Amy DeMorrow

ECENGR 219 Large Scale Data Mining: Models and Algorithms

6/7/2024

End to End ML Pipeline:

Time-Series Correlation between Superbowl Scores and Tweets

Describe your task

I chose the Time-Series Correlation Between Scores and Tweets. I created a game log CSV file from the ESPN website link: https://www.espn.com/nfl/playbyplay/_/gameId/400749027. I aligned the tweets metadata with that Superbowl game event log and used the combined dataset to predict, given a tweet, which team is winning the football game. I also created a generative model to generate a sample tweet from the Superbowl using a sample game score.

Explore the data

Report for each hashtag:

• Average number of tweets per hour, average number of followers of users posting the tweets per tweet, and average number of retweets per tweet

```
# function to calculate statistics on each hashtag
def calculate_statistics(df):
    hashtags = df['hashtag'].unique()
    stats = []

    for hashtag in hashtags:
        hashtag_df = df[df['hashtag'] == hashtag]

        # set 'time_posted' as the index for resampling
        hashtag_df.set_index('time_posted', inplace=True)
        # resample by hour (H) and count the number of tweets per hour
        tweets_per_hour = hashtag_df.resample('H').size().mean()
        avg_followers_per_tweet = hashtag_df['followers'].mean()
        avg_retweets_per_tweet = hashtag_df['retweets'].mean()
        total_tweets = hashtag_df.shape[0]

        stats.append({
            'hashtag': hashtag,
            'avg_tweets_per_hour': tweets_per_hour,
            'avg_followers_per_tweet': avg_followers_per_tweet,
            'avg_retweets_per_tweet': avg_retweets_per_tweet,
            'total_tweets': total_tweets
        })

    return stats
```

Suggested code may be subject to a license | yuhaoyin/UCLA-20W-ECE219-LargeScaleDataMining

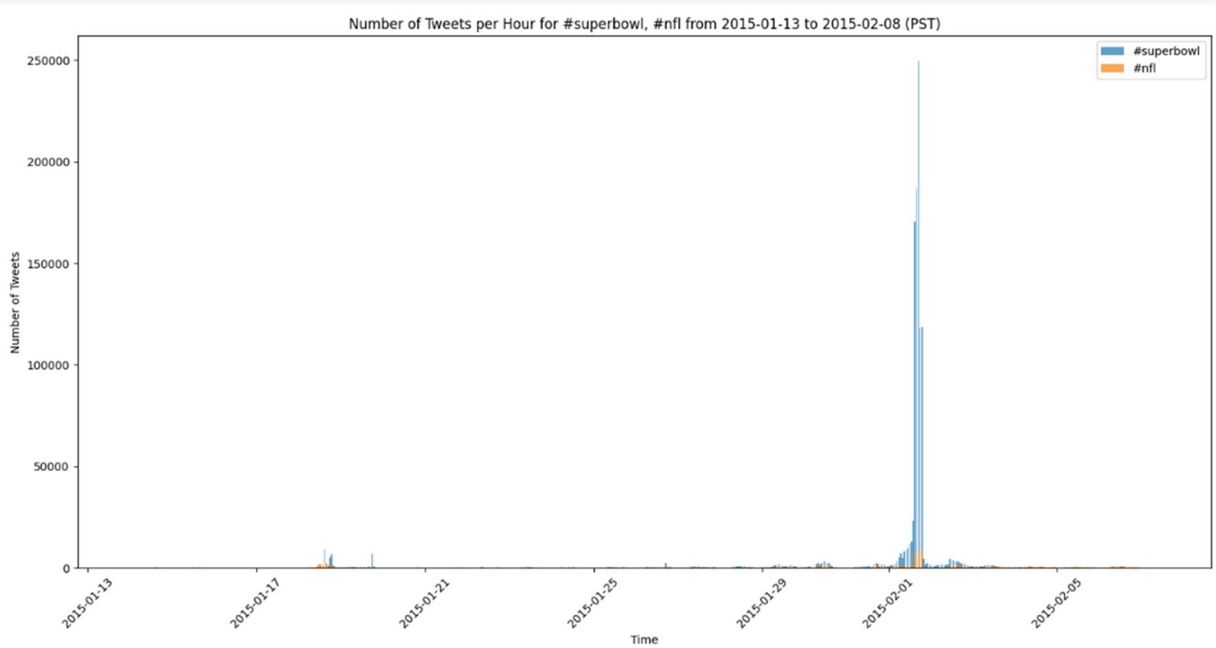
```python
# calculate statistics and print results
statistics = calculate_statistics(df)
for stat in statistics:
    print(f"Hashtag: {stat['hashtag']}")
    print(f"Average number of tweets per hour: {stat['avg_tweets_per_hour']:.2f}")
    print(f"Average number of followers per tweet: {stat['avg_followers_per_tweet']:.2f}")
    print(f"Average number of retweets per tweet: {stat['avg_retweets_per_tweet']:.2f}")
    print(f"Total tweets: {stat['total_tweets']}\n")
```

```
Hashtag: gohawks
Average number of tweets per hour: 292.09
Average number of followers per tweet: 2217.92
Average number of retweets per tweet: 2.01
Total tweets: 169122

Hashtag: gopatriots
Average number of tweets per hour: 40.89
Average number of followers per tweet: 1427.25
Average number of retweets per tweet: 1.41
Total tweets: 23511

Hashtag: nfl
Average number of tweets per hour: 396.97
Average number of followers per tweet: 4662.38
Average number of retweets per tweet: 1.53
Total tweets: 233022

Hashtag: patriots
Average number of tweets per hour: 750.63
Average number of followers per tweet: 3280.46
Average number of retweets per tweet: 1.79
Total tweets: 440621

Hashtag: sb49
Average number of tweets per hour: 1275.56
Average number of followers per tweet: 10374.16
Average number of retweets per tweet: 2.53
Total tweets: 743649

Hashtag: superbowl
Average number of tweets per hour: 2067.82
Average number of followers per tweet: 8814.97
Average number of retweets per tweet: 2.39
Total tweets: 1213813
```

Plot “number of tweets in hour” over time for #SuperBowl and #NFL.

