Spring 2020

Relating Sentiment Expressed by Financial Twitter Accounts and Financial Index Price Movement

Jacob Edelstein Lester
Bard College

Relating Sentiment Expressed by Financial Twitter Accounts and Financial Index Price Movement

"Jake" Jacob Edelstein Lester
Bard College
Division of Science, Mathematics, and Computation
Advisors: Sven Anderson, Japheth Wood

Table of Contents

Abstract

We apply sentiment analysis to correlate price movement for two financial indices with sentiment expressed on Twitter by a select group of 93 influential financial users. We gathered close prices for the VIX and SPX indices for one month from March through April 2020 during the early stage of the COVID-19 pandemic in the U.S. as schools and businesses shut down. Tweets were also gathered during this period, although there is a large gap in collected tweets of about two weeks. We examine correlations based on five temporal resolutions from 60 minutes to 1440, which is equivalently one day.. We also used temporal offsets to analyze the correlation between relatively future price movements and current tweet sentiment. We discovered that there are small correlations suggesting Twitter sentiment may be correlated with future VIX movement.

Introduction

In this paper we studied the relationship between sentiment expressed on Twitter from financial accounts [9] and the two financial indices SPX and VIX. The SPX is an index that tracks the composite price movement of the S&P 500. We grouped our data by datetime using different resolutions and taking the average sentiment within the time period and percent change of price between latest data points of consecutive intervals for financial indices. We also looked at the correlations when offsetting financial and Twitter data by datetime. We found a small positive correlation between sentiment and VIX price movement when using a resolution of 360 minutes and an offset of 1800 minutes. This correlation was derived using Pearson's method and was deemed statistically significant with a confidence interval of 95%.

VIX

In 1993 The Chicago Board Options Exchange (Cboe) introduced a new product called the Cboe Volatility Index, also known as the VIX Index. The purpose of this index was to provide a metric to track the market's expectation of 30-day volatility as implied by at-the-money S&P 100 option prices. The VIX Index became very popular among investors and is now commonly referred to as the "fear gauge."

In 2003 Cboe collaborated with Goldman Sachs to innovate the VIX Index so as to better reflect expected volatility. This included basing VIX Index calculations on S&P 500 options over a wide spread of strike prices and aggregating this spread to achieve a more sophisticated measure for the expected 30-day volatility.

There are other VIX Index's that measure expected volatility of different time frames such as 1-week, 3-month, and 6-month. However, the 30-day VIX is the most popular and the metric we use in this paper as a reflection of expected volatility. This 30-day target time frame is predicted using options on the index representing S&P 500, it's ticker being SPX, where the options chosen are constrained to those whose expirations dates land between

23 days and 37 days. Of this interval, there are always only two expiration dates chosen. These two expiration dates are always 7 days apart.

Just as indexes, such as the S&P 500, are calculated by aggregating their component stocks so too is the VIX Index but with Option prices. The calculations for an the S&P 500 index price can be as straightforward as $SPX_{price} = \frac{1}{Index\ Divisor} \sum_{i=1}^{n \in C} \frac{market\ cap_i^2}{total\ market\ cap}$ where C is the set of component stocks and $total\ market\ cap :== \sum_{j=1}^{n \in C} market\ cap_j$. Market cap is defined as the price of a share multiplied by the number of shares in circulation. The *Index Divisor* is a constant that changes whenever companies announce dividends, stock splits, or other factors that would significantly change the value of our market-cap weighted Index. The S&P 500 index is fairly straightforward to calculate. However, most The calculations for the VIX Index are much more complex than this simple example. To understand more how the VIX is calculated please refer to the white paper authored by the CBOE [3].

Related Work

Predicting movement in financial markets has remained a relevant and profitable field of research and has garnered a lot of attention in the realms of academia and business. The stock market typically reflects finance fundamentals where expectations and realizations about quarterly earnings drive stock price changes. However, studies have demonstrated that other factors such as human emotion can also drive investment decisions. Lerner showed that sadness leads to people wishing to change their current situation and thus resulting in reduced selling prices and increased buying prices [6]. In 2003 Hirshleifer used the weather as a proxy for human mood and found that sunshine is strongly correlated with stock returns [5]. Gilbert used blog posts to gauge public sentiment and demonstrated that increases in fear correlated with decreases in SPX prices [4].

With the rise of social media in the 21st century unprecedented amounts of social data are being created, Twitter reporting more than 500 million tweets per day in 2014 [11].

Tweets are easy to access and can be gathered in large quantities over time for free. For these reasons numerous papers have used tweets as a proxy for human emotion. Bollen popularized the application of sentiment analysis on tweets to make predictions about Stock market movement [2]. They used a sentiment analysis model called OpinionFinder (OF) to generate sentiment values ranging from -1 to +1 for negative to positive sentiment. They also compared this to another model, Google-Profile of Mood States (GPOMS). Using Granger Causality analysis they found correlations between OF reported sentiment and Dow Jones Industrial Average (DJIA) price changes with an offset of 1 day, suggesting that OF can be used to predict DJIA.

Rao 2012 [10] built on the methods and results of Bollen's study and found that positive sentiment had no linear correlation between closing price while negative sentiment had small negative linear correlation between closing price for the DJIA and NASDAQ. However, by looking at the return they found significant positive linear correlation between bullishness sentiment and returns for the DJIA and NASDAQ.

The Rao 2012 paper collected approximately four million English language tweets from around one million users during the 14 month time period of 2010-06-02 to 2011-07-2011. Each record contained a tweet identifier, the date time of posting, and the text and was grouped by day. Rao does not describe the filter used when streaming tweets, suggesting that tweets were streamed using undisclosed keywords. We know that streamed tweets were likely not filtered by user accounts because of the large number of accounts collected from. Furthermore, the Rao 2012 paper makes no mention of post-stream filtering to remove url tokens.

Bollen 2010 collected approximately ten million tweets from around three million users during the 10 month time period of 2008-02-28 to 2008-12-19. Each tweet record included the tweet identifier, the date time of posting, and the text type. The paper does not explain the stream filters they used. However, they do explain that their tweet preprocessing method filters only for tweets containing statements alluding to the user’s moodstate. These key expressions were phrases such as “I feel”, “I am”, and “makes me”. They then filtered out tweet records that possessed text matching regular expressions “http:” or “www.” to remove spam tweets and tweets with information that could only be accessed externally. They went on to remove punctuation and stop-words and group tweets together by day. The tweets collected by Bollen were composed of 61.68% positive tweets and 38.32% negative tweets.

Both papers took a similar approach in calculating Twitter sentiment per day. Rao used a JSON API from Twittersentiment to categorize each tweet as negative or positive using a Naive Bayes classifier. This classifier has a predictive accuracy of 82.7%. Rao calculated a bullishness feature for each time period t as

$$B_t = ln(\frac{1+M_t^{Positive}}{1+M_t^{Negative}})$$

where B_t is the bullishness, and $M_t^{Positive}$ and $M_t^{Negative}$ are the number of positive and negative tweets per time period, respectively. They use a logarithm to amplify the ratio of positive to negative tweets as well as generate positive and negative values corresponding to the sentiment shown in the time period.

Bollen used OpinionFinder to classify tweets as positive or negative. From there, they calculated the ratio of positive tweets to negative tweets per day t as

$$OF_t = \frac{1+M_t^{Positive}}{1+M_t^{Negative}} .$$

Financial data collection was done using the Yahoo Finance API in both papers. They both collected daily prices for the Dow Jones Industrial Average (DJIA). However, there is a large distinction in how return prices were calculated. Bollen took a simple approach of calculating point difference, that is returns per day were calculated as

$$R_t = Close_t - Close_{t-1} .$$

Rao took a logarithmic approach that allowed for returns to be dynamic towards the initial close price and point change. Larger point changes had higher magnitude returns where initial close price was small and point change was large. The formula used for calculating returns was

$$R_t = \{ln(Close_t) - ln(Close_{t-1})\} * 100 .$$

Both papers use Granger Causality Analysis to investigate a pattern of lagged correlation. Rao found that there was significant high correlation with 95% confidence between DJIA and positive sentiment with a lag of two weeks as well as DJIA dn bullish sentiment with a lag of two weeks. Bollen found high correlation with 90% confidence between DJIA and the OF sentiment with a lag of one day. Rao used Pearson correlation testing and found that there was a statistically significant positive linear relationship between return and bullishness.

Methods

Data collection

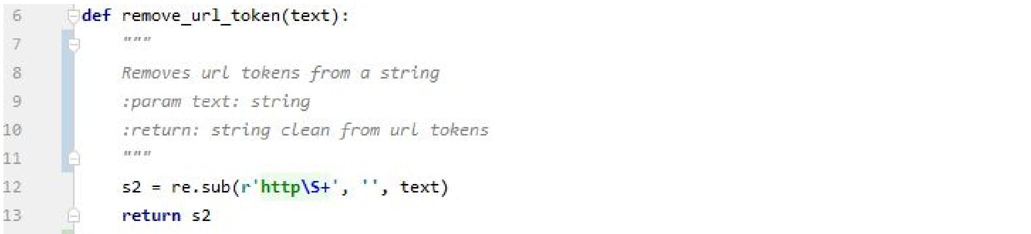
We collected tweets and financial records for the SPX and VIX indices during a historic event where the investing world reacted to the COVID-19 pandemic. We started collection on February 22nd 2020 and finished on April 7th 2020.

During the time period of 2020-02-29 to 2020-03-12 and 2020-03-28 to 2020-03-28 we collected Tweets from a select list of 93 financial influencer accounts using the Tweepy API. Tweets were streamed for all days at all times, although there are lapses in collection due to disconnect errors with the Twitter server.

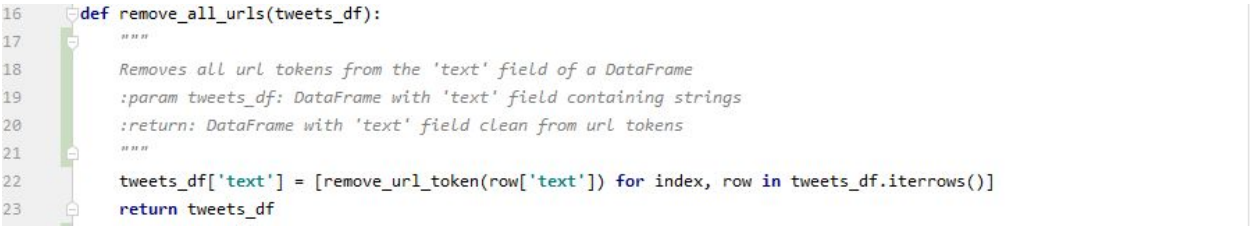
During the time period of 2020-03-02 to 2020-04-07 we collected financial quotes for the SPX and VIX indexes using the alphavantage API. The quotes for both indexes were collected on the intraday one-minute interval. Quotes are only available during market hours which excludes weekends and national holidays, such as Easter Friday.

Data Preprocessing

All url tokens in tweets.text were removed using regular expressions. Tokens such as twitter handles (ie: @realDonaldTrump), emoticons (ie: :-)), and hashtags (ie: #nyc) were not removed.



```
def remove_url_token(text):
    """
    Removes url tokens from a string
    :param text: string
    :return: string clean from url tokens
    """
    s2 = re.sub(r'http\S+', '', text)
    return s2
```



```
def remove_all_urls(tweets_df):
    """
    Removes all url tokens from the 'text' field of a DataFrame
    :param tweets_df: DataFrame with 'text' field containing strings
    :return: DataFrame with 'text' field clean from url tokens
    """
    tweets_df['text'] = [remove_url_token(row['text']) for index, row in tweets_df.iterrows()]
    return tweets_df
```

Figure 1.1.1. Code snippet for removing url tokens from tweet text using regular expressions.

Sentiment Models Used

Two different off-the-shelf sentiment analysis models, NLTK and FLAIR, were used to generate sentiment scores for tweets.VADER sentiment analysis, which will be hence referred to as NLTK, is a python library that specifically made for sentiment analysis on social media posts. VADER (Valence Aware Dictionary and sEntiment Reasoner) uses a sentiment lexicon and set of rules to predict sentiment on the closed interval [-1.0, +1.0]. The more negative or positive a value returned is, the more negative or positive the corresponding sentiment is predicted to be, respectively.

FLAIR is an NLP framework designed to provide a nice interface for training embedding-based models and performing text classification [1]. It allows for contextualized embeddings and uses PyTorch to perform sentiment classification on text. The pretrained model used in this study was trained on IMDB movie reviews and is not geared towards sentiment analysis expressed on social media. When performing sentimentiment analysis with FLAIR, the value returned is on the closed interval [-1.0, +1.0]. The more negative or positive a value returned is, the more negative or positive the corresponding sentiment is predicted to be, respectively.

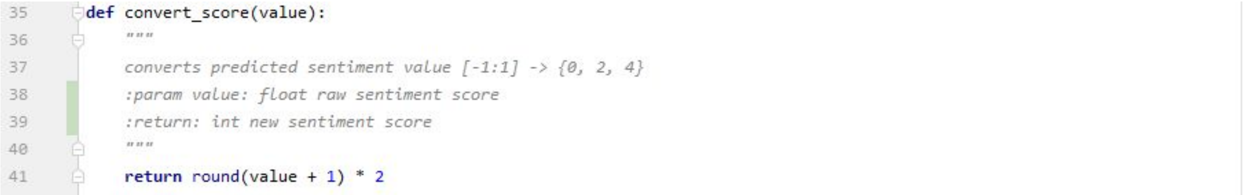
Validating Sentiment Models

Sentiment140 is a dataset of 16 million tweets labeled with sentiment value in the set {0, 2, 4}. The values 0, 2, and 4 correspond to negative, neutral, or positive sentiment respectively. The tweets were gathered using emoticons as keywords to filter the stream. The tweets were labeled automatically using those emoticons. Finally, all emoticon tokens were removed from the text.

We used a sample of 5000 tweets from the Sentiment140 testing dataset to evaluate the performance of NLTK and FLAIR on prelabeled tweets.The testing dataset did not contain any labels of 2, thereby omitting tweets with neutral sentiment.

Because both the NLTK and FLAIR models generate values on the interval [-1.0, +1.0], we developed a function (Figure 1.2.1) to convert this value into a new value contained in the set {0, 2, 4}. This conversion is done to match NLTK and FLAIR sentiment values to Sentiment140 values.

For NLTK and FLAIR individually, the counts of actual labels vs sentiment predictions were computed and stored in a table. Row indexes are the actual sentiment label and columns headers were the predicted sentiment. The value in each cell was the computed count.

