stockanalysis.com/etf/ech/. Accessed 23 Jan. 2025.

4. BlackRock. *iShares MSCI Chile ETF (ECH) Overview.* iShares by BlackRock, www.ishares.com/us/products/239618/ishares-msci-chile-capped-etf. Accessed 23 Jan. 2025.

5. Lamsal, R. "Design and Analysis of a Large-Scale COVID-19 Tweets Dataset." *Applied Intelligence*, vol. 51, 2021, pp. 2790–2804.

6. Ngai, Eric WT, et al. "Social Media Research: Theories, Constructs, and Conceptual Frameworks." *International Journal of Information Management*, vol. 35, no. 1, 2015, pp. 33–44.

7. Stieglitz, Stefan, et al. "Social Media Analytics – Challenges in Topic Discovery, Data Collection, and Data Preparation." *International Journal of Information Management*, vol. 39, 2018, pp. 156–168.

8. Bollen, Johan, et al. "Twitter Mood Predicts the Stock Market." *Journal of Computational Science*, vol. 2, no. 1, 2011, pp. 1–8.

9. García, David, and Frank Schweitzer. "Social Resilience in Online Communities: The Autopsy of Friendster." *Proceedings of the First ACM Conference on Online Social Networks*, 2013, pp. 39–50.

10. Sagaceta-Mejía, Alma Rocío, Sánchez-Gutiérrez, Máximo Eduardo and Fresán-Figueroa, Julián Alberto. "An Intelligent Approach for Predicting Stock Market Movements in Emerging Markets Using Optimized Technical Indicators and Neural Networks" *Economics*, vol. 18, no. 1, 2024, pp. 20220073. https://doi.org/10.1515/econ-2022-0073

11. Sun, Y., Liu, L., Xu, Y. *et al.* Alternative data in finance and business: emerging applications and theory analysis (review). *Financ Innov* 10, 127 (2024). https://doi.org/10.1186/s40854-024-00652-0