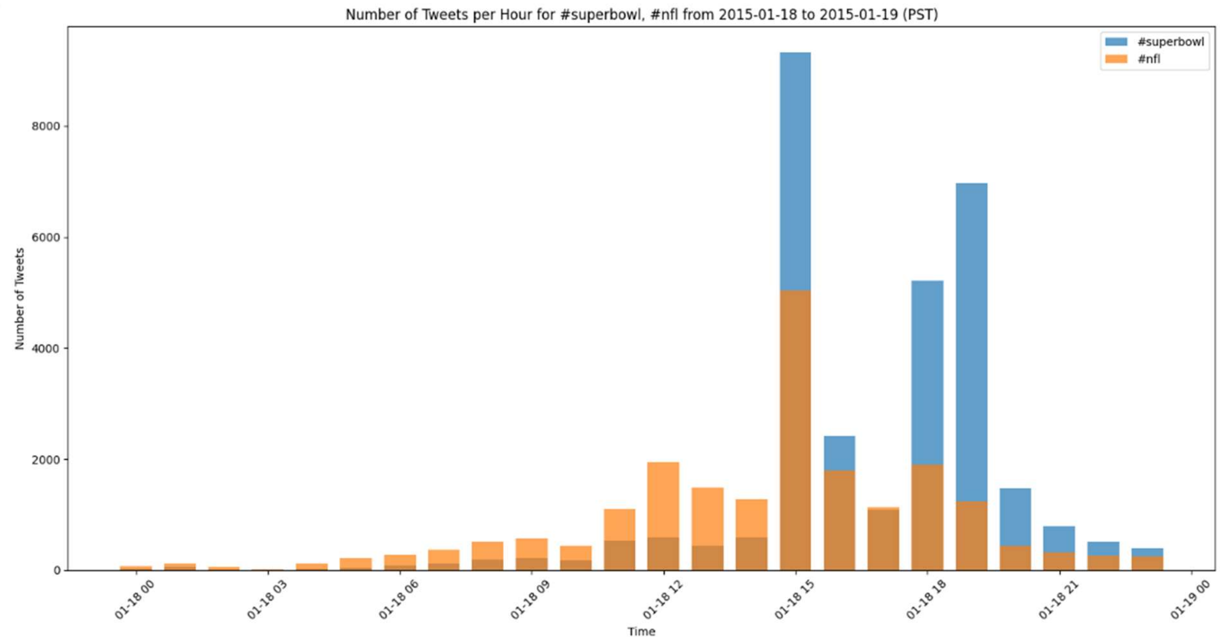
```python
# function to filter, resample, and plot by date and hashtag
def filter_resample_plot(df, start_date_str, end_date_str, hashtags):
    """
    Parameters:
    start_date_str (str): The start date in 'YYYY-MM-DD' format.
    end_date_str (str): The end date in 'YYYY-MM-DD' format.
    hashtags (list): List of hashtags to filter and plot.
    """
    # convert start and end dates to datetime with timezone
    seattle_tz = pytz.timezone('US/Pacific')
    start_date = seattle_tz.localize(datetime.strptime(start_date_str, '%Y-%m-%d'))
    end_date = seattle_tz.localize(datetime.strptime(end_date_str, '%Y-%m-%d'))

    # filter data for the specific date range in PST timezone
    filtered_df = df[(df['time_posted'] >= start_date) & (df['time_posted'] < end_date)]

    plt.figure(figsize=(15, 8))

    # iterate over each hashtag and plot the resampled data
    for hashtag in hashtags:
        hashtag_df = filtered_df[filtered_df['hashtag'] == hashtag]

        # resample data to 1-hour bins
        hashtag_df.set_index('time_posted', inplace=True)
        resampled = hashtag_df.resample('H').size()

        # plot
        plt.bar(resampled.index, resampled, width=0.03, label=f'#{hashtag}', alpha=0.7)

    # Formatting the plot
    plt.xlabel('Time')
    plt.ylabel('Number of Tweets')
    plt.title(f'Number of Tweets per Hour for {", ".join(["#" + h for h in hashtags])} from {start_date_str} to {end_date_str} (PST)')
    plt.legend()
    plt.xticks(rotation=45)
    plt.tight_layout()

    plt.show()
```

```python
# lets look at the entire time span
filter_resample_plot(df, '2015-01-13', '2015-02-08', ['superbowl', 'nfl'])
```



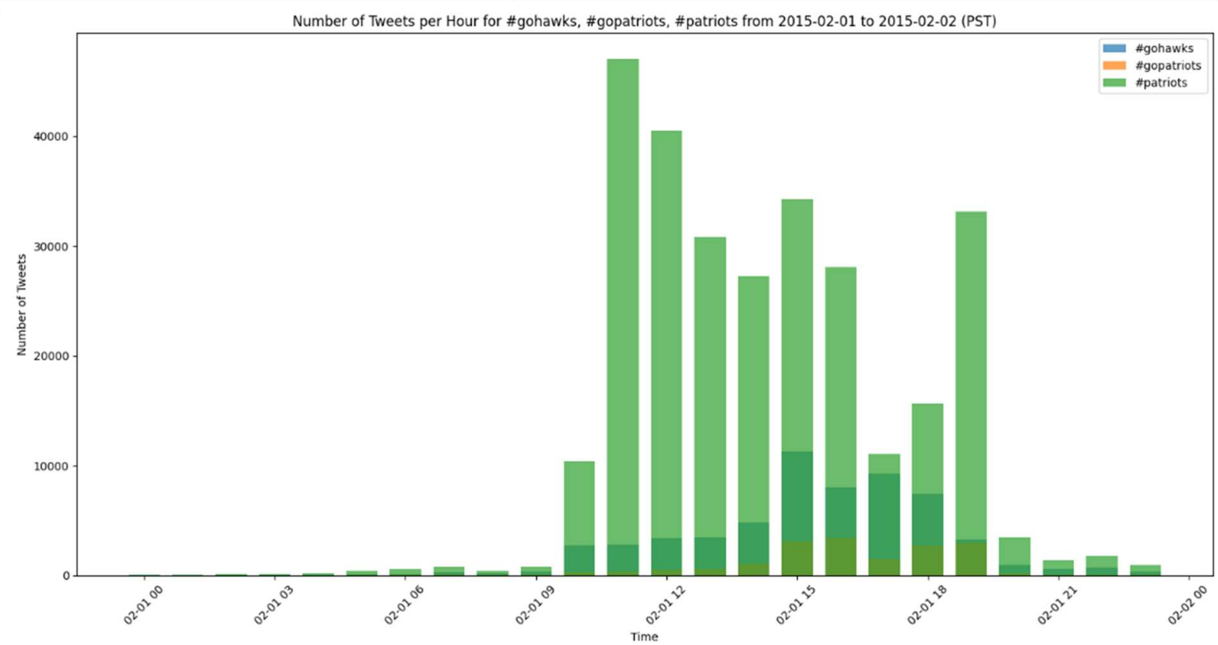
Based on the bar chart above, most tweet activity centers around the championship games on 1/18/15 and the Superbowl on 2/1/15.

```python
# lets look at the championship games
filter_resample_plot(df, '2015-01-18', '2015-01-19', ['superbowl', 'nfl'])
```



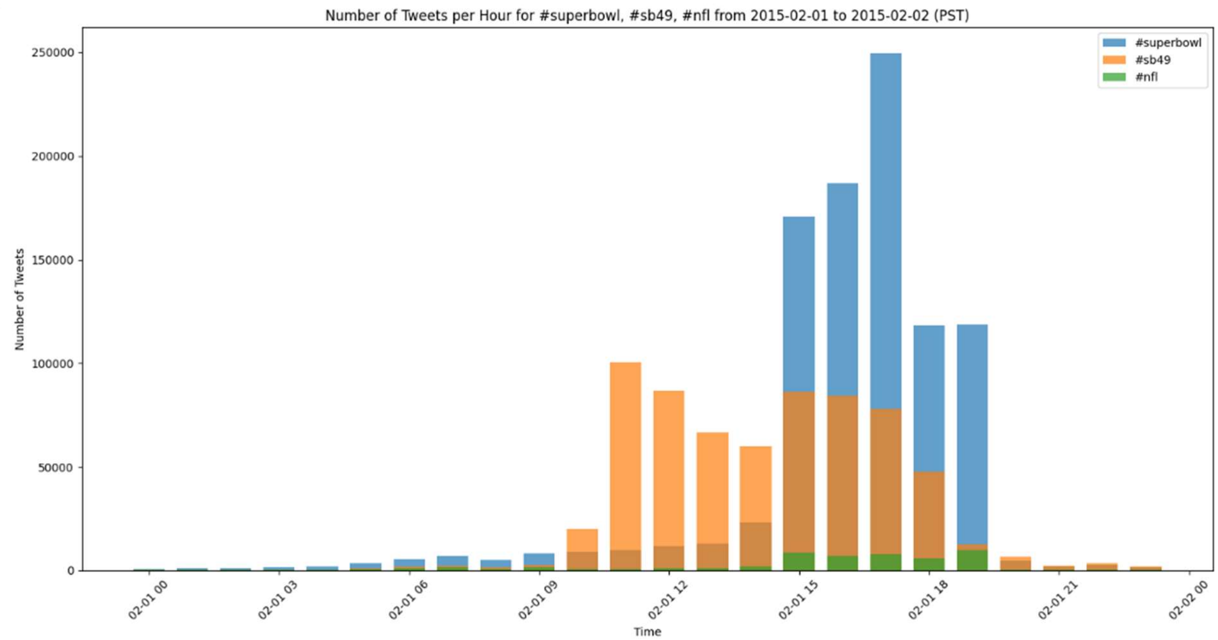
The NFL dataset (above) will contain more generalized data compared to the Superbowl dataset. This dataset will be discarded for further analysis.

```python
# lets look at other hashtags
filter_resample_plot(df, '2015-02-01', '2015-02-02', ['gohawks', 'gopatriots', 'patriots'])
```