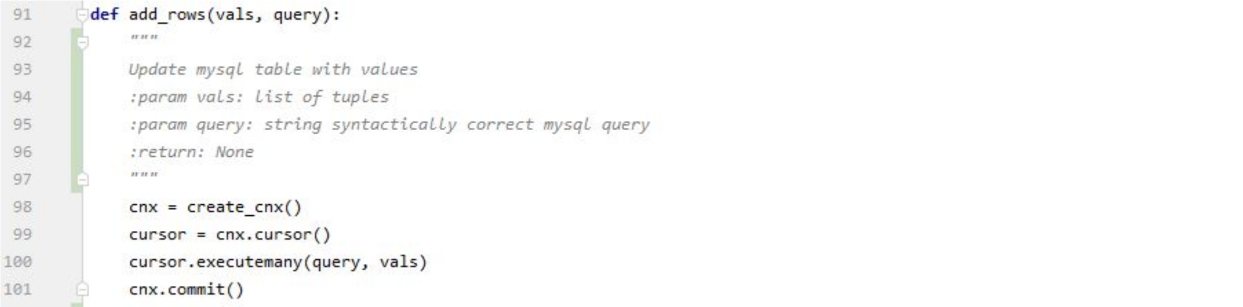


```
def convert_score(value):
    """
    converts predicted sentiment value [-1:1] -> {0, 2, 4}
    :param value: float raw sentiment score
    :return: int new sentiment score
    """
    return round(value + 1) * 2
```

Figure 1.2.1. Code snippet for converting FLAIR and NLTK sentiment scores to Sentiment140 scores.

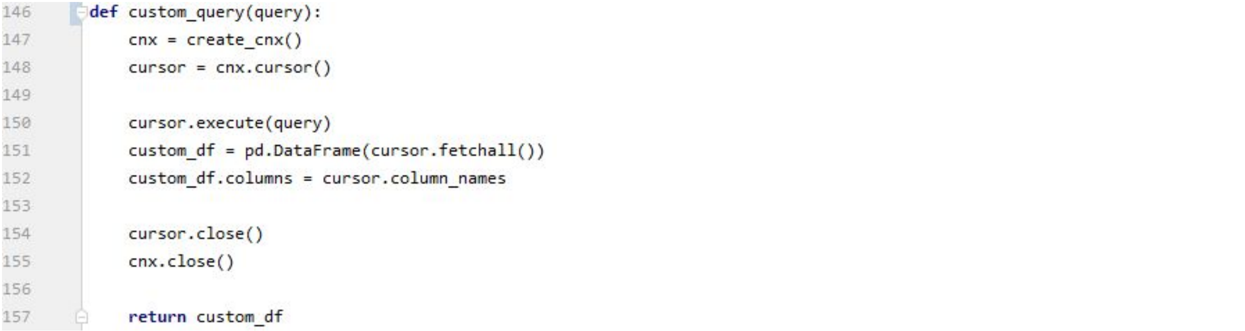
Data Storage

For this project we created a database using MySQL Workbench. We accessed this database with a Python wrapper using the Python library mysql-connector.



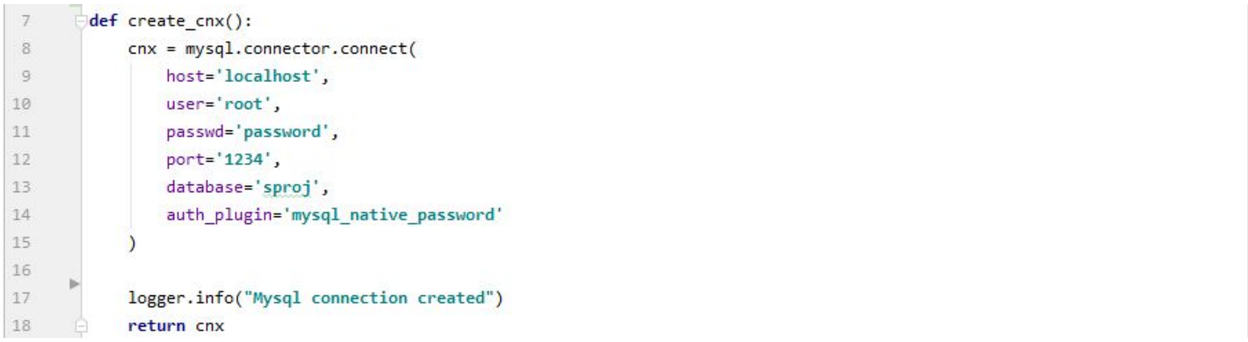
```
def add_rows(vals, query):
    """
    Update mysql table with values
    :param vals: list of tuples
    :param query: string syntactically correct mysql query
    :return: None
    """
    cnx = create_cnx()
    cursor = cnx.cursor()
    cursor.executemany(query, vals)
    cnx.commit()
```

Figure 1.3.1. Code snippet for updating existing MySQL table with new rows of data.



```
def custom_query(query):
    cnx = create_cnx()
    cursor = cnx.cursor()

    cursor.execute(query)
    custom_df = pd.DataFrame(cursor.fetchall())
    custom_df.columns = cursor.column_names

    cursor.close()
    cnx.close()

    return custom_df
```

Figure 1.3.2. Code snippet for running a query on an existing MySQL table.



```
def create_cnx():
    cnx = mysql.connector.connect(
        host='localhost',
        user='root',
        passwd='password',
        port='1234',
        database='sproj',
        auth_plugin='mysql_native_password'
    )

    logger.info("Mysql connection created")
    return cnx
```

Figure 1.3.3. Code snippet of configuration file for creating connection to MySQL database.

Our MySQL database contains six entities (Figure 1.3.4). Each entity has a private key that acts as the unique identifier.

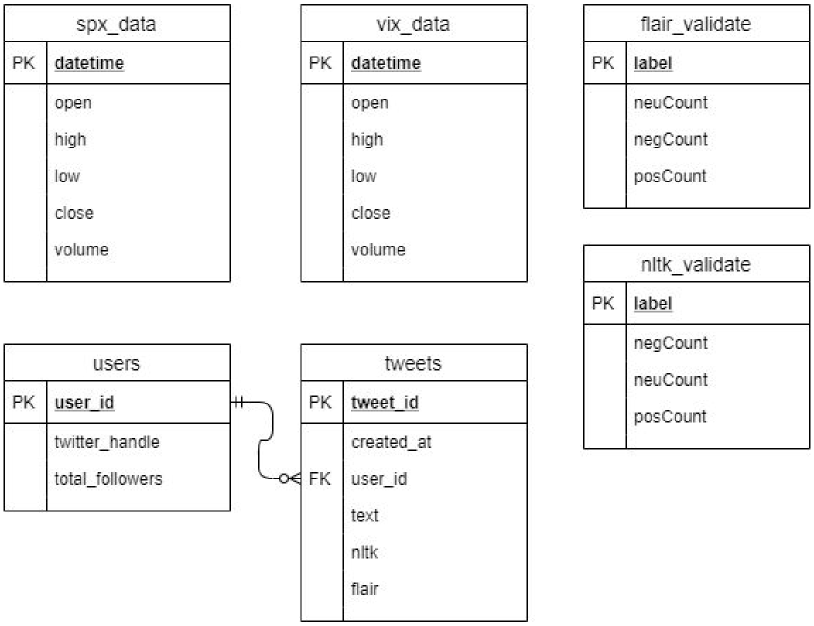
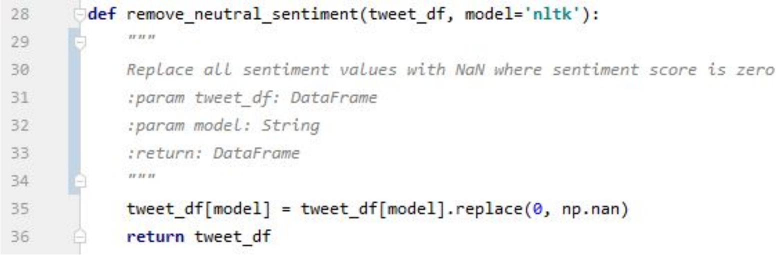


Figure 1.3.4. Entity relation diagram of MySQL database used to store data. Each box represents a table contained in the database. The name of the table is included as head. Attributes for each table are listed below. Private key per table is denoted by "PK" and is underlined.

Loading/Processing Data for Correlation

Before we could perform correlation testing, the data had to be processed. For some of the runs we set NLTK sentiment scores of zero to NaN so that they would be dropped later. We weighted sentiment scores using follower counts for the user posting and then normalized over all sentiment scores for NLTK and FLAIR individually.



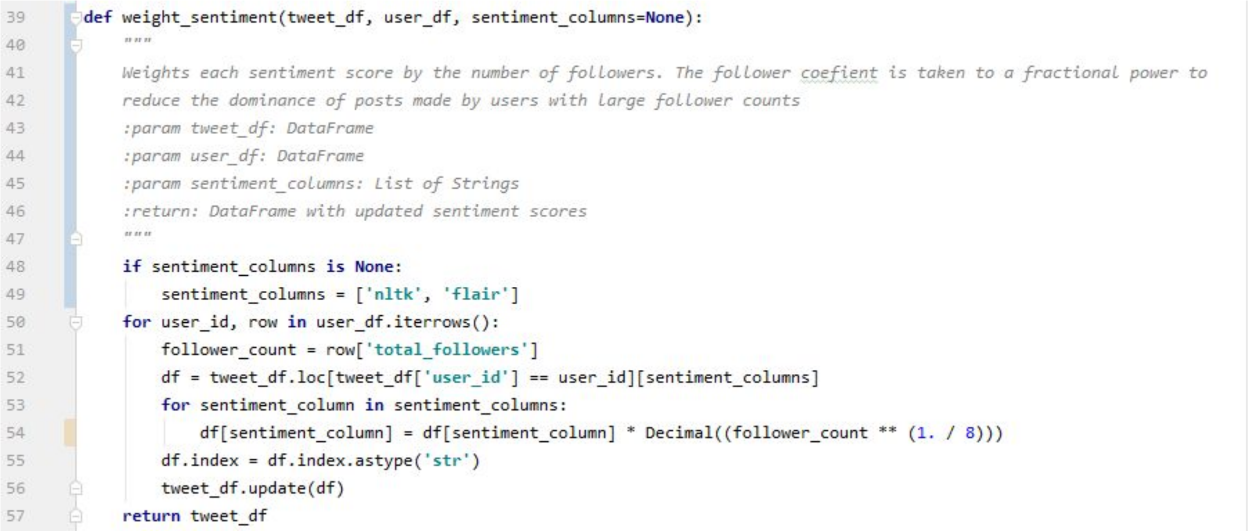
```python
def remove_neutral_sentiment(tweet_df, model='nltk'):
    """
    Replace all sentiment values with NaN where sentiment score is zero
    :param tweet_df: DataFrame
    :param model: String
    :return: DataFrame
    """
    tweet_df[model] = tweet_df[model].replace(0, np.nan)
    return tweet_df
```

Figure 1.4.1. Code snippet for removing neutral sentiment from model. Sentiment scores with value zero were replaced by numpy nan values. Removal was done later by dropping rows in Pandas DataFrame that contained nan values.

The formula we used for weighting sentiment per tweet T was

$$Score_T = Sentiment_T * (followers_U)^{\frac{1}{8}}$$

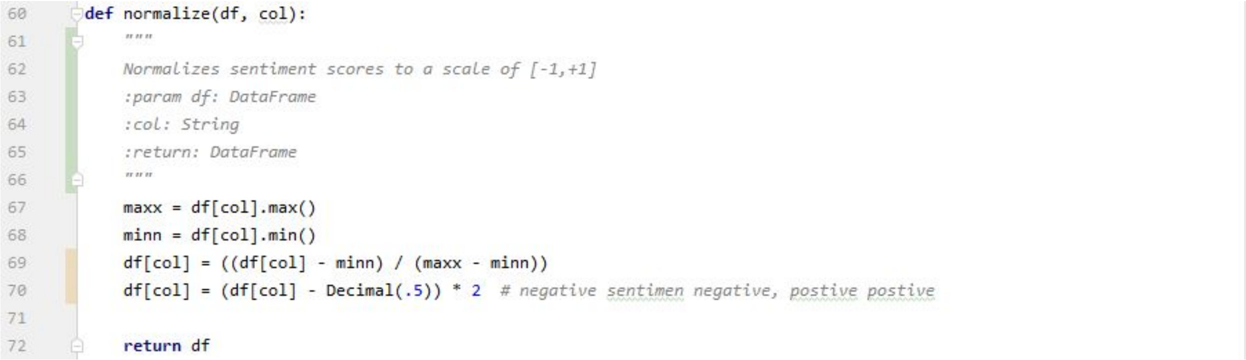
where $followers_U$ was the total number of followers the user U had and $Sentiment_T$ was the raw sentiment score generated by either NLTK or FLAIR. We operate under the assumption that users with more followers will reach more people and have a greater effect on market sentiment. The follower count coefficient was taken to the 1/8th in order to reduce the effect that follower count disparity had on sentiment weighting and allow for smaller accounts to have a greater impact and prevent only a few larger accounts from completely dominating sentiment score per day. The power 1/8th was chosen arbitrarily and it is possible that other fractional values could also work.



```python
def weight_sentiment(tweet_df, user_df, sentiment_columns=None):
    """
    Weights each sentiment score by the number of followers. The follower coefient is taken to a fractional power to
    reduce the dominance of posts made by users with large follower counts
    :param tweet_df: DataFrame
    :param user_df: DataFrame
    :param sentiment_columns: List of Strings
    :return: DataFrame with updated sentiment scores
    """
    if sentiment_columns is None:
        sentiment_columns = ['nltk', 'flair']
    for user_id, row in user_df.iterrows():
        follower_count = row['total_followers']
        df = tweet_df.loc[tweet_df['user_id'] == user_id][sentiment_columns]
        for sentiment_column in sentiment_columns:
            df[sentiment_column] = df[sentiment_column] * Decimal((follower_count ** (1. / 8)))
        df.index = df.index.astype('str')
        tweet_df.update(df)
    return tweet_df
```

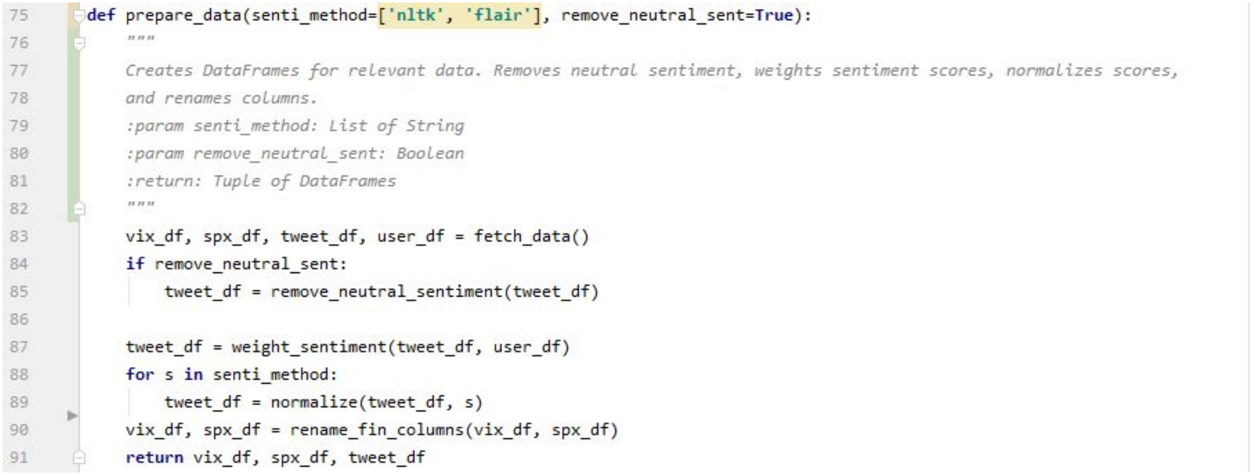
Figure 1.4.2. Code snippet for weighting sentiment by total followers user had.