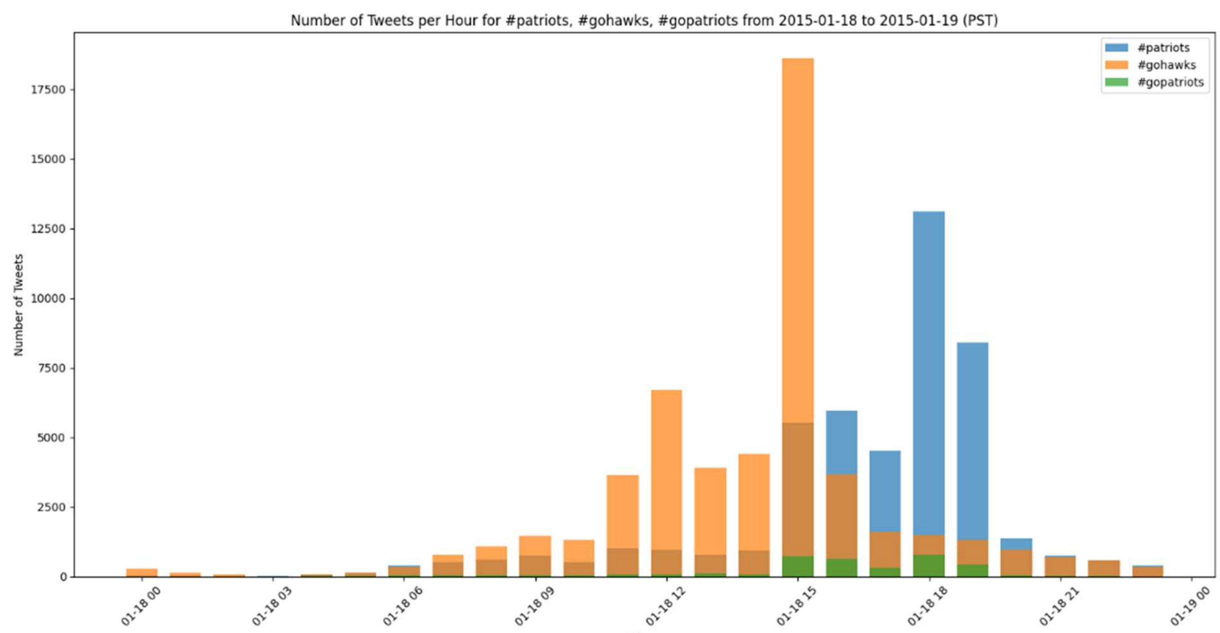


This chart above shows an imbalance of tweets in favor of the patriots.

```
# what about just the superbowl or neutral hashtags
filter_resample_plot(df, '2015-02-01', '2015-02-02', ['superbowl', 'sb49', 'nfl'])
```




```
# how about the patriots and seahawks on 1/18/15
filter_resample_plot(df, '2015-01-18', '2015-01-19', ['patriots', 'gohawks', 'gopatriots'])
```



After data exploration, the Superbowl timeframe was chosen for analysis due to large data volume available. The ESPN game event log was very detailed, but only the most significant events most likely to correlate with score changes were logged.

The twitter dataset was comprised of 6 files of tweets. To best predict score changes and keep noise to a minimum, the fan-based hashtags #GoHawks, #Patriots, and #GoPatriots were chosen for analysis. The other hashtags were either too small, or so large they likely contained an extreme amount of noise (topics like Superbowl commercials, advertising, etc).

Describe the feature engineering process

To create the feature set for model training, the following steps were accomplished:

1. **The tweets for the #gohawks, #patriots, and #gopatriots datasets were extracted for the following features:**

```
return {
  'time_posted': time_posted,
  'retweets': retweets,
  'followers': followers,
  'author_name': author_name,
  'hashtag': hashtag,
  'text': text
}
```

These features looked most promising early on, but not all would be used in the final model.

2. **A game event log was created as a CSV file:**

```
   team_action        action game_time  quarter  patriots_score  \
0     patriots          punt     11:44      1.0             0.0
1     seahawks          punt     09:30      1.0             0.0
2     patriots  interception     01:50      1.0             0.0
3     seahawks          punt     14:08      2.0             0.0
4     patriots     touchdown     09:47      2.0             7.0
5     seahawks          punt     08:17      2.0             7.0
6     patriots          punt     07:17      2.0             7.0
7     seahawks     touchdown     02:16      2.0             7.0
8     patriots     touchdown     00:31      2.0            14.0
9     seahawks     touchdown     00:02      2.0            14.0
10    seahawks    field_goal     11:09      3.0            14.0
11    patriots  interception     08:15      3.0            14.0
12    seahawks     touchdown     04:54      3.0            14.0
13    patriots          punt     03:24      3.0            14.0
14    seahawks          punt     01:05      3.0            14.0
15    patriots          punt     14:28      4.0            14.0
16    seahawks          punt     12:22      4.0            14.0
17    patriots     touchdown     07:55      4.0            21.0
18    seahawks          punt     07:00      4.0            21.0
19    patriots     touchdown     02:02      4.0            28.0
20    seahawks  interception     00:26      4.0            28.0

    seahawks_score
0              0.0
1              0.0
2              0.0
3              0.0
4              0.0
5              0.0
6              0.0
7              7.0
8              7.0
9             14.0
10            17.0
11            17.0
12            24.0
13            24.0
14            24.0
15            24.0
16            24.0
17            24.0
18            24.0
19            24.0
20            24.0
```

3. **A function was created to correlate the game time with the real time for each Superbowl event:**

```
   team_action        action game_time  quarter  patriots_score  \
0     patriots          punt     11:44      1.0             0.0
1     seahawks          punt     09:30      1.0             0.0
2     patriots  interception     01:50      1.0             0.0
3     seahawks          punt     14:08      2.0             0.0
4     patriots     touchdown     09:47      2.0             7.0
5     seahawks          punt     08:17      2.0             7.0
6     patriots          punt     07:17      2.0             7.0
7     seahawks     touchdown     02:16      2.0             7.0
8     patriots     touchdown     00:31      2.0            14.0
9     seahawks     touchdown     00:02      2.0            14.0
10    seahawks    field_goal     11:09      3.0            14.0
11    patriots  interception     08:15      3.0            14.0
12    seahawks     touchdown     04:54      3.0            14.0
13    patriots          punt     03:24      3.0            14.0
14    seahawks          punt     01:05      3.0            14.0
15    patriots          punt     14:28      4.0            14.0
16    seahawks          punt     12:22      4.0            14.0
17    patriots     touchdown     07:55      4.0            21.0
18    seahawks          punt     07:00      4.0            21.0
19    patriots     touchdown     02:02      4.0            28.0
20    seahawks  interception     00:26      4.0            28.0

    seahawks_score           real_time
0              0.0 2015-02-01 15:33:16
1              0.0 2015-02-01 15:35:30
2              0.0 2015-02-01 15:43:10
3              0.0 2015-02-01 16:37:22
4              0.0 2015-02-01 16:41:43
5              0.0 2015-02-01 16:43:13
6              0.0 2015-02-01 16:44:13
7              7.0 2015-02-01 16:49:14
8              7.0 2015-02-01 16:50:59
9             14.0 2015-02-01 16:51:28
10            17.0 2015-02-01 17:59:51
11            17.0 2015-02-01 18:02:45
12            24.0 2015-02-01 18:06:06
13            24.0 2015-02-01 18:07:36
14            24.0 2015-02-01 18:09:55
15            24.0 2015-02-01 19:03:02
16            24.0 2015-02-01 19:05:08
17            24.0 2015-02-01 19:09:35
18            24.0 2015-02-01 19:10:30
19            24.0 2015-02-01 19:15:28
20            24.0 2015-02-01 19:17:04
```

Because game times have no correspondence to real times, especially with all the time outs, standing around, halftime, the game start time, the game end time, and Tom Brady's winning touchdown time (7:15 pm PST) were used to bound the time series data. The halftime show was reported to be around 13 minutes long, which also helped bound the times series data. By lining up these known events and times, real times versus game times were estimated.

4. **Tweet data was parsed to filter the tweets from 3 pm to 8 pm on Superbowl Sunday for the #gohawks, #gopatriots, and #patriots datasets:**

```
# we'll parse this data down to tweets on superbowl sunday from 3 pm to 8 pm
# and only use the known fans in the dataset
df_filtered = filter_by_datetime_and_hashtag(df, '2015-02-01 15:00', '2015-02-01 20:00', ['gohawks', 'gopatriots', 'patriots'])
```

```
# check content and size of dataset
print(df_filtered.head())
print(df_filtered.shape)
```

```
                     time_posted  retweets  followers     author_name  \
14074   2015-02-01 16:01:21-08:00         3       27.0    Karinna Bunn
94124   2015-02-01 16:23:20-08:00         8      191.0        12121212
102917  2015-02-01 16:56:44-08:00         6      340.0    The Orca Inn
103247  2015-02-01 17:09:02-08:00       155      141.0  Keirstin Ariel
103458  2015-02-01 16:52:03-08:00        21    15086.0      Alexsandra

        hashtag                                            text
14074   gohawks                     #GoHawks http://t.co/5tIhnn3TMG
94124   gohawks  Just for Super Bowl Week and #Tittytuesday  #g...
102917  gohawks  "@NW_Music_Scene: #GoHawks and Go @NikkiSixx ....
103247  gohawks  RT to win! 1 winner will receive a signed @Mon...
103458  gohawks  .@BostonBallet #SB49 challenge: @Seahawks win=...
(175137, 6)
```

5. **The tweet text was cleaned of URLs, mentions, hashtags (text retained), words with numbers, special characters, extra whitespace, and converted to lowercase. The text was lemmatized to prepare for analysis.**

6. **Sentiment features were generated to add to the dataset using the cardiffnlp/twitter-roberta-base-sentiment-latest pretrained model:**

```
# Load the tokenizer and model
tokenizer = AutoTokenizer.from_pretrained("cardiffnlp/twitter-roberta-base-sentiment-latest")
model = AutoModelForSequenceClassification.from_pretrained("cardiffnlp/twitter-roberta-base-sentiment-latest")
```

```
print(df_filtered.head())
```

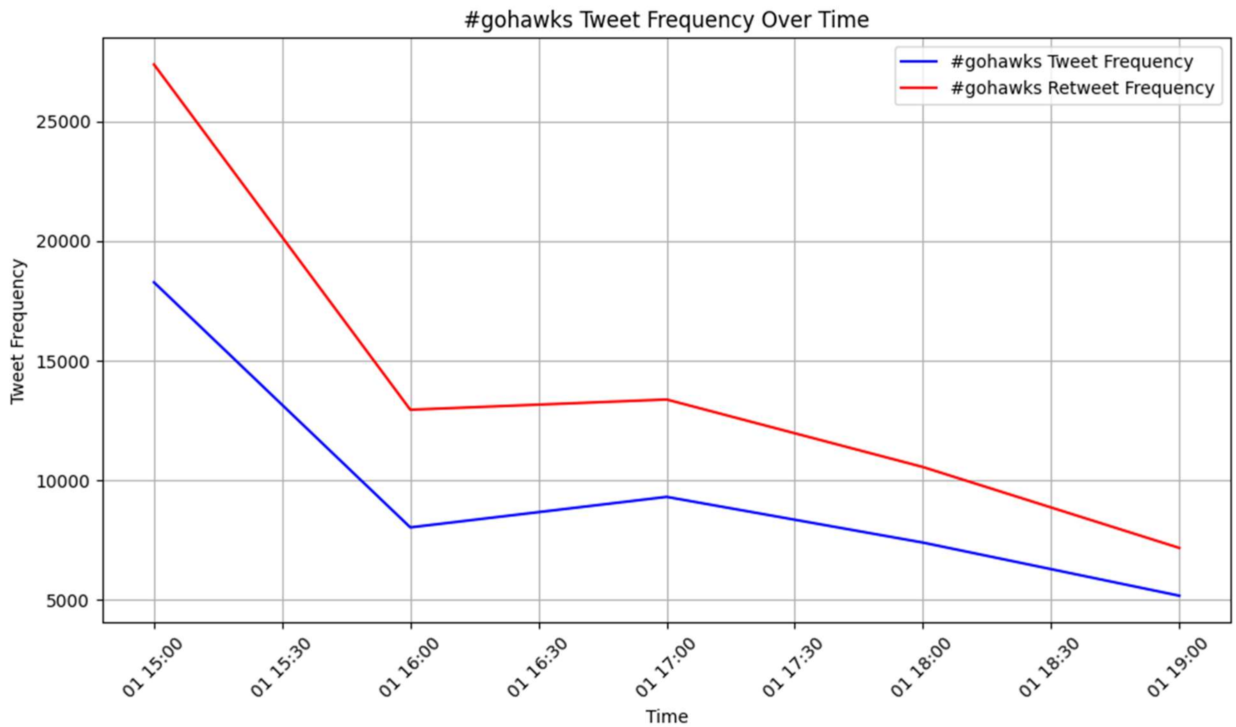
```
                     time_posted  retweets  followers    author_name  \
14074   2015-02-01 16:01:21-08:00         3       27.0   Karinna Bunn
94124   2015-02-01 16:23:20-08:00         8      191.0       12121212
102917  2015-02-01 16:56:44-08:00         6      340.0  The Orca Inn
103247  2015-02-01 17:09:02-08:00       155      141.0  Keirstin Ariel
103458  2015-02-01 16:52:03-08:00        21    15086.0     Alexsandra

        hashtag                                          text  \
14074   gohawks                  #GoHawks http://t.co/5tIhnn3TMG
94124   gohawks  Just for Super Bowl Week and #Tittytuesday  #g...
102917  gohawks  "@NW_Music_Scene: #GoHawks and Go @NikkiSixx ....
103247  gohawks  RT to win! 1 winner will receive a signed @Mon...
103458  gohawks  .@BostonBallet #SB49 challenge: @Seahawks win=...

                                          cleaned_text  \
14074                                          gohawks
94124                super bowl week tittytuesday gohawks
102917                                      gohawks go
103247  rt win winner receive signed football winner w...
103458  challenge winyou manege seattle tunepatriots w...

                                       lemmatized_text  \
14074                                          gohawks
94124                super bowl week tittytuesday gohawks
102917                                      gohawks go
103247  rt win winner receive sign football winner win...
103458  challenge winyou manege seattle tunepatriots w...

                                      sentiment_scores  sentiment_negative  \
14074   {'negative': 0.17648447, 'neutral': 0.60134536...            0.176484
94124   {'negative': 0.09190893, 'neutral': 0.552486, ...            0.091909
102917  {'negative': 0.11995082, 'neutral': 0.70555514...            0.119951
103247  {'negative': 0.0057257037, 'neutral': 0.231541...            0.005726
103458  {'negative': 0.02970109, 'neutral': 0.80486184...            0.029701

        sentiment_neutral  sentiment_positive
14074            0.601345            0.222170
94124            0.552486            0.355605
102917           0.705555            0.174494
103247           0.231541            0.762733
103458           0.804862            0.165437
```