Normalization was performed using linear scaling. The shortcomings of this normalization approach are reviewed in the discussion section.



```python
def normalize(df, col):
    """
    Normalizes sentiment scores to a scale of [-1,+1]
    :param df: DataFrame
    :col: String
    :return: DataFrame
    """
    maxx = df[col].max()
    minn = df[col].min()
    df[col] = ((df[col] - minn) / (maxx - minn))
    df[col] = (df[col] - Decimal(.5)) * 2  # negative sentimen negative, postive postive

    return df
```

Figure 1.4.3. Code snippet for normalizing weighted sentiment scores to interval [-1.0, +1.0] using linear scaling.



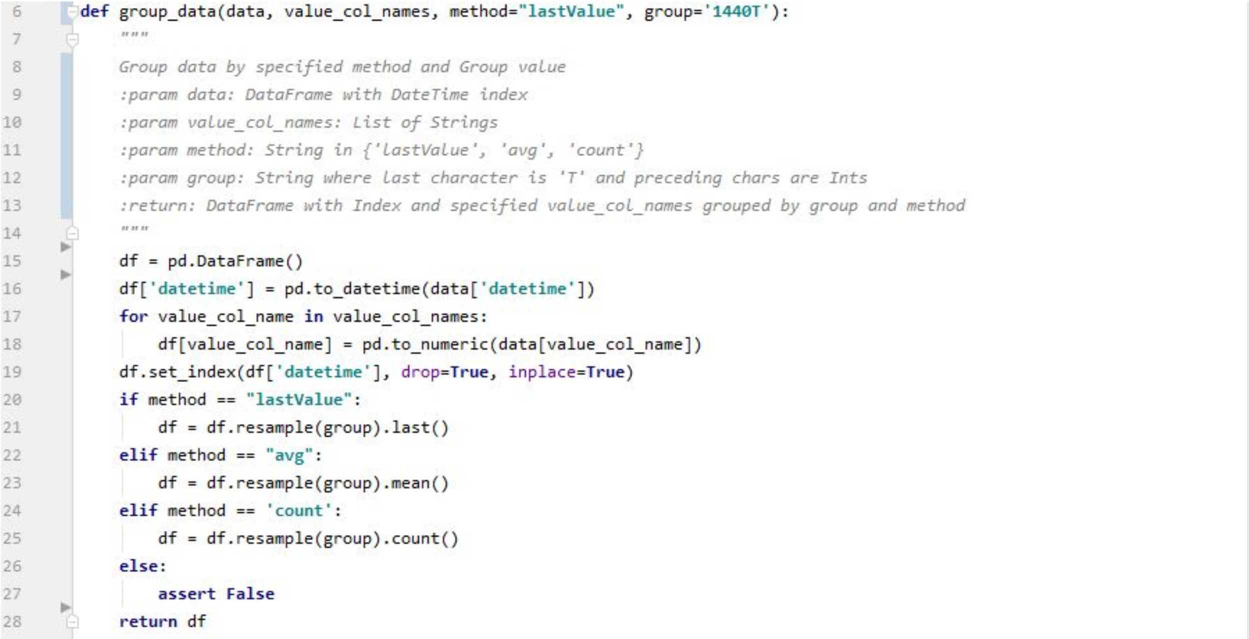
```python
def prepare_data(senti_method=['nltk', 'flair'], remove_neutral_sent=True):
    """
    Creates DataFrames for relevant data. Removes neutral sentiment, weights sentiment scores, normalizes scores,
    and renames columns.
    :param senti_method: List of String
    :param remove_neutral_sent: Boolean
    :return: Tuple of DataFrames
    """
    vix_df, spx_df, tweet_df, user_df = fetch_data()
    if remove_neutral_sent:
        tweet_df = remove_neutral_sentiment(tweet_df)

    tweet_df = weight_sentiment(tweet_df, user_df)
    for s in senti_method:
        tweet_df = normalize(tweet_df, s)
    vix_df, spx_df = rename_fin_columns(vix_df, spx_df)
    return vix_df, spx_df, tweet_df
```

Figure 1.4.4.. Code snippet for preparing data for analysis. Fetches data from MySQL database, removes neutral sentiment based on the passed parameter, normalizes the weighted scores. Returns DataFrames with datetime indexes for VIX, SPX, and tweets.

Creating Correlation Matrices

We captured the correlation coefficients between all size-two combinations of SPX close price percent change, VIX close price percent change, NLTK average weighted sentiment score, FLAIR average weighted sentiment score. Correlations were performed using data in which completely neutral sentiment was included and excluded. That is to say that sentiment scores of exactly zero were excluded. Both Pearson and Spearman correlation testing were used. These correlations were gathered using different combinations of grouping values and temporal shift values.

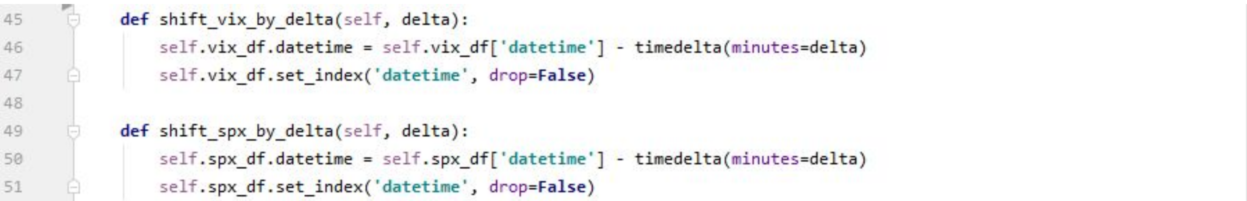
Grouping values are temporal values measured in minutes in which we grouped either the financial or Twitter data. For financial data, we grouped by the specified time period and took the last value of that period. Grouped tweet data was replaced by the mean of all sentiment scores in the specified time period. We use the terms temporal resolution and group-by values synonymously throughout this paper.



```python
def group_data(data, value_col_names, method="lastValue", group='1440T'):
    """
    Group data by specified method and Group value
    :param data: DataFrame with DateTime index
    :param value_col_names: List of Strings
    :param method: String in {'lastValue', 'avg', 'count'}
    :param group: String where last character is 'T' and preceding chars are Ints
    :return: DataFrame with Index and specified value_col_names grouped by group and method
    """
    df = pd.DataFrame()
    df['datetime'] = pd.to_datetime(data['datetime'])
    for value_col_name in value_col_names:
        df[value_col_name] = pd.to_numeric(data[value_col_name])
    df.set_index(df['datetime'], drop=True, inplace=True)
    if method == "lastValue":
        df = df.resample(group).last()
    elif method == "avg":
        df = df.resample(group).mean()
    elif method == 'count':
        df = df.resample(group).count()
    else:
        assert False
    return df
```

Figure 1.5.1. Code snippet for grouping data together by specified resolution. The method used for grouping depends on the passed parameter. Grouping can be done to return the last value per group, the average by mean, or a count of instances per group. Returns new DataFrame with datetime index where the difference between consecutive indexes are of time difference equal to the group or resolution value.

When not looking at correlations between fields at the same time periods, we shifted the financial data backward by a delta value measured in minutes relative to the timestamp of the corresponding to the sentiment data. That is to say, we looked at future financial close price percent changes and current average weighted sentiment scores as though they had occured at the same time. Throughout this paper we refer to this shift as a temporal offset or delta at times.



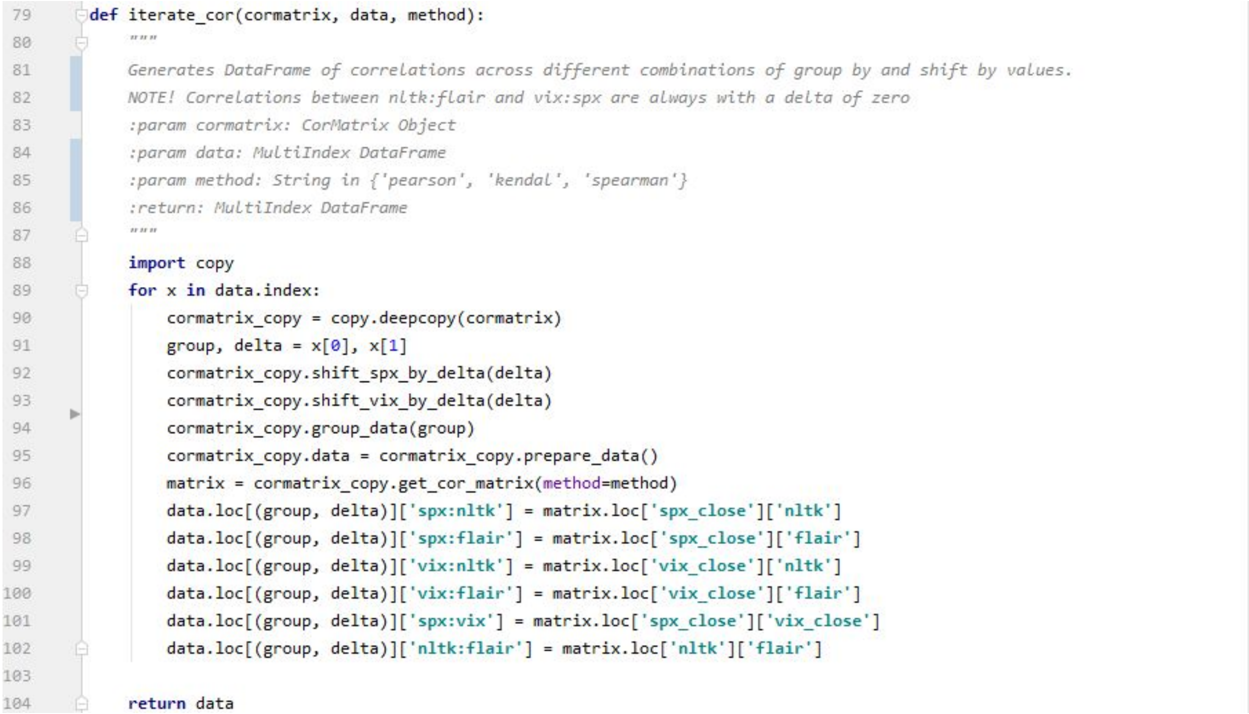
```python
    def shift_vix_by_delta(self, delta):
        self.vix_df.datetime = self.vix_df['datetime'] - timedelta(minutes=delta)
        self.vix_df.set_index('datetime', drop=False)

    def shift_spx_by_delta(self, delta):
        self.spx_df.datetime = self.spx_df['datetime'] - timedelta(minutes=delta)
        self.spx_df.set_index('datetime', drop=False)
```

Figure 1.5.2. Code snippet for applying temporal offset. A delta value measured in minutes is passed to function and depending on which method is called either the VIX or SPX rows are shifted backward. This occurs in consolidated DataFrame containing VIX, SPX, and tweet sentiments. The result is future VIX or SPX values now have indexes with datetimes equal to the initial minus the offset or delta.

Because we shifted only financial data, correlations between SPX and VIX, as well as NLTK and FLAIR were calculated without the use of a non-zero delta value.

From there SPX close price percent change, VIX close price percent change, NLTK average weighted sentiment, and FLAIR average weighted sentiment were concatenated into one Pandas DataFrame. All rows of this concatenated DataFrame containing a value of NaN were dropped.

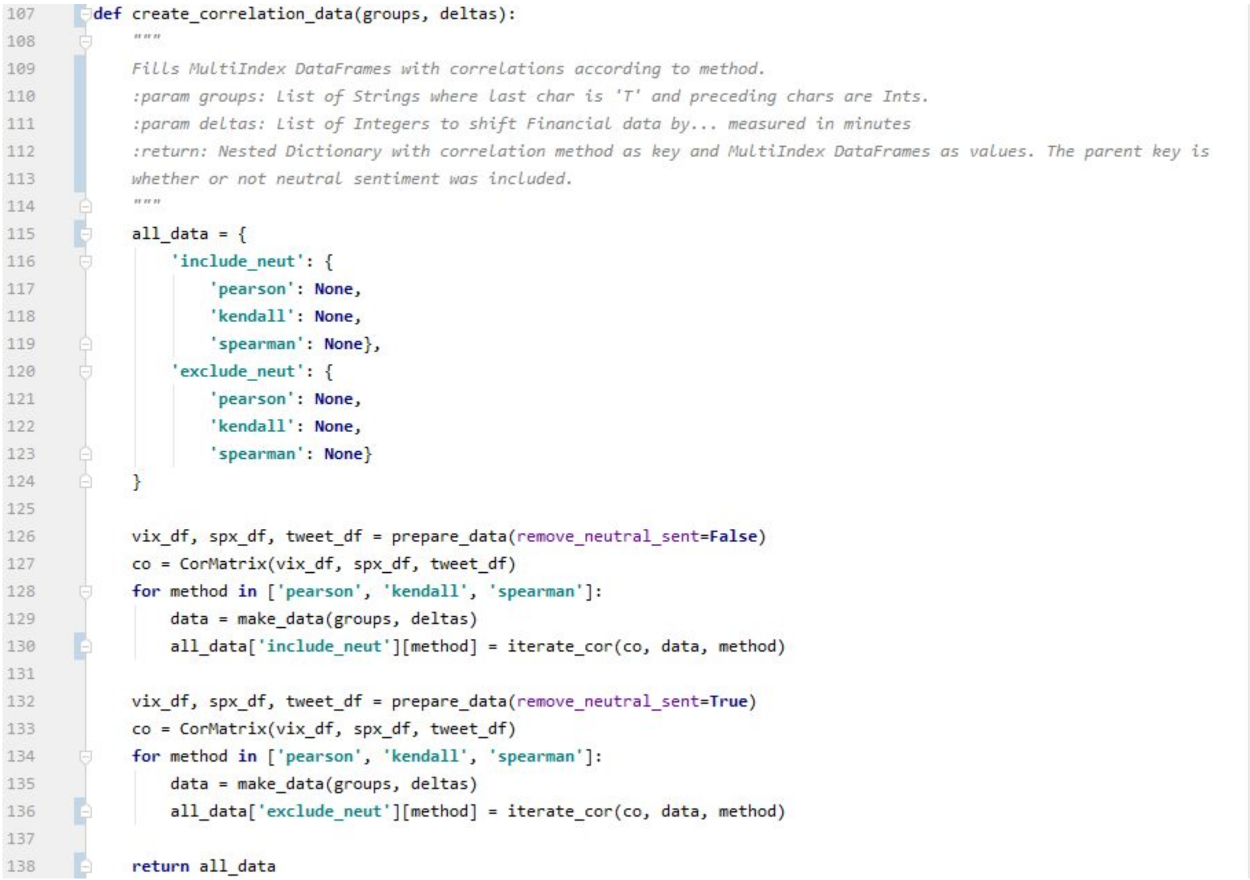
We calculated correlations across the following combinations of group-by values and delta values. Our group by values were 60, 180, 360, 720, and 1440 minutes (one day). Our delta values were a range of 0 to 4320 minutes with a step size of 30 minutes. For example, given a group value of 60 minutes and a delta value of 120 minutes, we would be finding the correlation between SPX close price percent change over 60 minutes and NLTK average weighted sentiment over 60 minutes where NLTK data at 12:00pm corresponded to SPX data at 2:00pm.



```python
def iterate_cor(cormatrix, data, method):
    """
    Generates DataFrame of correlations across different combinations of group by and shift by values.
    NOTE! Correlations between nltk:flair and vix:spx are always with a delta of zero
    :param cormatrix: CorMatrix Object
    :param data: MultiIndex DataFrame
    :param method: String in {'pearson', 'kendal', 'spearman'}
    :return: MultiIndex DataFrame
    """
    import copy
    for x in data.index:
        cormatrix_copy = copy.deepcopy(cormatrix)
        group, delta = x[0], x[1]
        cormatrix_copy.shift_spx_by_delta(delta)
        cormatrix_copy.shift_vix_by_delta(delta)
        cormatrix_copy.group_data(group)
        cormatrix_copy.data = cormatrix_copy.prepare_data()
        matrix = cormatrix_copy.get_cor_matrix(method=method)
        data.loc[(group, delta)]['spx:nltk'] = matrix.loc['spx_close']['nltk']
        data.loc[(group, delta)]['spx:flair'] = matrix.loc['spx_close']['flair']
        data.loc[(group, delta)]['vix:nltk'] = matrix.loc['vix_close']['nltk']
        data.loc[(group, delta)]['vix:flair'] = matrix.loc['vix_close']['flair']
        data.loc[(group, delta)]['spx:vix'] = matrix.loc['spx_close']['vix_close']
        data.loc[(group, delta)]['nltk:flair'] = matrix.loc['nltk']['flair']

    return data
```