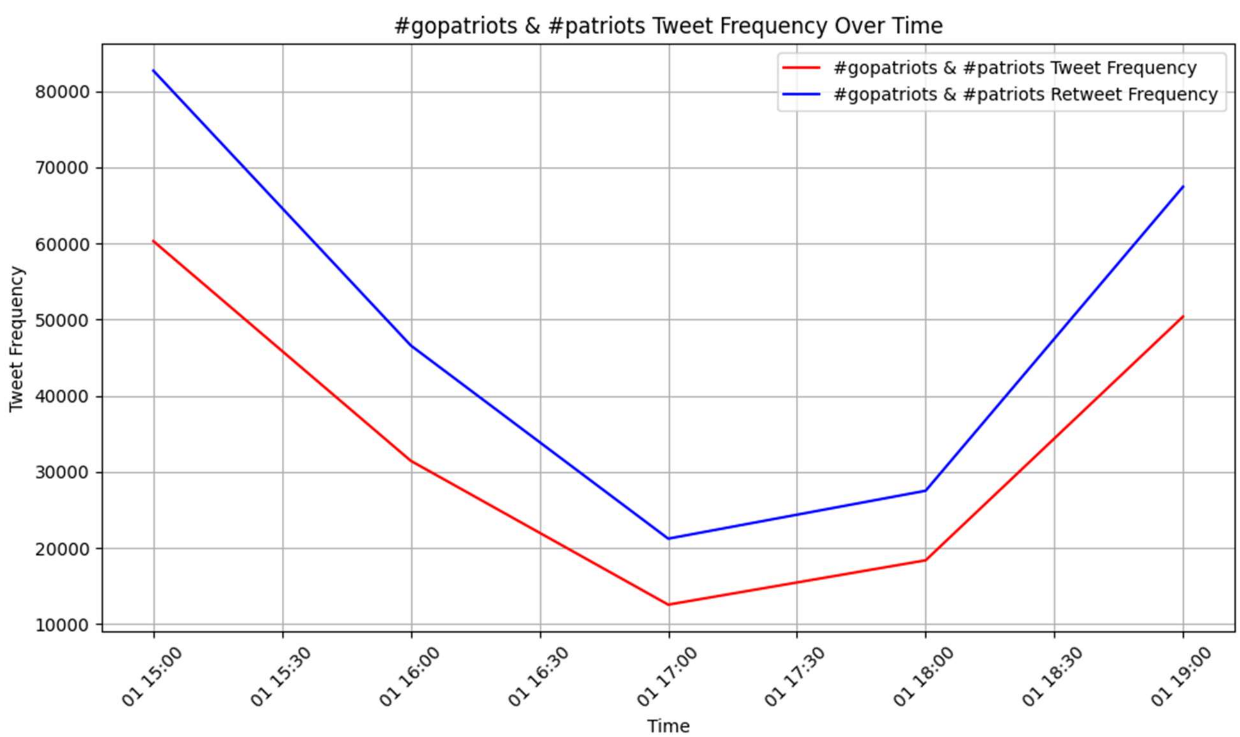
7. **The tweets dataframe and game log dataframe were combined by merging the tweet time posted and the real time of the game events. The tweets that occur prior to the first event are assigned to that first event and tweets that occur after the final event are assigned the last event:**

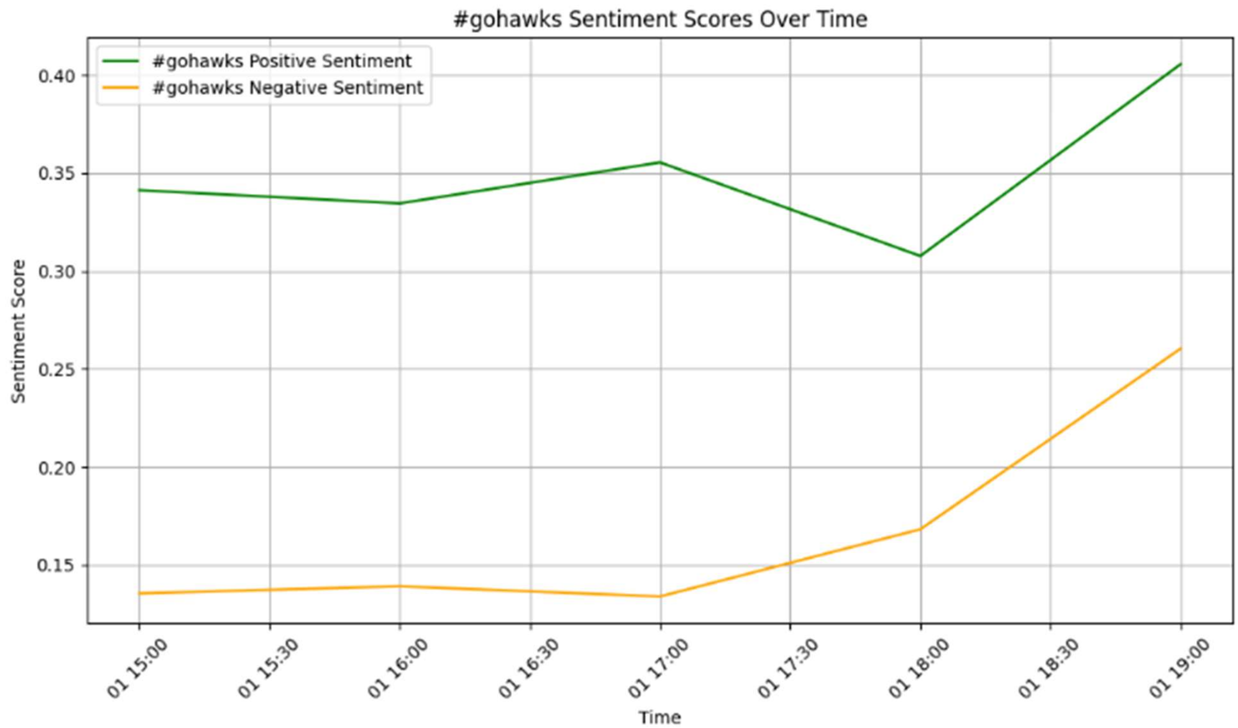
```
                time_posted  retweets  followers    author_name     hashtag  \
0 2015-02-01 15:00:00-08:00         2     7664.0  Dennis Bounds     gohawks
1 2015-02-01 15:00:00-08:00         1       37.0    mrs.Jozelia  gopatriots
2 2015-02-01 15:00:00-08:00         1       53.0    Heidi Inman     gohawks
3 2015-02-01 15:00:00-08:00         2       45.0    Gwenie Rose     gohawks
4 2015-02-01 15:00:00-08:00         1      381.0     Sean Mason     gohawks

                                                text  \
0  Touchdown in Seattle. Now for #seahawks TDs in...
1  Hoje nÃ£o tem pra ninguÃ©m e patriots #GoPatri...
2        #GoHawks that is all http://t.co/fV7t5QPXys
3  Ready to party...#GoHawks #SB49 #Seahawks http...
4                           30 minutes!!! #GoHawks

                                        cleaned_text  \
0    touchdown seattle seahawks tds see game gohawks
1  hoje tem pra ningum e patriots gopatriots gopa...
2                                            gohawks
3                       ready partygohawks seahawks
4                                    minutes gohawks

                                     lemmatized_text  \
0    touchdown seattle seahawks tds see game gohawks
1  hoje tem pra ningum e patriot gopatriots gopat...
2                                            gohawks
3                       ready partygohawks seahawks
4                                     minute gohawks

                                    sentiment_scores  sentiment_negative  \
0  {'negative': 0.031198062, 'neutral': 0.8569932...            0.031198
1  {'negative': 0.065646134, 'neutral': 0.7817235...            0.065646
2  {'negative': 0.17648447, 'neutral': 0.60134536...            0.176484
3  {'negative': 0.021161215, 'neutral': 0.3651389...            0.021161
4  {'negative': 0.08994685, 'neutral': 0.685586, ...            0.089947

   sentiment_neutral  sentiment_positive team_action action game_time  \
0           0.856993            0.111809    patriots   punt     11:44
1           0.781724            0.152630    patriots   punt     11:44
2           0.601345            0.222170    patriots   punt     11:44
3           0.365139            0.613700    patriots   punt     11:44
4           0.685586            0.224467    patriots   punt     11:44

   quarter  patriots_score  seahawks_score                 real_time
0      1.0             0.0             0.0 2015-02-01 15:33:16-08:00
1      1.0             0.0             0.0 2015-02-01 15:33:16-08:00
2      1.0             0.0             0.0 2015-02-01 15:33:16-08:00
3      1.0             0.0             0.0 2015-02-01 15:33:16-08:00
4      1.0             0.0             0.0 2015-02-01 15:33:16-08:00
```