Figure 1.5.3. Code snippet for creating a table of correlations using a method of Pearson or Spearman for correlation.

Correlation coefficients were calculated using Pearson and Spearman via a built-in Pandas method. We computed these correlations when completely neutral NLTK sentiment was excluded and when it was included.



```python
def create_correlation_data(groups, deltas):
    """
    Fills MultiIndex DataFrames with correlations according to method.
    :param groups: List of Strings where last char is 'T' and preceding chars are Ints.
    :param deltas: List of Integers to shift Financial data by... measured in minutes
    :return: Nested Dictionary with correlation method as key and MultiIndex DataFrames as values. The parent key is
    whether or not neutral sentiment was included.
    """
    all_data = {
        'include_neut': {
            'pearson': None,
            'kendall': None,
            'spearman': None},
        'exclude_neut': {
            'pearson': None,
            'kendall': None,
            'spearman': None}
    }

    vix_df, spx_df, tweet_df = prepare_data(remove_neutral_sent=False)
    co = CorMatrix(vix_df, spx_df, tweet_df)
    for method in ['pearson', 'kendall', 'spearman']:
        data = make_data(groups, deltas)
        all_data['include_neut'][method] = iterate_cor(co, data, method)

    vix_df, spx_df, tweet_df = prepare_data(remove_neutral_sent=True)
    co = CorMatrix(vix_df, spx_df, tweet_df)
    for method in ['pearson', 'kendall', 'spearman']:
        data = make_data(groups, deltas)
        all_data['exclude_neut'][method] = iterate_cor(co, data, method)

    return all_data
```

Figure 1.5.4. Code snippet for iterating through correlation methods and data containing neutral or no neutral sentiment and creating table of correlations accordingly.

Validation of Sentiment Algorithms

We considered two different off-the-shelf sentiment analysis models, NLTK and FLAIR, and evaluated their performance on a sample of prelabeled Sentiment140 tweets.

The NLTK model scored higher than the FLAIR model in total predictive accuracy as well as higher accuracy predicting positive and negative sentiment separately (neutral sentiment predictions were excluded from accuracy scoring). Accuracy was measured by

$$accuracy = \frac{\#\, correct\ predictions}{\#\, total\ predictions} .$$

Including neutral predictions strongly decreased NLTK accuracy in total and in part. When neutral predictions were included, NLTK scored lower than the FLAIR model in regards to total accuracy as well as higher accuracy in predicting positive and negative sentiment separately.

Sentiment Predictions on Sentiment140 Data

	Total Predictive Accuracy	Predicting Negative Sentiment Successfully	Predicting Positive Sentiment Successfully
NLTK (excluding neutral predictions)	77.8%	56.8%	93.4%
NLTK	29.0%	18.1%	39.8%
FLAIR	57.4%	48.5%	66.2%

Figure 2.1.1. Table showing accuracy of FLAIR and NLTK on the Sentiment140 data sample. NLTK (excluding neutral predictions) used data of size 1864 tweets with 792 negative labeled tweets and 1072 positive labeled tweets. The other two rows, FLAIR and NLTK, used the same sample data of 5000 tweets with 2486 negative labeled tweets and 2514 positive labeled tweets. The accuracy of each model is measured by dividing the number of correct predictions by the size of the data predicted on. Total predictive accuracy is the accuracy of all predictions, while the other two columns are for their respectively labeled predictions. The NLTK model (excluding neutral predictions) achieved great success in predicting positive sentiment correctly with 93.4% accuracy.

Sentiment140 Sample Data

The randomly sampled 5000 prelabled tweets from the Sentiment140 data had an almost uniform distribution of positive and negative labels in the range [-1.0,+1.0]. Of these 5000 tweets, 49.7% possessed negative sentiment labels and 50.3% possesed positive sentiment labels. All sampled tweets had text in which URLs and Twitter user screen names had been removed.

NLTK Model (including neutral sentiment)

The NLTK model performed rather poorly predicting labels for the Sentiment140 tweets because of a tendency to predict neutral labels despite there being no tweets labeled as neutral. The NLTK model incorrectly predicted neutral labels in 62.7% of all predictions. Total accuracy in correctly predicting sentiment labels was 29.0%. Accuracy in predicting negative labels was 18.10% and positive labels was 39.8%.

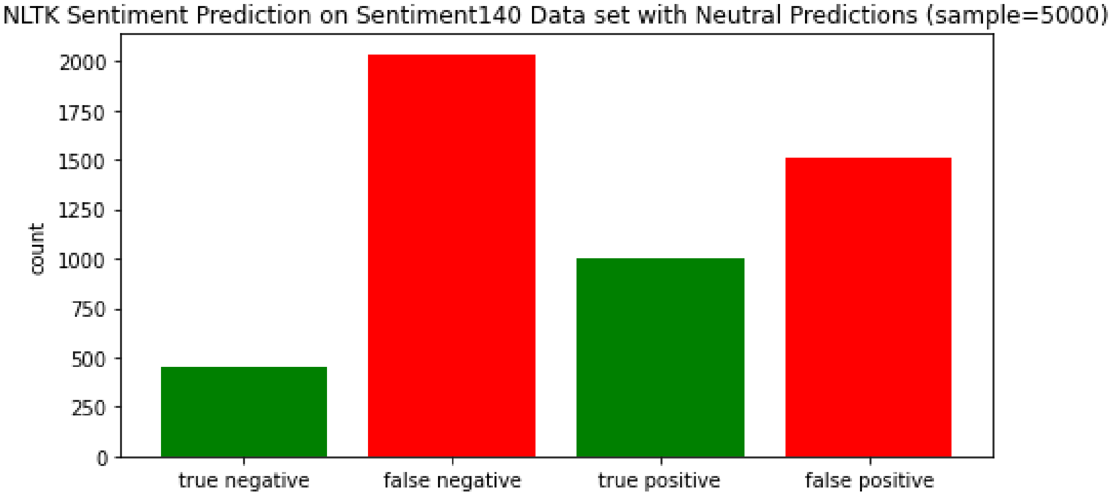


Figure 2.2.1. Bar chart of NLTK's performance on Sentiment140 sample data of size 5000. Green bars are correct predictions, while red bars are incorrect predictions. Neutral predictions were included. NLTK did poorly in predicting positive and negative sentiment, with particular shortcomings for predicting negative sentiment.

NLTK Predictions On Sentiment 140 Data Set (Size 5000)

	Negative Predictions	Neutral Predictions	Positive Predictions	Total Label Count

Negative Labels	18.1% (450)	68.1% (1694)	13.8% (342)	(2486)
Positive Labels	2.8% (71)	57.4% (1442)	39.8% (1001)	(2514)
Total Prediction Count	(521)	(3136)	(1343)	(5000)

Figure 2.2.2. Predictive accuracy of NLTK on Sentiment140 data of sample size 5000. The row labels signify which labels the tweets were actually labeled and the column headers signify what the predictions were. The values in parenthesis are the raw number of predictions, except for the last row and column where those values are the counts of either predictions or labels for corresponding fields. NLTK predicted neutral tweets more than negative and positive predictions combined. The accuracy of NLTK was very poor in both predicting negative and positive tweets, especially so with correctly predicting negative tweets.

NLTK Model (excluding neutral sentiment)

We removed all neutral predictions from accuracy scoring and saw a much different result for NLTK prediction accuracy. Excluding neutral sentiment predictions, NLTK had a total accuracy of 77.8% and FLAIR had a total accuracy of 57.4%.

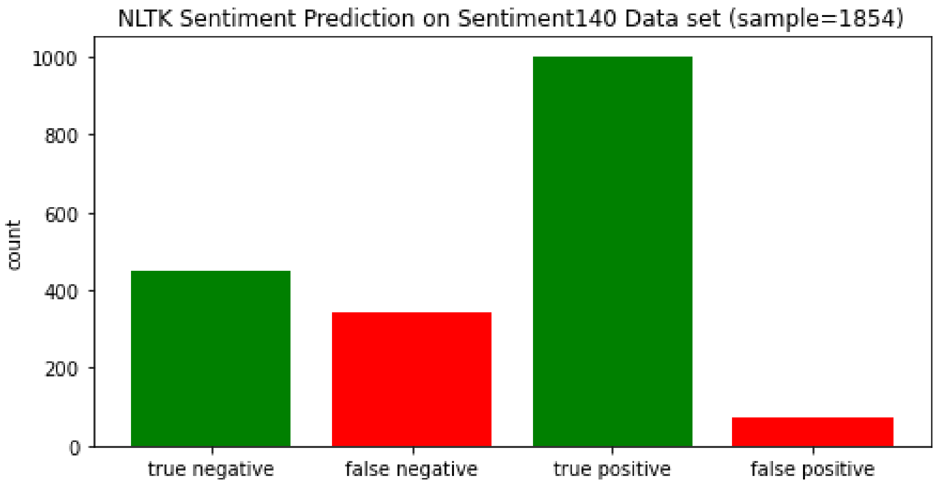


Figure 2.3.1. Bar chart of NLTK's performance on Sentiment140 sample data of size 1854. Green bars are correct predictions, while red bars are incorrect predictions. Neutral predictions were not included. NLTK did well in predicting positive and negative sentiment, with particular success for predicting positive sentiment.

NLTK Predictions On Sentiment 140 Data Set Excluding Neutral Predictions (Size 1864)

	Negative Predictions	Positive Predictions	Total Label Count
Negative Labels	56.8% (450)	43.2% (342)	(792)
Positive Labels	6.6% (71)	93.4% (1001)	(1072)
Total Prediction Count	(521)	(1343)	(1864)

Figure 2.3.2. Predictive accuracy of NLTK on Sentiment140 data of sample size 1864. Neutral predictions were not included. The row labels signify which labels the tweets were actually labeled and the column headers signify what the predictions were. The values in parenthesis are the raw number of predictions, except for the last row and column where those values are the counts of either predictions or labels for corresponding fields. Upon removing neutral predictions, the accuracy of predictions increased. The proportion of negative to positve labeled tweets shifted more towards positive, but the sample remained composed of both labels with no super majority. Ie, neither labeled composed more than two thirds of the data sample. NLTK had a bias toward predicting positive values. Despite this. the accuracy of NLTK was fairly good in both predicting negative and positive tweets, especially so with correctly predicting positive tweets at a 93.4% success rate..

FLAIR Model

The FLAIR model predicted sentiment with a 57.4% accuracy. It correctly predicted negative sentiment labeled tweets 48.5% and positive sentiment labeled tweets 66.2%. Flair had a total prediction accuracy higher than a coin toss, but failed to correctly predict negative labeled tweets for more than half of the negatively labeled tweets.

The specific FLAIR model used was trained on IMDB movie reviews and was not tailored to predicting sentiment for tweet text. Because of this, it is possible that FLAIR would have improved results if a model trained on Twitter posts was used instead.

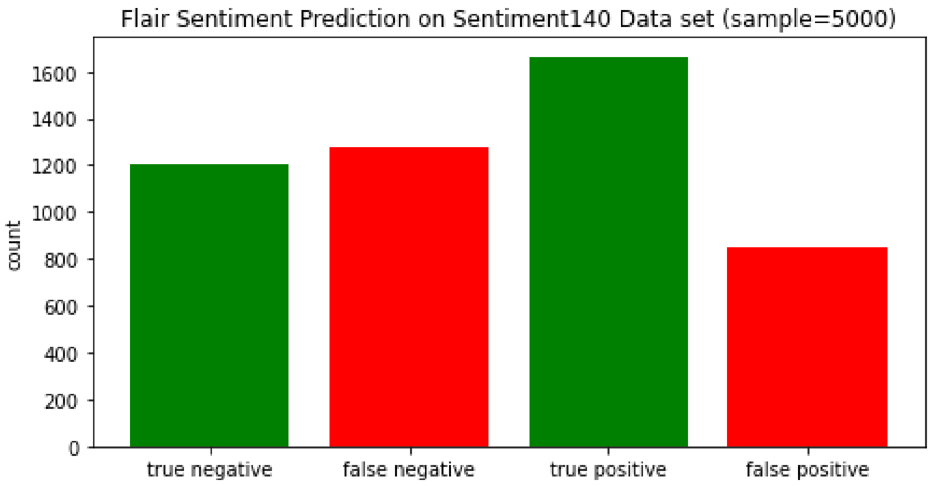


Figure 2.4.1. Bar chart of FLAIR's performance on Sentiment140 sample data of size 5000. Green bars are correct predictions, while red bars are incorrect predictions. Neutral predictions were not included.