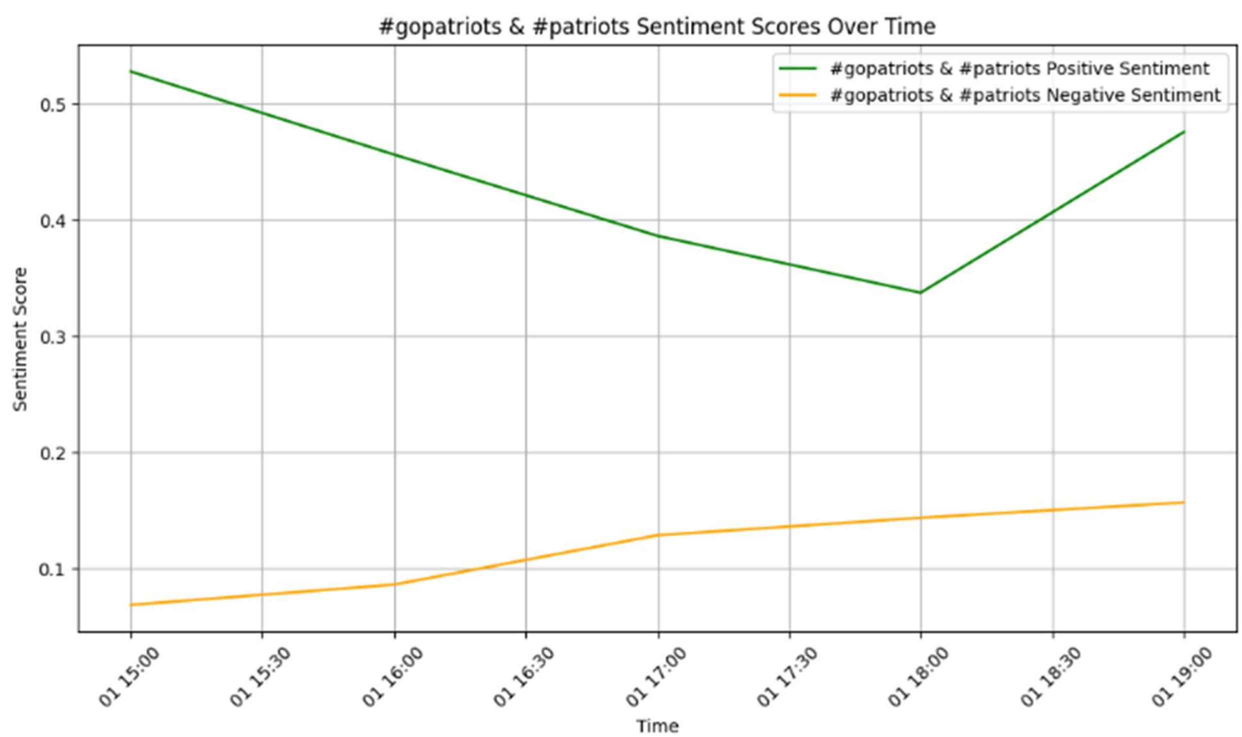
```
                     time_posted  retweets  followers        author_name  \
221351 2015-02-01 19:59:50-08:00         1      149.0      Norman Norman
221352 2015-02-01 19:59:55-08:00         1       60.0       Pete Naiukow
221353 2015-02-01 19:59:57-08:00         4     1664.0  FoxboroughFire2252
221354 2015-02-01 19:59:57-08:00         1      443.0         Lisa Mezik
221355 2015-02-01 19:59:59-08:00         1     1423.0      cheryl daniel

         hashtag                                               text  \
221351  patriots  Congrats to Tom Brady for breaking a post-seas...
221352  patriots  #superbowlcommercials \n#PatriotsWIN #Patriots...
221353  patriots  @Patriots Congratulations to our hometown New ...
221354  patriots  The #Patriots Win 28-24!!!! #Superbowl49 http:...
221355   gohawks  @eigenseide @DaynaOG @bcondotta Hope #Seahawks...

                                             cleaned_text  \
221351  congrats tom brady breaking postseason record ...
221352  superbowlcommercials patriotswin patriotsvssea...
221353  congratulations hometown new england patriots ...
221354                                       patriots win
221355  hope seahawks stay strong difficult let hope k...

                                          lemmatized_text  \
221351  congrats tom brady break postseason record ' t...
221352  superbowlcommercials patriotswin patriotsvssea...
221353  congratulation hometown new england patriot su...
221354                                        patriot win
221355  hope seahawks stay strong difficult let hope k...

                                         sentiment_scores  sentiment_negative  \
221351  {'negative': 0.002747489, 'neutral': 0.0349802...            0.002747
221352  {'negative': 0.010164936, 'neutral': 0.4776532...            0.010165
221353  {'negative': 0.0029651353, 'neutral': 0.031372...            0.002965
221354  {'negative': 0.031509362, 'neutral': 0.4552024...            0.031509
221355  {'negative': 0.02216066, 'neutral': 0.15873042...            0.022161

        sentiment_neutral  sentiment_positive team_action       action  \
221351           0.034980            0.962272    seahawks  interception
221352           0.477653            0.512182    seahawks  interception
221353           0.031373            0.965662    seahawks  interception
221354           0.455202            0.513288    seahawks  interception
221355           0.158730            0.819109    seahawks  interception

       game_time  quarter  patriots_score  seahawks_score  \
221351     00:26      4.0            28.0            24.0
221352     00:26      4.0            28.0            24.0
221353     00:26      4.0            28.0            24.0
221354     00:26      4.0            28.0            24.0
221355     00:26      4.0            28.0            24.0

                       real_time
221351 2015-02-01 19:17:04-08:00
221352 2015-02-01 19:17:04-08:00
221353 2015-02-01 19:17:04-08:00
221354 2015-02-01 19:17:04-08:00
221355 2015-02-01 19:17:04-08:00
```