FLAIR Predictions On Sentiment 140 Data Set (Size 5000)

	Negative Predictions	Neutral Predictions	Positive Predictions	Total Label Count
Negative Labels	48.5% (1206)	0.0% (0)	51.5% (1280)	(2486)
Positive Labels	33.8% (850)	0.0% (0)	66.2% (1664)	(2514)
Total Prediction Count	(2056)	(0)	(2944)	(5000)

Figure 2.4.2. Predictive accuracy of FLAIR on Sentiment140 data of sample size 5000. The row labels signify which labels the tweets were actually labeled and the column headers signify what the predictions were. The values in parenthesis are the raw number of predictions, except for the last row and column where those values are the counts of either predictions or labels for corresponding fields. FLAIR never predicted neutral tweets. Flair had a slight bias toward predicting positive tweets, just as with NLTK. The accuracy of FLAIR on negative tweets was comparable to guessing a coin toss at around 50%. FLAIR was more successful at predicting positive tweets with 66.2% success rate.

Conclusions

NLTK had a higher sentiment predictive accuracy on sample tweets than FLAIR when excluding neutral predictions. Because of this, we decided to exclusively use the NLTK model in correlation testing while excluding tweets with neutral sentiment predictions from the data set. The NLTK model was developed to predict tweet sentiment, whereas the FLAIR model was trained on IMDB movie reviews. This factor probably played a role in NLTK outperforming FLAIR on sentiment prediction on tweets.

Both models were more successful in correctly predicting positively labeled tweets than negatively labeled tweets. Also, both models were more likely to predict positive sentiment than negative sentiment. These two observations lead to the conclusion that both FLAIR and NLTK have a bias towards positive sentiment.

Collected Data

Financial Data

SPX

There were 28 days of SPX data recorded. The SPX Index started the month of March 2020 at around 3000 points. As COVID-19 gained recognition as a serious disease, the SPX experienced a historic selloff. This resulted in the SPX dropping 30% in value over the course of three weeks. On March 23rd the SPX hit its low of roughly 2250 points. This was not only the 52-week low but a 170-week low as the last time the SPX was this cheap was at the end of the year in 2016. After hitting this low the SPX recovered 400 points, or roughly a 17% gain. March 24th marked a historical moment for the Dow Jones which closed 11% higher, the largest percent gain since the 1930's [7]. The SPX closed roughly 9% higher that same day, marking the largest percent gain in over a decade. The SPX Index managed to maintain momentum into April and stabilized at around 2800 points.

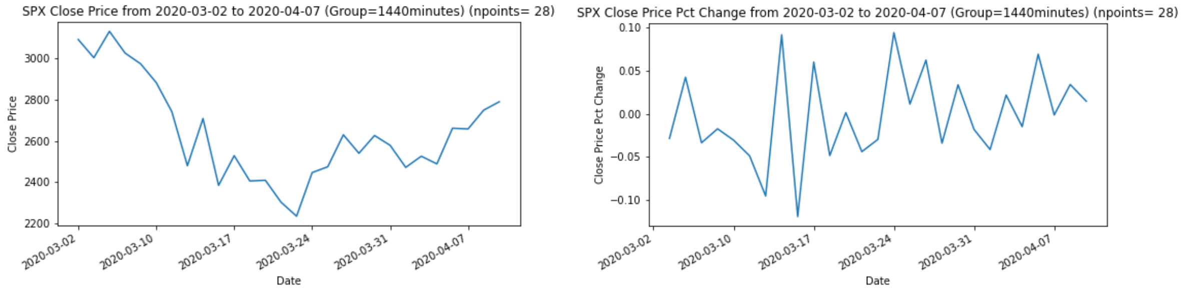


Figure 3.1.1. Trend showing SPX price over time (left) and SPX price change in percent over time (right). Both trends are taken from 2020-03-02 to 2020-04-07. Data is aggregated with a resolution of 1440 minutes, or 1 day equivalently. Missing points such as weekends are excluded to provide a smoother trend. The data captured shows the steep drop in price as markets reacted to COVID-19.

VIX

The VIX index was also affected by COVID-19 fears and hit it's high on March 16th, of around $82.70. This marked an almost 150% gain in value. The VIX hasn't closed this high since December 2008. On March 24th, the VIX index closed at a little over $62.50. The VIX index has continued to decline from there into April. The close price trend of the VIX is negatively related to the SPX close price trend.

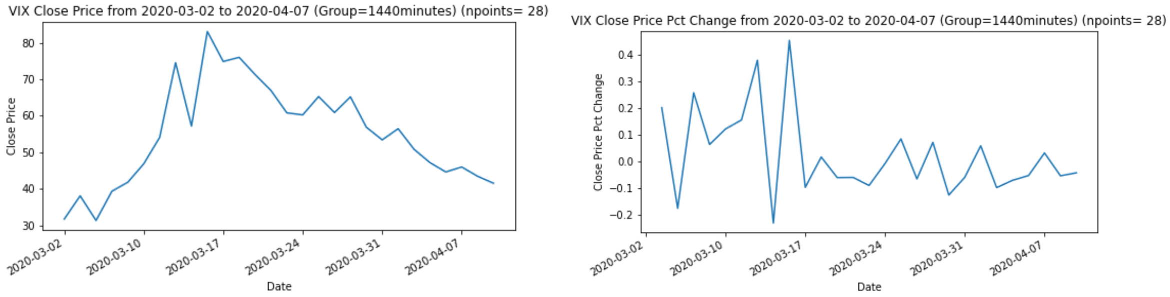


Figure 3.1.2. Trend showing VIX price over time (left) and VIX price change in percent over time (right). Both trends are taken from 2020-03-02 to 2020-04-07. Data is aggregated with a resolution of 1440 minutes, or 1 day equivalently. Missing points such as weekends are excluded to provide a smoother trend. The data captured on the left shows an increase in price followed by a slower decrease in price as markets reacted to COVID-19.

Quantity of Data

The relevant data collected from the two indexes was an aggregation of close price to determine percent change over defined temporal resolutions. The greatest number of data points occurred with smaller resolutions and the least were with higher resolutions. This occurred because, given a fixed time period to draw data from, as we increased resolution size the number of periods grouped by resolution that fit within the time period of collection decreased.

Number of Available Datapoints at Different Resolutions

	60 Minutes	180 Minutes	360 Minutes	720 Minutes	1440 Minutes
SPX Close Data Points	**920**	**307**	**154**	**78**	**39**
VIX Close Data Points	**920**	**307**	**154**	**78**	**39**
SPX Close Pct Diff Data Points	**919**	**306**	**153**	**77**	**38**
VIX Close Pct Diff Data Points	**919**	**306**	**153**	**77**	**38**

Figure 3.1.3. Number of data points available at different resolutions for SPX and VIX, where resolutions are column headers. We see that the number of points decreases as the resolution increases. Specifically, they exhibit an inverse relationship were doubling the resolution size halves the number of data points. As we increase our group by size, we decrease the number of datapoints available to us.

There is one less datapoint when looking at percent difference for obvious reasons, as the earliest date does not have a previous date to calculate percent change from.

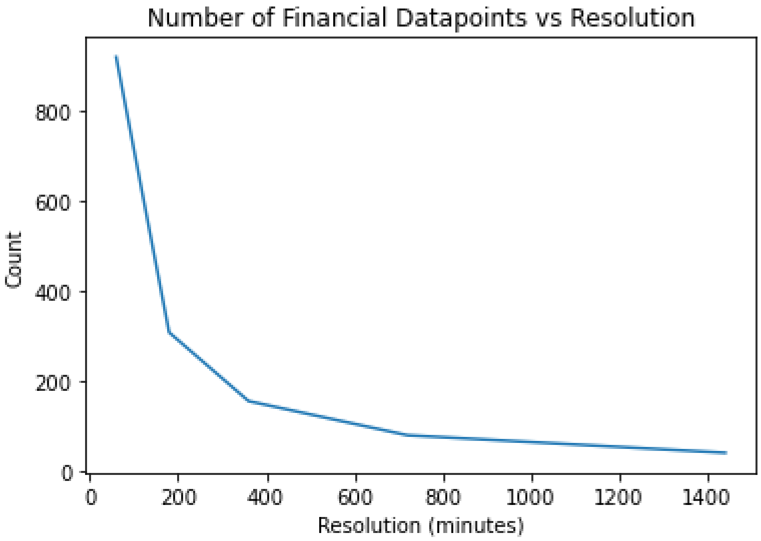
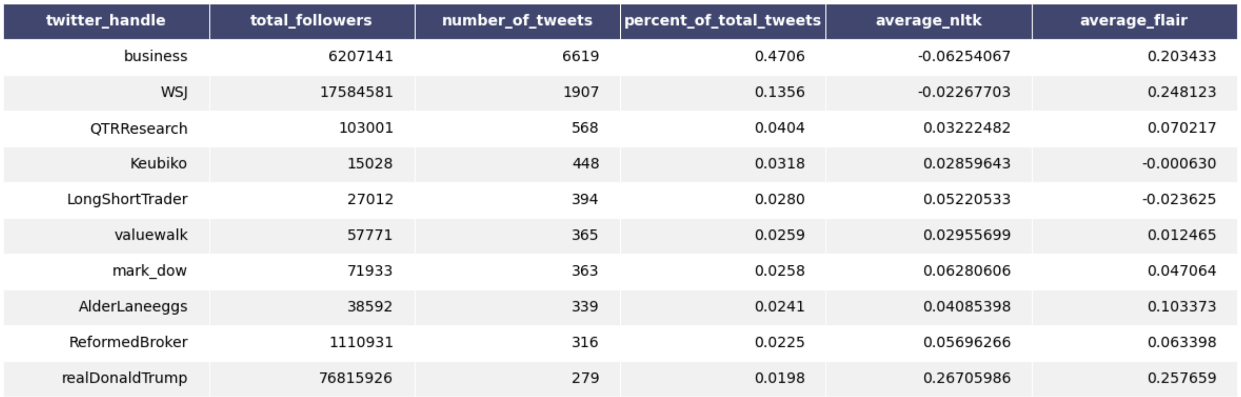


Figure 3.1.4. Trend showing number of financial data points over resolution sizes in minutes for SPX data. Because the number of points for SPX and VIX are so similar, it is redundant to plot both.

Twitter Data

There were 14066 tweets in total from 2020-02-29 to 2020-03-12 and 2020-03-28 to 2020-03-28. Due to technical issues regarding facility access, there were no tweets collected during a 17 day period from 2020-03-12 to 2020-03-28. Of the 14066 tweets, 13942 of them contained text content that was more than just a url. Thus the total number of usable tweets is 13942.

<u>Top 10 most active tweeters</u>



twitter_handle	total_followers	number_of_tweets	percent_of_total_tweets	average_nltk	average_flair
business	6207141	6619	0.4706	-0.06254067	0.203433
WSJ	17584581	1907	0.1356	-0.02267703	0.248123
QTRResearch	103001	568	0.0404	0.03222482	0.070217
Keubiko	15028	448	0.0318	0.02859643	-0.000630
LongShortTrader	27012	394	0.0280	0.05220533	-0.023625
valuewalk	57771	365	0.0259	0.02955699	0.012465
mark_dow	71933	363	0.0258	0.06280606	0.047064
AlderLaneeggs	38592	339	0.0241	0.04085398	0.103373
ReformedBroker	1110931	316	0.0225	0.05696266	0.063398
realDonaldTrump	76815926	279	0.0198	0.26705986	0.257659

Figure 3.2.1. Table of most active twitter accounts whose tweets were collected from, ordered by activity and cropped to show top 10. Other attributes per account are displayed such as average sentiment and number of tweets posted during the collection period.

Of the 93 twitter accounts that were streamed, 24 users did not make a single post. The remaining 69 accounts made at least one post. There was not a uniform distribution of accounts and postings, and some accounts posted much more frequently than others. Bloomberg @business and the Wall Street Journal @WSJ were the most active. Bloomberg took a commanding lead in tweet quantity constituting 47.06% of all tweets streamed. The Wall street Journal was next up with 13.56%. Donald Trump @realDonaldTrump, while having the most followers on this list, came in 10th for tweet quantity contributing 1.98% of all tweets streamed.

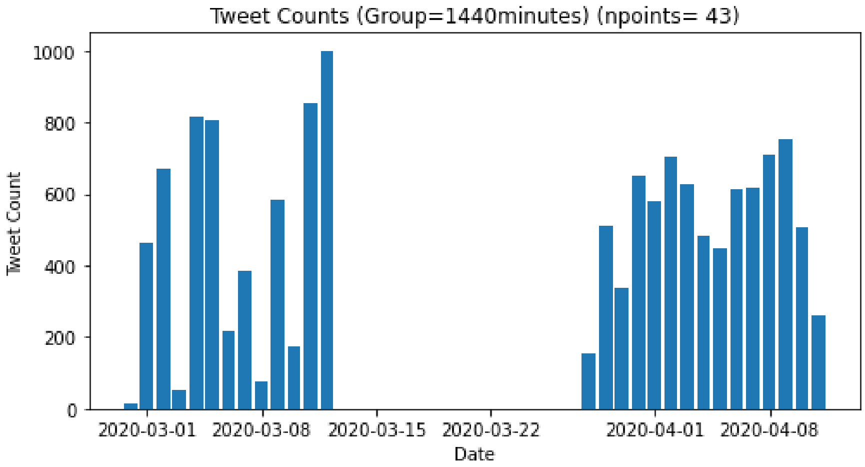


Figure 3.2.2. Histogram of number of tweets posted per day. There is a large gap with no tweets due to extraneous circumstances with data collection as a result of workplace restrictions imposed by the COVID-19 pandemic. Tweets are inconsistent in count in the earlier period because of programming bugs resulting in loss of data.

Due to other technical difficulties with the computer used for collecting tweets, there are some days in early March with fewer tweets streamed. However, each group was treated equally regardless of how many tweets it contained so long as that count was greater than zero.

	60 Minutes	180 Minutes	360 Minutes	720 Minutes	1440 Minutes
Tweet Data Points	1002	335	168	85	43

Figure 3.2.3. Number of data points available at different resolutions for tweets, where resolutions are column headers. We see that the number of points decreases as the resolution increases. Specifically, they exhibit an inverse relationship were doubling the resolution size halves the number of data points. As we increase our group by size, we decrease the number of datapoints available to us.

As with the financial data, the number of tweet data points available decreased as we increased the group size with an inverse relation.