8. **Tweet and retweet frequencies and sentiment scores over time were plotted to ensure they track with the game events and features are chosen for the model:**






#gohawks Sentiment Scores Over Time
#gohawks Positive Sentiment
#gohawks Negative Sentiment
Sentiment Score
0.40
0.35
0.30
0.25
0.20
0.15
01 15:00
01 15:30
01 16:00
01 16:30
01 17:00
01 17:30
01 18:00
01 18:30
01 19:00
Time


#gopatriots & #patriots Sentiment Scores Over Time
#gopatriots & #patriots Positive Sentiment
#gopatriots & #patriots Negative Sentiment
Sentiment Score
0.5
0.4
0.3
0.2
0.1
01 15:00
01 15:30
01 16:00
01 16:30
01 17:00
01 17:30
01 18:00
01 18:30
01 19:00
Time

These charts do track known game events, although fans (regardless of team) get more negative over the course of a game. Both the positive and negative sentiment will be added as features, because both sentiments have very clear and distinct trends that align with game events, especially when separated into team fan buckets.

The tweet and retweet frequencies show distinct trends. Several attempts to add retweet frequency by fan base over time as a feature were attempted, but the merges failed to align properly with the time stamps on the original tweets. The "retweets" were retained in the feature set, knowing they won't likely predict team scores, and that hunch was proven later in the feature analysis. These models would have been stronger with that feature correctly configured and included, but sadly, the attempt failed.

Other features that should align with team scores are team action (who has the ball), action (punt, touchdown, etc.), hashtag (to identify the fan), and the lemmatized text. Although the text is already "incorporated" into the model as sentiment scores, the text as features is also included to capture common words such as "touchdown", "winning", "losing", etc. Although Twitter data is garbage, at volume scale enough signals in the text help aid the model.

The final list of features for model training are:

Choose features for model training

```
final_df['sentiment_positive'] = final_df['sentiment_positive'].astype(float)
final_df['sentiment_negative'] = final_df['sentiment_negative'].astype(float)
final_df['retweets'] = final_df['retweets'].astype(int)
```

```
# Feature selection
features = ['sentiment_positive', 'sentiment_negative', 'retweets', 'hashtag', 'lemmatized_text', 'action', 'team_action']
X = final_df[features]
```

```
# get the hashtag counts from final_df
hashtag_counts = final_df['hashtag'].value_counts()
print(hashtag_counts)
```

```
hashtag
patriots      156704
gohawks        48208
gopatriots     16444
Name: count, dtype: int64
```

9. **And lastly, the team scores were converted to dynamic labels so we can predict either "tie score", "seahawks winning", or "patriots winning":**

```
# Assign dynamic labels for the winning team for labels
def determine_winning_team(row):
    if row['patriots_score'] > row['seahawks_score']:
        return 2
    elif row['patriots_score'] < row['seahawks_score']:
        return 1
    else:
        return 0

# patriots winning = 2
# seahawks winning = 1
# tie score = 0
```

```
[ ] # Apply the function to each row to create a 'winning_team' column
final_df['winning_team'] = final_df.apply(determine_winning_team, axis=1)

#create labels as y
y = final_df['winning_team']
```

Generate baselines for final ML model

Several competing models were trained and evaluated. The best performing models were the LightGBM, CatBoost, XGBoost, Random Forest, OLS Logistic Regression, and SVM. A hybrid ensemble model using a pretrained BERT model for the lemmatized text and Random Forest for the remaining features was attempted, but results were much weaker than a single model. This hybrid model failure revealed that text data features must be analyzed in tandem with the other features, so a single tree-based model is the best choice for this dataset.

The Light GBM model was selected and performed with a final 5-fold cross validated grid search to fine tune parameters. Performance degraded slightly, likely because this was the only step chosen for 5-fold cross validation. The train/test split evaluations might be slightly overfit, but it saved computation time as an initial screening.

The best parameters for the Light GBM model were:

```
Best parameters: {'classifier__max_depth': -1, 'classifier__n_estimators':
50, 'classifier__num_leaves': 51}
```

The F-1 scores and model accuracies for the models are reported in Table I:

Table I. Model Comparisons and Final Fine Tuned Model Performance

Initial Model Comparison	Tie Score (F1-Score)	Seahawks Winning (F-1 Score)	Patriots Winning (F-1 Score)	Model Accuracy
LightGBM	**0.96**	**0.89**	**0.92**	**0.94**
CatBoost	**0.96**	**0.88**	**0.92**	**0.93**
XGBoost	**0.95**	**0.88**	**0.92**	**0.93**
Random Forest	**0.95**	**0.88**	**0.92**	**0.93**
OLS Logistic Regression	**0.94**	**0.88**	**0.88**	**0.92**
SVM	**0.94**	**0.88**	**0.89**	**0.92**
BERT + Random Forest	**0.92**	**0.79**	**0.89**	**0.89**
Final Model With 5-Fold Cross Validation and Parameter Tuning:				
LightGBM	**0.95**	**0.88**	**0.92**	**0.93**

Feature importance was assessed on the fine-tuned LightGBM model (see Figure 1)l:

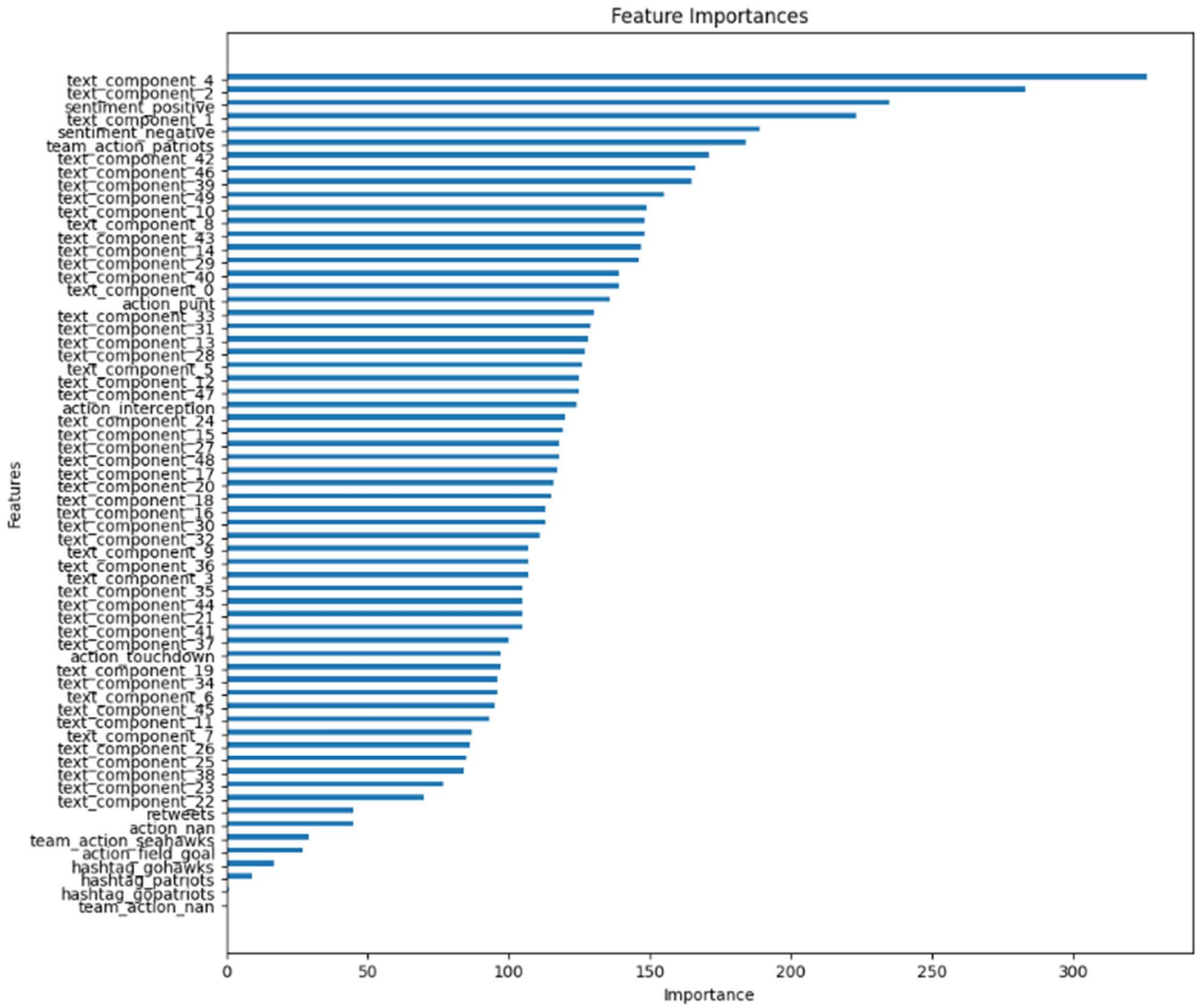


Figure 1. Feature importance analysis on the final LightGBM model.

Analyzing the Top 10 most salient features (see Figure 2), the list contains positive and negative sentiments, the patriots having possession of the ball, and several text components.

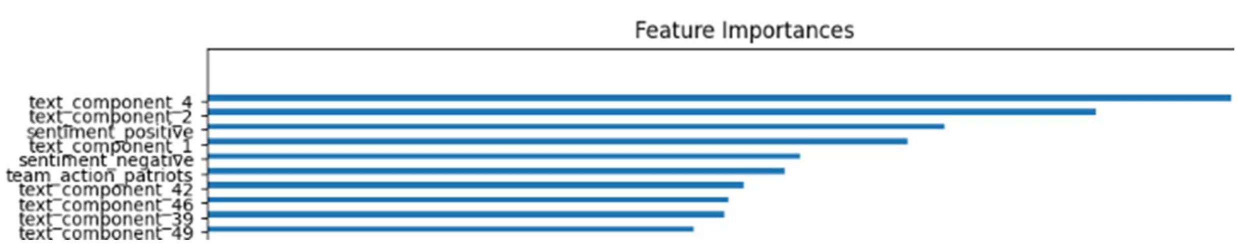


Figure 2. Feature importances for the Top 10 most salient feature

A deeper look into the text components reveals the terms (words) influencing the feature importances. The Top 10 most salient words in each text component are shown in Figure 3 (below):

```
Text component 4:
  touchdown: 0.8635935330979175
  patriot: 0.05168363508340023
  ve: 0.02253383122095975
  yes: 0.015415402661766013
  gronk: 0.01259418458942539
  beastmode: 0.011838002718384512
  gohawks: 0.009627668025291535
  another: 0.008607078200562676
  lafell: 0.008057362415336049
  baldwin: 0.007429695545960937
```

```
Text component 2:
  gohawks: 0.9481047266059519
  go: 0.07077487503199784
  superbowlxlix: 0.0665840047907548
  seahawks: 0.06649007025304439
  let: 0.054680497967131
  touchdown: 0.04617582707347313
  superbowl: 0.03768725917974085
  patriot: 0.03289655069572725
  game: 0.03276601993756433
  beastmode: 0.02922159424945246
```

```
Text component 1:
  gopatriots: 0.9343071220707548
  gohawks: 0.21788676736324813
  superbowl: 0.13705989042245742
  superbowlxlix: 0.1288892153260202
  touchdown: 0.11565525108661853
  patriot: 0.08189353480696794
  go: 0.06268244894737637
  gopats: 0.049298790117621984
  seahawks: 0.04078638220609031
  let: 0.040544411994697764
```

```
Text component 42:
  watch: 0.5184479529224391
  tombrady: 0.37681768052436754
  patsnation: 0.2038307356695118
  catch: 0.16006732561782583
  new: 0.12372821564936644
  england: 0.11645361249277343
  xlix: 0.10337345689450303
  superbowi: 0.10199137103182036
  congrats: 0.07544113558847022
  patriotsnation: 0.07512601170619781
```

```
Text component 46:
  watch: 0.3998004728591332
  doyourjob: 0.3021588671746467
  champion: 0.2550370172472826
  fan: 0.19095042025900344
  great: 0.14176449177808892
  superbowi: 0.13823742888299886
  team: 0.1324572870687765
  lob: 0.12954250224795896
  football: 0.11203984205889682
  xlix: 0.10916442082829232
```

```
Text component 39:
  catch: 0.8135919764120302
  love: 0.139668326951371
  superbowi: 0.13277835609718525
  matthew: 0.11722550411032844
  doyourjob: 0.09653829448685777
  defense: 0.09086513151781919
  kearse: 0.08588857929256717
  football: 0.08126350659966779
  chris: 0.07756396969405287
  finishthejob: 0.07348839164381886
```

```
Text component 49:
  hawk: 0.5092478150150436
  defense: 0.3941364483254001
  xlix: 0.30669119921085064
  team: 0.25947194977045124
  baby: 0.14519611867238666
  beast: 0.13866905199977025
  seattleseahawks: 0.11809564402606046
  ball: 0.11363036225807728
  sea: 0.10852377665509658
  catch: 0.10710418025953239
```

Figure 3. The Top 10 most salient words present in the Figure 2 text components.

Looking at the Top 10 words in each text component, the top word or the top 2 words (depending on weights) are listed as the most important terms driving model predictions:

1. touchdown
2. gohawks
3. gopatriots
4. watch
5. tom brady
6. doyourjob
7. champion
8. catch
9. hawk
10. defense

Predicting "seahawks winning" was the toughest category, which makes sense considering the Seahawks spent much less clock time ahead of the Patriots during this game. A tie score was easier to predict because the game remained scoreless prior to game time, and for a large portion of the first half. And, the Patriots had more action associated with them, especially with so much attention on Tom Brady, so the Patriots team was much easier for the model to predict.

Initially it seemed surprising that textual components played so large a role in model predictions, but after analyzing the Top 10 most salient words, this result makes sense. Twitter text data requires sufficient augmentation with other datasets and external features for meaningful analysis to take place.

Use a generative model to create a tweet based on game score

```python
# Load gpt2 pre-trained generative model
generator = pipeline('text-generation', model='gpt2')

def generate_synthetic_tweet(score_change):
    prompt = f"The current score is: {score_change}. Update tweet: "
    generated_tweet = generator(prompt, max_length=50, truncation='longest_first', num_return_sequences=2)
    return generated_tweet[0]['generated_text']

# Use the function to generate a synthetic tweet
score_change = "Patriots 14 - Seahawks 10"
tweet = generate_synthetic_tweet(score_change)
print(tweet)
```

```
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.

The current score is: Patriots 14 - Seahawks 10. Update tweet:  "NFL is not
worried about their safety. We have a number of issues, and this game goes
down as an aberration." pic.twitter.com/zS
```

This model works surprisingly well, but also generates some nonsense statements, which isn't much different than X (Twitter) on a regular basis. We'll call this model a winner.