Total Quantity of Useable Data Points

We concatenated the already grouped financial and tweet data along the datetime index of the dataframes. Due to insufficient market data or tweet data during certain periods such as weekends when markets were closed or gaps in tweet collection, some rows of this concatenated DataFrame contained null values. These rows could not be used in correlation testing and where dropped. The final quantity of data points achieved for each resolution was reduced by about a half from the quantity of raw data collected.

	60 Minutes	180 Minutes	360 Minutes	720 Minutes	1440 Minutes
Total Data Points	446	153	80	42	23

Figure 3.3.1. Number of data points from concatenated VIX, SPX, and tweet data. Resolutions are given as column headers. The number of total points is equivalent to the count of datetime indexes post-grouping where each field for VIX, SPX and sentiment score has a value not equal to nan. There are fewer total data points available than there are available for VIX, SPX or tweets individually. There is the

same decreasing relationship in quantity of points as resolution increases. An offset of zero was and larger offsets will decrease the total number of points available.

Twitter Sentiment

NLTK

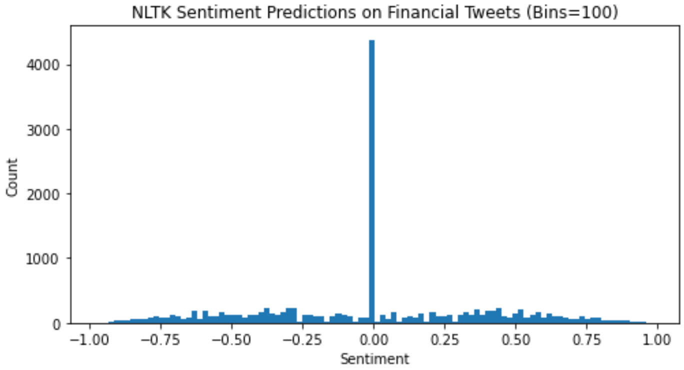


Figure 3.4.1. Histogram showing distribution of NLTK Sentiment predictions on collected data. A large number of NLTK sentiment tweets gravitated towards neutral predictions.

Upon plotting the distribution of sentiment for the streamed tweets, the bias NLTK has for labeling tweets as neutral is clear. The number of tweets predicted to be completely neutral in sentiment contains 31.0% of the probability mass.

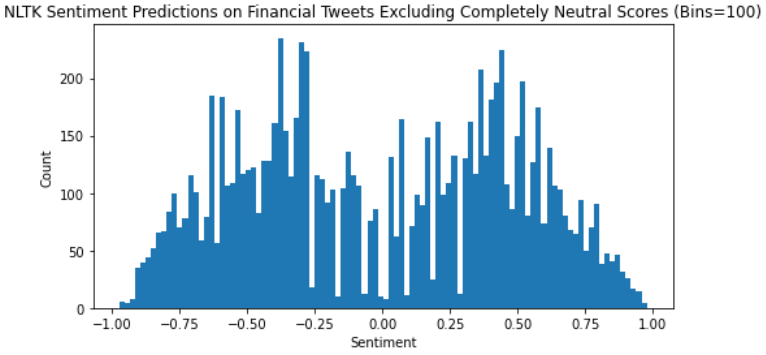


Figure 3.4.1. Histogram showing distribution of NLTK sentiment predictions on collected data, excluding predictions that gave completely neutral sentiment scores. The distribution seems to be bimodal in nature, although testing is required to make that claim.

By removing tweets with completely neutral predicted sentiment, a new distribution forms that seems to be bimodal in nature, although testing is required to support that claim. Predicted tweet sentiment seems to have two modes at approximately -0.3 and +0.4.

Completely neutral tweets, that is,tweets with sentiment scores of zero, were removed. Tweet sentiment was weighted by follower count and then averaged over different temporal resolutions. Plotting the average weighted sentiment over time with a resolution of 1 day, or 1440 minutes equivalently, shows the NLTK model sentiment prediction trend over time.

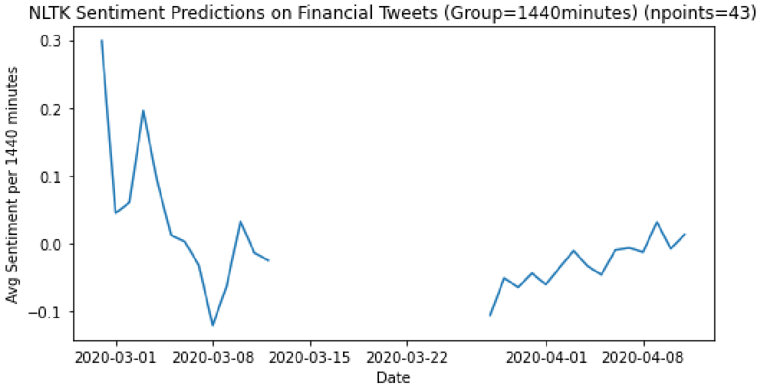


Figure 3.4.2. NLTK sentiment over time for collected tweets. There is a sharp drop in sentiment followed by, after gaps in data, a smaller increase in sentiment. This has resemblance to the SPX trend from Fig 3.1.1 (left).

Sentiment starts off very positive and decreases before the data interruption. The trend is more volatile before the break in data, although this volatility was not calculated. The increased volatility in the earlier segment of sentiment data is possibly due to the low quantity of tweets captured at those points (Fig 3.2.2). However, this claim cannot be made without first first determining the nature of the NLTK sentiment distribution on collected tweets and whether or not the average sentiment follows the Law of Large numbers. If the limit of average sentiment approaches a stable sentiment value as the number of tweets averaged increases, then that would explain why days such as 2020-02-28, 2020-03-03 and 2020-03-08 with relatively few tweets deviated the most as far as sentiment change over time.

On the other hand, the volatility of the later segment of data seems to be lower for sentiment change over time and the counts more consistently large (Fig 3.2.2).

FLAIR

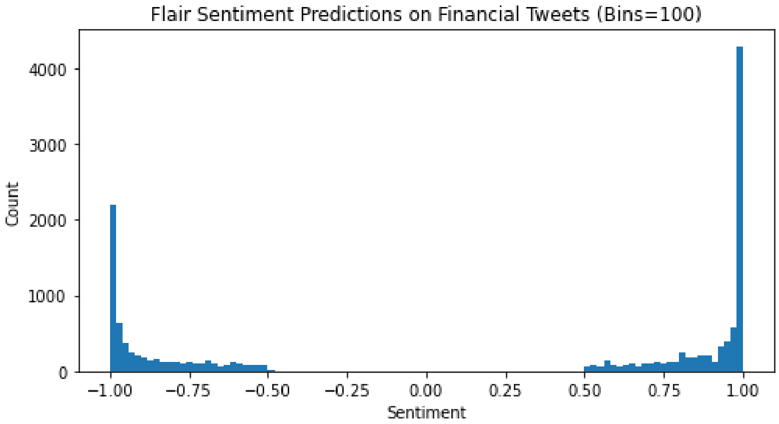


Figure 3.4.3. Histogram showing distribution of FLAIR sentiment from collected financial tweets. FLAIR exhibits a bias towards more extreme sentiment values, with bias towards positive predictions.

The flair model has a clear bias towards predicting extreme values for sentiment. Both ends of the distribution are modes in themselves as we can see the counts increase in a non-linear growth rate as we approach either -1.0 or +1.0. There are virtually no sentiment values with magnitude less than 0.50. Furthermore, the count of tweets with predicted sentiment of value greater than +0.98 is roughly double the size of the tweets with sentiment less than -0.98. Given an x-axis range of [-1.0+1.0] with 100 bins, each bin contains tweets within a .02 range.

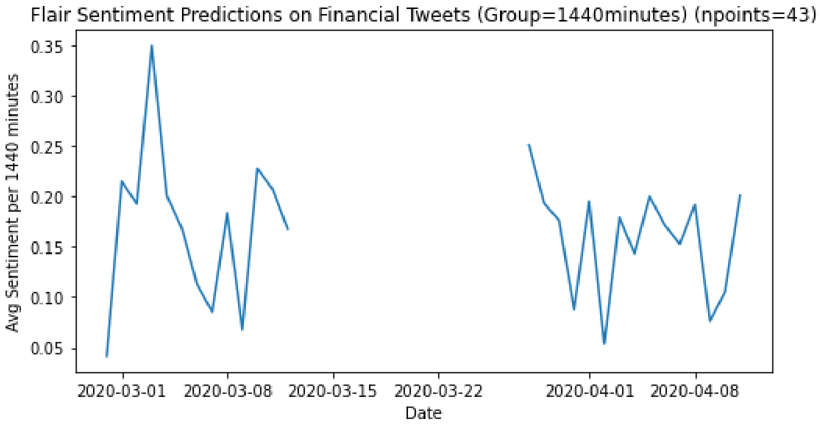
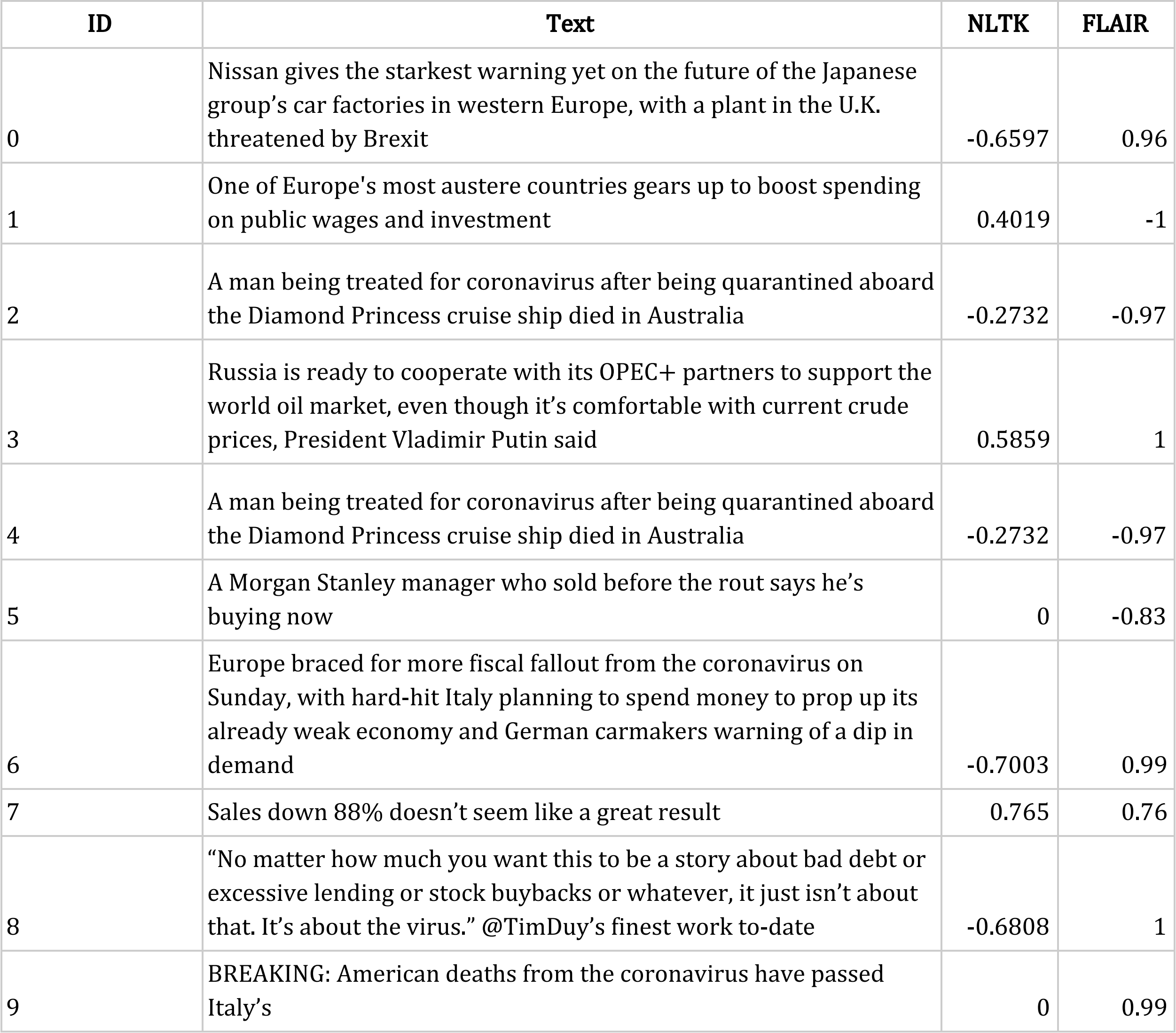


Figure 3.4.4. FLAIR sentiment over time for collected tweets.

When looking at sentiment over time, according to the flair model, the data is much more noisy and never goes below zero. This makes sense given how biased the flair model is towards extreme sentiment predictions and also its bias towards predicting positive sentiment.

Interesting Tweets

We compiled some sample tweets collected along with their sentiment scores from both models. The examples given in Fig 3.4.5 are cherry picked to demonstrate times when the models were in agreement or disagreement as well as illogical predictions.



ID	Text	NLTK	FLAIR
0	Nissan gives the starkest warning yet on the future of the Japanese group’s car factories in western Europe, with a plant in the U.K. threatened by Brexit	-0.6597	0.96
1	One of Europe's most austere countries gears up to boost spending on public wages and investment	0.4019	-1
2	A man being treated for coronavirus after being quarantined aboard the Diamond Princess cruise ship died in Australia	-0.2732	-0.97
3	Russia is ready to cooperate with its OPEC+ partners to support the world oil market, even though it’s comfortable with current crude prices, President Vladimir Putin said	0.5859	1
4	A man being treated for coronavirus after being quarantined aboard the Diamond Princess cruise ship died in Australia	-0.2732	-0.97
5	A Morgan Stanley manager who sold before the rout says he’s buying now	0	-0.83
6	Europe braced for more fiscal fallout from the coronavirus on Sunday, with hard-hit Italy planning to spend money to prop up its already weak economy and German carmakers warning of a dip in demand	-0.7003	0.99
7	Sales down 88% doesn’t seem like a great result	0.765	0.76
8	“No matter how much you want this to be a story about bad debt or excessive lending or stock buybacks or whatever, it just isn’t about that. It’s about the virus.” @TimDuy’s finest work to-date	-0.6808	1
9	BREAKING: American deaths from the coronavirus have passed Italy’s	0	0.99

Figure 3.4.5. Selection of tweets collected along with the sentiment scores predicted by both NLTK and FLAIR. These examples highlight instances where the two models agree and disagree, as well as some logical and nonsensical predictions.

We will not be going into each of these examples in depth, but from a high level it is apparent that FLAIR has some serious issues understanding tweet sentiment. If we look at the tweets of IDs 0, 6, and 9 we see that FLAIR labeled some obviously negative sentiment tweets as extremely positive. On the flipside, tweet 1 can be interpreted as mostly positive from a financial perspective because planned increases in spending and investing usually signify growth which is good. However, FLAIR predicted it had highly negative sentiment. Overall, FLAIR seemed to be performing poorly prior to validation.

Differences

An interesting observation to be made on the average sentiment for the first day is how different that value is between models. The nltk model has an extremely positive prediction and is in fact the highest average sentiment calculated for a day. On the other hand, the flair model hits its lowest average sentiment for a day going very close to zero. Because the number of tweets for this first day is so small, that sort of outcome is possible. However, as the number of tweets in a day increases so should the flair model's predicted sentiment.

Results

We correlated Twitter sentiment and index movement using different temporal resolutions to data. We looked at the correlation between data points with different temporal offsets. We selected the NLTK model to perform sentiment predictions on collected tweets. We removed all tweets with neutral sentiment predictions. We used Pearson and Spearman correlation testing to generate r-values. We tested significance with the Pearson r-values but did not test significance for Spearman. Our significance threshold was with a 95% confidence interval. The most, and only, significant correlation we observed was between sentiment and VIX movement with a resolution of 360 minutes at an offset of 1800 minutes.

Overview

Looking at the Financial Indexes price movement and average weighted sentiment over time we cannot immediately spot obvious trend relationships between sentiment and price movement. The data is plotted separately because of a large gap in tweets from 2020-03-13 to 2020-03-27. However, when performing correlation testing both portions of the data were used.

We only run correlation testing with temporal offsets that either correspond data at the same time or correspond future financial data with Twitter data. We do not run any analysis looking at offsets that would correspond future Twitter sentiment with financial index price movements. The reasoning behind this was because we were initially searching for correlations that would help predict market movement. However, this approach has taken away from other insights that could have been made about other relevant relationships.

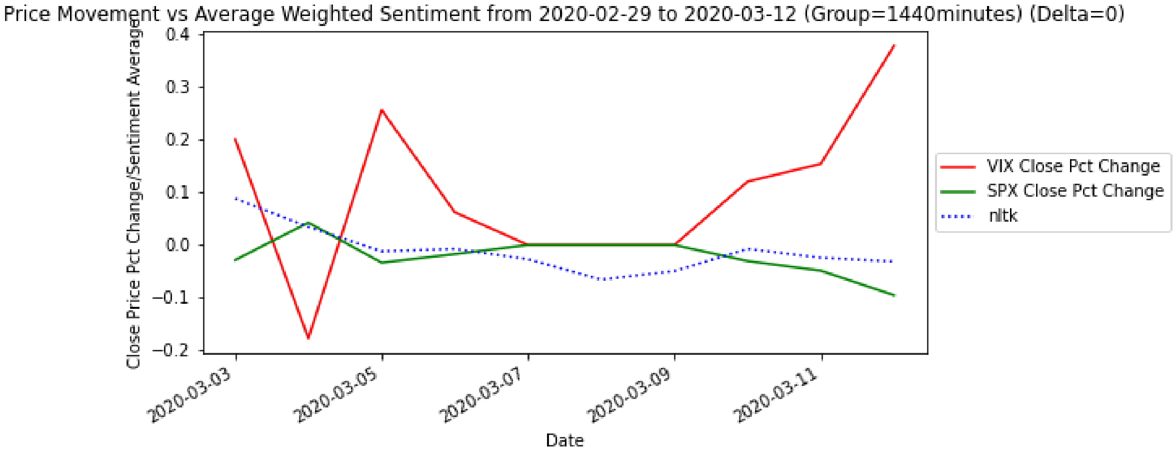


Figure 4.1.1. NLTK sentiment, VIX and SPX price movement over time during 2020-02-29 to 2020-03-12, the period of time before the gap in twitter data.

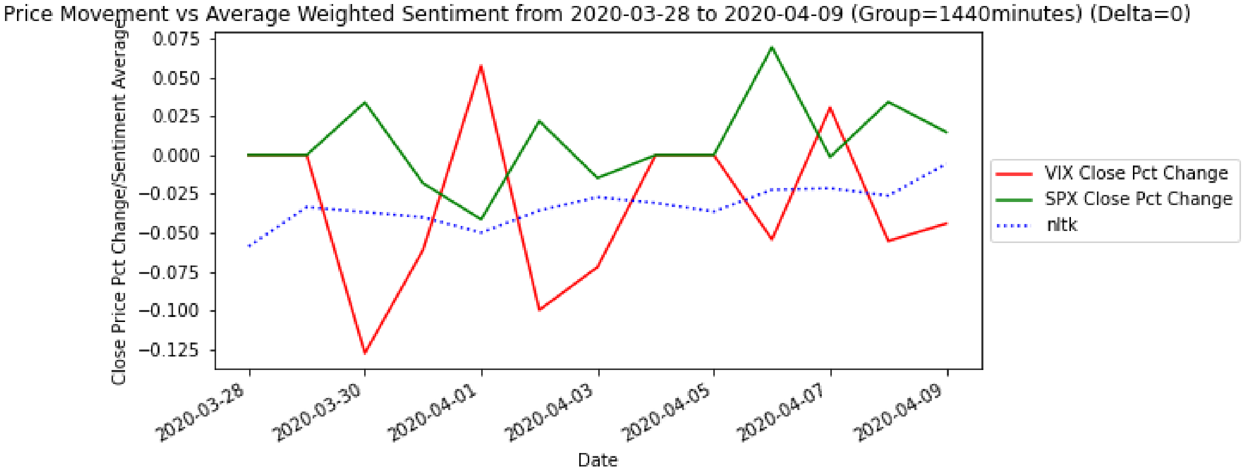


Figure 4.1.2. NLTK sentiment, VIX and SPX price movement over time during 2020-03-28 to 2020-04-09, the period of time after the gap in twitter data. Shown separately from Fig 4.1.1 to display higher resolution without gap.

NLTK vs SPX

We found that given an absence of temporal offset of zero there was no correlation between NLTK sentiment and SPX movement for all temporal resolutions used. The absolute value of correlation coefficients calculated using both Pearson and Spearman testing was less than 0.15 for all resolutions with a zero offset and p-values were greater than 0.05.

Pearson Correlations for SPX Percent Change and NLTK Average Weighted Sentiment *(* :== p-value > 0.05)*

Pearson SPX:NLTK	60 Minutes	180	360	720	1440
0	0.025	0.058	0.107	0.082	0.00
1	-0.001	0.115	0.093	0.080	-0.004
2	0.088	0.133	0.185	0.070	-0.118
3	-0.055	-0.032	-0.130	-0.138	NaN
4	0.028	-0.017	0.080	-0.006	NaN
5	0.034	0.093	-0.140	0.031	NaN

Figure 4.2.1. Table showing Pearson correlation values and statistical significance between NLTK average weighted sentiment and SPX percent change over a range of resolutions and offsets. Resolutions are given as column headers and are all measured in minutes. The row labels are coefficients that when multiplied by a resolution provide the corresponding offset in minutes. For example, at resolution=60 and offset=0 the r-value is 0.025. At resolution=60 and offset=300 the r-value is 0.034. Some cells are NaN for when there were not enough data points to provide an r-value. Statistically significant r-values within a 95% confidence are denoted with an asterisk.

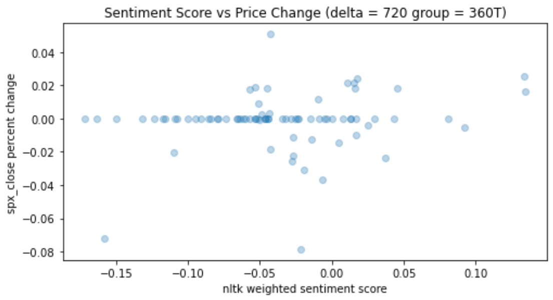
Pearson correlation coefficients measure the linear relationship between two variables through generating r-values. Whether or not this value is actually significant is determined by the number of points used to calculate the r-value as well as the r-value itself. Larger magnitude r-values and larger quantities of data result in increased significance. We did not observe any significant correlations between SPX and NLTK using Pearson's method.

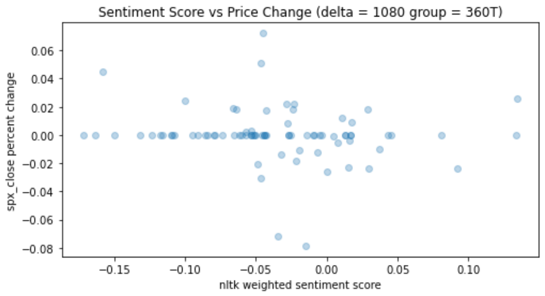
Spearman Correlations for SPX Percent Change and NLTK Average Weighted Sentiment

Spearman SPX:NLTK	60 Minutes	180	360	720	1440
0	0.004	0.006	0.020	0.113	-0.050
1	0.013	0.144	0.123	0.039	-0.126
2	0.035	0.122	0.057	-0.095	0.085
3	-0.048	-0.020	-0.211	-0.127	NaN
4	-0.023	-0.083	-0.089	-0.106	NaN
5	0.056	0.038	-0.129	0.111	NaN

Figure 4.2.1. Table showing Spearman correlation values and statistical significance between NLTK average weighted sentiment and SPX percent change over a range of resolutions and offsets. Resolutions are given as column headers and are all measured in minutes. The row labels are coefficients that when multiplied by a resolution provide the corresponding offset in minutes.

Spearman correlation coefficients measure the monotonic relationship between two variables by generating r-values, that is to say, they measure the change of values without caring so much about the rate of that change. Typically coefficients that round to -0.3 show that there is a small negative monotonic relationship. In our constructed analysis, we found one instance at resolution=360mins, offset=1080mins where we obtained a Spearman coefficient that rounded to -0.3. However, we cannot claim that this correlation is statistically significant as we failed to generate a p-value due to extraneous reasons.





VIX vs NLTK

We found that given an offset of 0 minutes, there was an observed correlation between VIX movement and NLTK for both Pearson and Spearman at resolution=1440mins. This was the only correlation found with a zero offset. However, because neither of these r-values had established significance, we cannot make claims that there is correlation at this zero offset. However, we did observe a significant correlation of small positive linear correlation using Pearson's method at resolution=360 minutes, offset=1800 minutes. This was the only significant r-value recorded from this research.

Pearson Correlations for VIX Percent Change and NLTK Average Weighted Sentiment _(* :== p-value > 0.05)_

Pearson VIX:NLTK	60 Minutes	180	360	720	1440
0	-0.066	-0.053	-0.060	0.021	0.179
1	0.046	-0.044	-0.061	-0.031	0.107
2	-0.085	-0.097	-0.161	0.086	0.333
3	0.033	0.053	0.197	0.278	NaN
4	0.034	0.039	-0.051	0.117	NaN
5	-0.063	-0.013	0.250*	0.126	NaN

Figure 4.3.1. Table showing Pearson correlation values and statistical significance between NLTK average weighted sentiment and VIX percent change over a range of resolutions and offsets. Resolutions are given as column headers and are all measured in minutes. The row labels are coefficients that when multiplied by a resolution provide the corresponding offset in minutes. Statistically significant r-values within a 95% confidence are denoted with an asterisk.

The most interesting correlation value recorded for all cross-correlation testing was observed between VIX and NLTK with a resolution of 360 minutes with an offset of 1800 minutes. The correlation measured was statistically significant with a p-value less than 0.05 allowing us to say with 95% confidence that there is a small positive correlation.

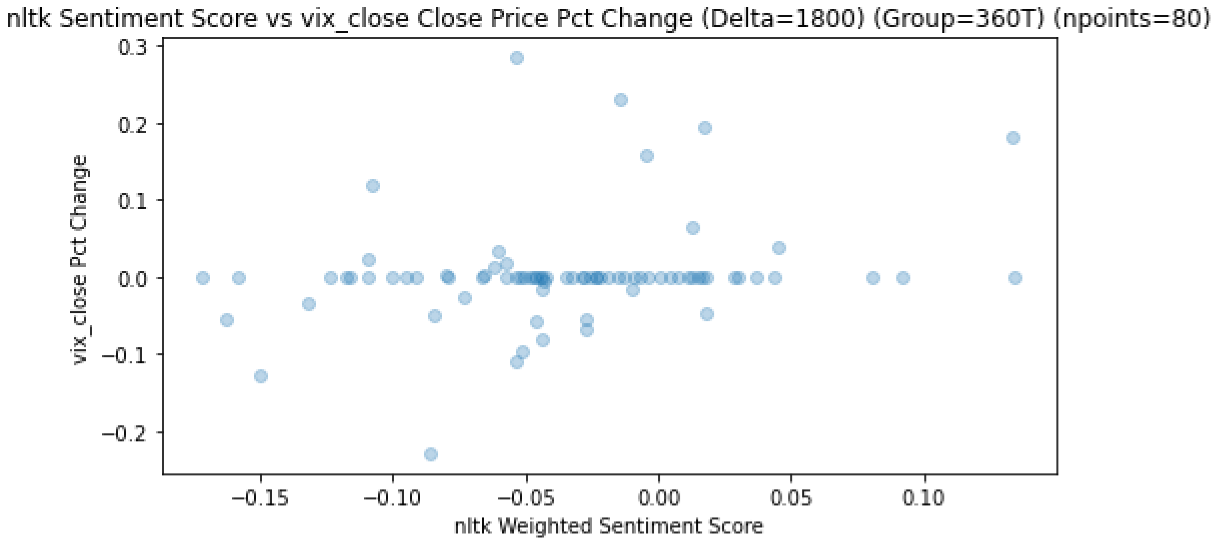


Figure 4.3.2. Scatter plot of VIX percent change and NLTK average weighted sentiment score with resolution=360 minutes and offset=1800 minutes. This plot was included because this resolution and offset provided statistically significant small positive correlation. Most of the VIX percent changes are around zero.

Spearman VIX:NLTK	60 Minutes	180	360	720	1440
0	-0.003	-0.014	-0.041	-0.074	0.199
1	0.098	-0.064	-0.096	0.008	0.286
2	0.048	-0.059	-0.063	0.162	0.190
3	0.093	0.016	0.197	0.215	NaN
4	0.101	0.043	0.149	0.205	NaN
5	-0.023	-0.034	0.163	-0.026	NaN

Figure 4.3.3. Table showing Spearman correlation values and statistical significance between NLTK average weighted sentiment and VIX percent change over a range of resolutions and offsets. Resolutions are given as column headers and are all measured in minutes. The row labels are coefficients that when multiplied by a resolution provide the corresponding offset in minutes.

Spearman correlation showed numerous points where the VIX movement and NLTK sentiment were positively correlated. In all recorded instances of resolution=1440 a small positive correlation was observed. Correlations were observed with larger resolutions and with a few different offsets for those resolutions. It would be worthwhile to test for significance to determine if these r-values do in fact show correlation.

The interpretation that can be drawn from this testing is that there was a significant small positive monotonic relationship between VIX movement and NLTK sentiment when grouping by 360minutes and offsetting by 1800 minutes.

Noteworthy Resolutions

Below are plots for the resolution 360 minutes describing the correlation between both NLTK and SPX as well as NLTK and VIX. The plots show changes in correlation as offset changes. An interesting observation can be made regarding the relationship between correlation of NLTK and VIX as well as NLTK and SPX for resolution of 360 minutes. The two move in opposite directions with respect to changing offset values. This occurs at every offset for both Pearson and Spearman correlations.

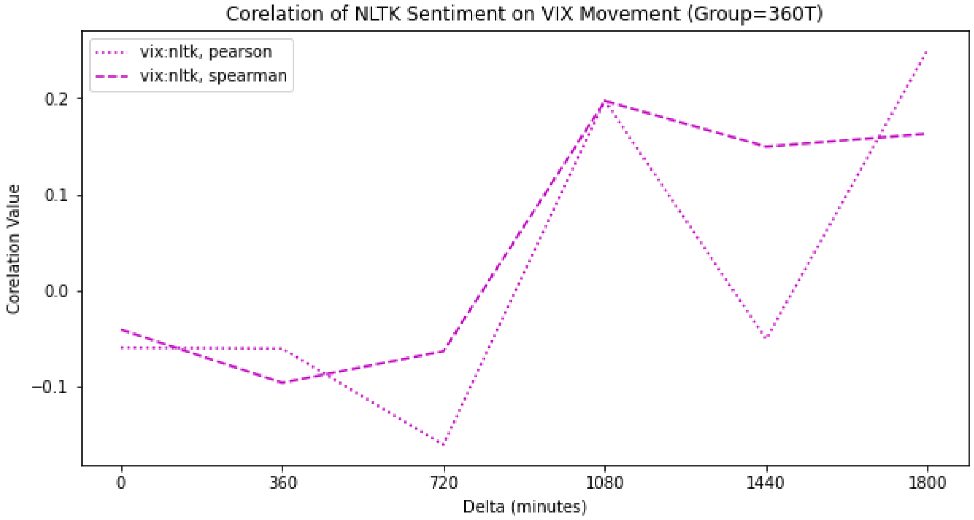


Figure 4.3.4. Plotted correlation over changing offsets for NLTK vs VIX with resolution 360 minutes. Pearson correlation increases and oscillates as offset increases. Statistically significant correlation is observed at 1800.

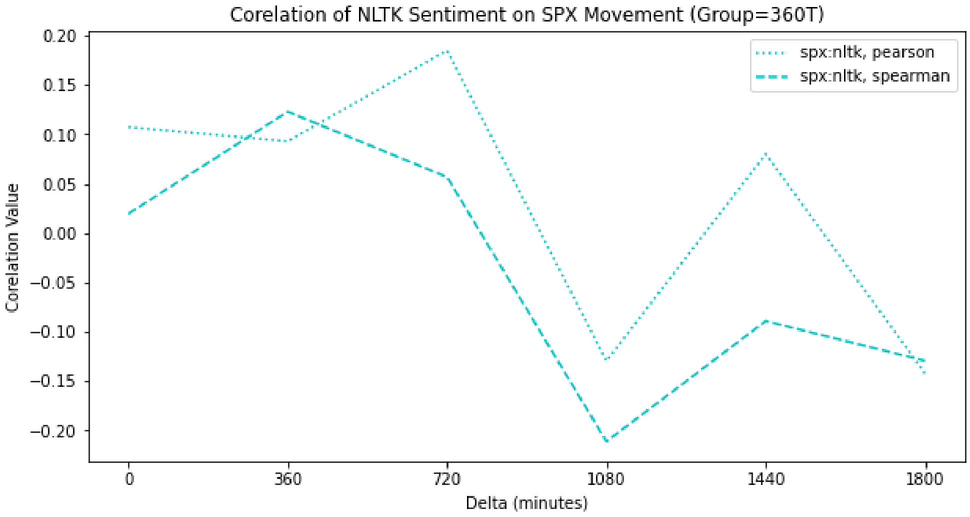


Figure 4.3.5. Plotted correlation over changing offsets for NLTK vs SPX with resolution 360 minutes. Pearson correlation decreases and oscillates as offset increases. Has inverse relationship to Fig 4.3.4.

Discussion

There is an age old saying that correlation is not causation. Pearson correlation testing does not measure the extent of how much one variable will influence another. Pearson correlation testing measures how well a linear function fits the relationship between two variables. When there is high correlation that is statistically significant, it is possible to predict the value of one variable using the other. However, even in that scenario causation can not be claimed to exist between the two variables. For instance, while we might find a high correlation between clams consumed in a year and the number of redheads born it would be scientifically based to suggest that there is a cause-effect relationship present. Even if we repeat this statistical experiment over the course of one thousand years and find the results stick we could never claim this as evidence for causation. In the words of Judith Butler, "causation for Pearson is only a matter of repetition, and in the deterministic sense can never be proven" [8].

Our own experiment of looking at the relationship between Twitter sentiment and market movement made no attempt to prove that twitter sentiment causes changes in the market. We instead examined the linear relationship these two sets of data shared. We were able to find a single combination of offsets and resolutions that demonstrated a statistically significant linear correlation between sentiment and VIX movement with 95% confidence This correlation we found between sentiment and VIX seems to contradict the findings of Rao. Rao found there is a large correlation between DJIA and bullishness sentiment with 95% confidence using Pearson's method. However, given that the DJIA and SPX typically move together and SPX and VIX move in opposite directions it should be the case that if positive sentiment is positively correlated with DJIA movement, as was observed [rao], then positive sentiment should be negatively correlated with VIX movement. However, our results contradict this reasoning. If we were to conduct a future study using the research done in this paper, we would focus on that combination with resolution 360 minutes and offset 1800 minutes. It would be worthwhile to repeat the experiment with more data points and during more normal market conditions to see if the correlation holds.

Market conditions were abnormal during the collection phase of our research due to the COVID-19's transition from Chinese epidemic to historic pandemic. On February 19th 2020 the SPX hit an all time high of 3,393.52 points. Over the next 22 days the SPX dropped 30% in value to a 160-week low. This was the fastest 30% drop from record highs the S&P 500 has ever experienced, with 1934, 1931, and 1929 being the second, third, and fourth fastest 30% drops respectively at 23 days, 24 days, and 31 days. Even the infamous 2008 recession took 250 days to achieve a 30% drop from its record high. This goes to show how rare and historic the SPX movement we recorded was. The VIX, which typically moves in the opposite direction to the SPX, experienced similar movement in regards to extremity. Because of the rarity of the movement we recorded in our financial indexes, it is obvious that our correlations can not be said to hold for general market conditions.

Unfortunately we weren't able to gather tweets for part of the drop. COVID-19 affected the financial data we collected but also the logistics in methodology used to collect tweets. We were using a computer in a public facility on campus before strict regulations were put in place to prevent the spread of disease. With no warning, the facility we were using became inaccessible and we had no idea if our computer was still streaming tweets. After about two weeks we were able to transport the computer to a private dormitory to allow for tweet collection to resume. It was discovered that the machine restarted during this period of inaccessibility and no tweets were collected between that restart and transportation. This could have been easily fixed if we had published the tweet-streaming code on GitHub. We could have streamed tweets from a different machine if those precautions were taken prior to facility lockdown. However, another fear we had was that if we streamed using the same Twitter API key on another machine then Twitter might revoke the API key. Twitter prohibits using a distinct API key with two distinct server connections. Breaking this rule can result in the API key being deactivated, along with Twitter Developer privileges being revoked. Because of this fear and code not uploaded to GitHub, we missed out on valuable tweet data.

COVID-19 impacted almost every aspect of this senior project. Investor reactions to COVID-19 created incredibly anomalous financial data. Federal and state response to prevent spread resulted in non-essential public facilities being closed and, along with lack of backing up code, prevented over two weeks of Twitter data from being streamed midway through the data collection process of this research.

Aside from the obstacles we faced in logistical tweet collection, we are very satisfied with how the data extraction was conducted. We chose to deviate from similar studies such as Rao and Bollen in the demographic streamed on Twitter. We made no attempt to generate sentiment representing the greater public and focused on only tweets posted by accounts with high influence in the financial world according to an empirical study conducted by Forbes in 2018 [9]. Part of the reason we chose this different approach was because we wanted to distinguish our research from Bollen and Rao. Looking at our results through this perspective, it would be interesting to compare the correlations against correlations composed of public sentiment expressed on Twitter without the account limitation. Future work could be done to compare correlations that public sentiment and financial influencer sentiment have against financial data collected during a consistent time period. It would be interesting to see if the sentiment expressed by financial users on Twitter has a stronger relationship with SPX and VIX price movement than the aggregated sentiment of random Twitter users. That is to say, do the sentiments expressed on Twitter by financial influencers like Warren Buffet and John Hempton have a stronger relationship with stock market price movement than the sentiments expressed by a larger and more random collection of Twitter users? This could be a very interesting study indeed and could be expanded to compare results using many user demographics.

When we were deciding on how to measure price change, we referred to the methods used by Rao. While we did not incorporate logarithms, we decided that the magnitude of price change depended not only on the magnitude of point change but rather that the magnitude of price change relative to the initial price. Thus we settled for percent change as our metric. We considered and rejected Bollen's approach of using point change to calculate returns. We rejected this approach because we felt that it did not accurately capture market movement. This is illustrated in the following example. Suppose Goldman Sachs has $100 million invested in the VIX under each of the two cases: (A) VIX is priced at $10.00, (B) VIX is priced at $50.00. In case (A) we know that Goldman Sachs has 10 million shares of VIX. In case (B) they have 2 million shares. Now suppose that in both cases the VIX increases in value by $5.00 per share. Goldman Sachs made profits of $50 million and $10 million from cases (A) and (B) respectively.

Under Bollen's approach of point change, the price change of each case is both equal to $5.00. However, from the perspective of Goldman Sachs, the returns from case (A) far surpass the returns from case (B). On the other hand, using a percent change approach we find that the price changes derived from case (A) and (B) are 150% and 110% respectively. This example illuminates the advantage of using a dynamic approach like percent change against a static approach like point change. Percent change captures a metric whose magnitude more accurately reflects the perspective of investors regarding price changes.

With regard to program structure, we succeeded in modularizing the codebase. This modularization makes the structure easier to understand and allows codebase modules to be re-used for future projects. However, there was a massive oversight when developing the modules. We failed to create automated tests to assure that each module was operating correctly. Testing was done manually but this is not sufficient for quality assurance. Due to the number of modules and their interdependency, it was imperative that unit testing could be run for all modules and their methods after any refactors. Because we failed to implement automated testing, at each point of refactoring a module there was a risk that another module it was imported by could have started operating differently in unexpected ways. Furthermore, this project should not be considered to be fully operational because there is no evidence that everything is working as it should or as it was at a previous point before refactor.

Ultimately, we learned a lot throughout this process. We developed valuable proficiency creating an ETL framework. We chose to code in Python and further honed our skills using libraries such as Pandas and working with modules. We were able to put lessons learned in Computational Statistics to use with correlation and significance testing. Finally, we had the opportunity to pursue self-guided research in a way that we have never done before. Many things were unfamiliar to us throughout this process and looking back, we wish that we had referred more frequently to the related works throughout implementation. We performed the background research and then moved onto implementation afterward. While refactoring the code we never went back to the background research to inform the current implementation. What we should have done was sandwiched every refactor with consolidation from our related work. A huge oversight when it came to processing tweets was that user handles weren't removed. Referring to Bollen before refactoring to remove url links would have made this preprocessing omission obvious and created more meaningful FLAIR generated sentiment scores, since FLAIR was mishandling usernames contained in tweet text.

All things considered, this project has value for future research. We talked about how different twitter demographics could be studied. After implementing automated testing for the code and reworking sentiment weighting/normalization for aggregated tweets, it would be valuable to further delve into the correlations and determine whether tweets could be used to predict market movement. From there, one could implement a trading strategy and backtest the predictive capabilities on historical market prices. Also we would want to examine the relations between sentiment and market movement with an offset in both directions. Throughout this study we never considered to look at the relationship between relatively future tweets and present price changes. It is worth studying the offset in both directions in order to better understand the relation.

Conclusion

We found only one significant correlation and that was between average weighted sentiment and VIX price movement. This correlation was observed with resolutions of 360 minutes at an offset of 1800 minutes and was generated using Pearson's method. The correlation was significant and showed a small positive linear relation between sentiment and VIX price movement.

It is quite interesting that twitter sentiment was positively correlated with VIX price movement given how VIX is supposed to be the "Fear Indicator" and positive sentiment is not usually associated with fear. While counter-intuitive we found that with a resolution of 360 minutes, or six hours, the average weighted sentiment had a small positive monotonic linear relationship with VIX close price percent change.

Work Cited

[1] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. Flair: An easy-to-use framework for state-of-the-art nlp. In NAACL, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics: System Demonstrations

[2] Bollen, J., Mao, H. and Zeng, X. Twitter mood predicts the stock market. J. Computational Science 2, 1 (2011), 1–8.

[3] Cboe, (2003). “The Cboe Volatility Index - VIX”, Cboe White Paper, 2003.

[4] Gilbert, Eric & Karahalios, Karrie. (2010). Widespread Worry and the Stock Market.. ICWSM 2010 - Proceedings of the 4th International AAAI Conference on Weblogs and Social Media.

[5] Hirshleifer, David A., and Tyler Shumway. “Good Day Sunshine: Stock Returns and the Weather.” *SSRN Electronic Journal*, 2001, doi:10.2139/ssrn.265674.

[6] Lerner, Jennifer S., et al. “Heart Strings and Purse Strings. Carryover Effects of Emotions on Economic Decisions.” *Psychological Science*, vol. 15, no. 5, 2004, pp. 337–341., doi:10.1111/j.0956-7976.2004.00679.x.

[7] Li, Yun. “This Was the Fastest 30% Sell-off Ever, Exceeding the Pace of Declines during the Great Depression.” *CNBC*, CNBC, 23 Mar. 2020, www.cnbc.com/2020/03/23/this-was-the-fastest-30percent-stock-market-decline-ever.html.

[8] Pearl, Judea. “GALTON AND THE ABANDONED QUEST.” *The Book of Why: The New Science of Cause and Effect*, Basic Books, 2018, pp. 72–72.

[9] Shah, Alap. “The 100 Best Finance Twitter Accounts You Should Be Following.” *Forbes*, Forbes Magazine, 17 Nov. 2017, www.forbes.com/sites/alapshah/2017/11/16/the-100-best-twitter-accounts-for-finance/#24ba8baa7ea0.

[10] Rao, Tushar, and Saket Srivastava. "Analyzing stock market movements using twitter sentiment analysis." Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012). IEEE Computer Society, 2012.

[11] “The 2014 #YearOnTwitter.” *Twitter*, Twitter, 2014, blog.twitter.com/official/en_us/a/2014/the-2014-yearontwitter.html.