



Ballon D'or Sentiment Analysis

Jonathan Fuhrman, Joseph Busacca, Talal Hakki, Evan Garvey, Austin Johnson

About Us

Talal & Jonathan

Data Gathering

Evan & Austin

Data Analysis

Joseph

Data Visualization



Approach

THE BALLON D'OR

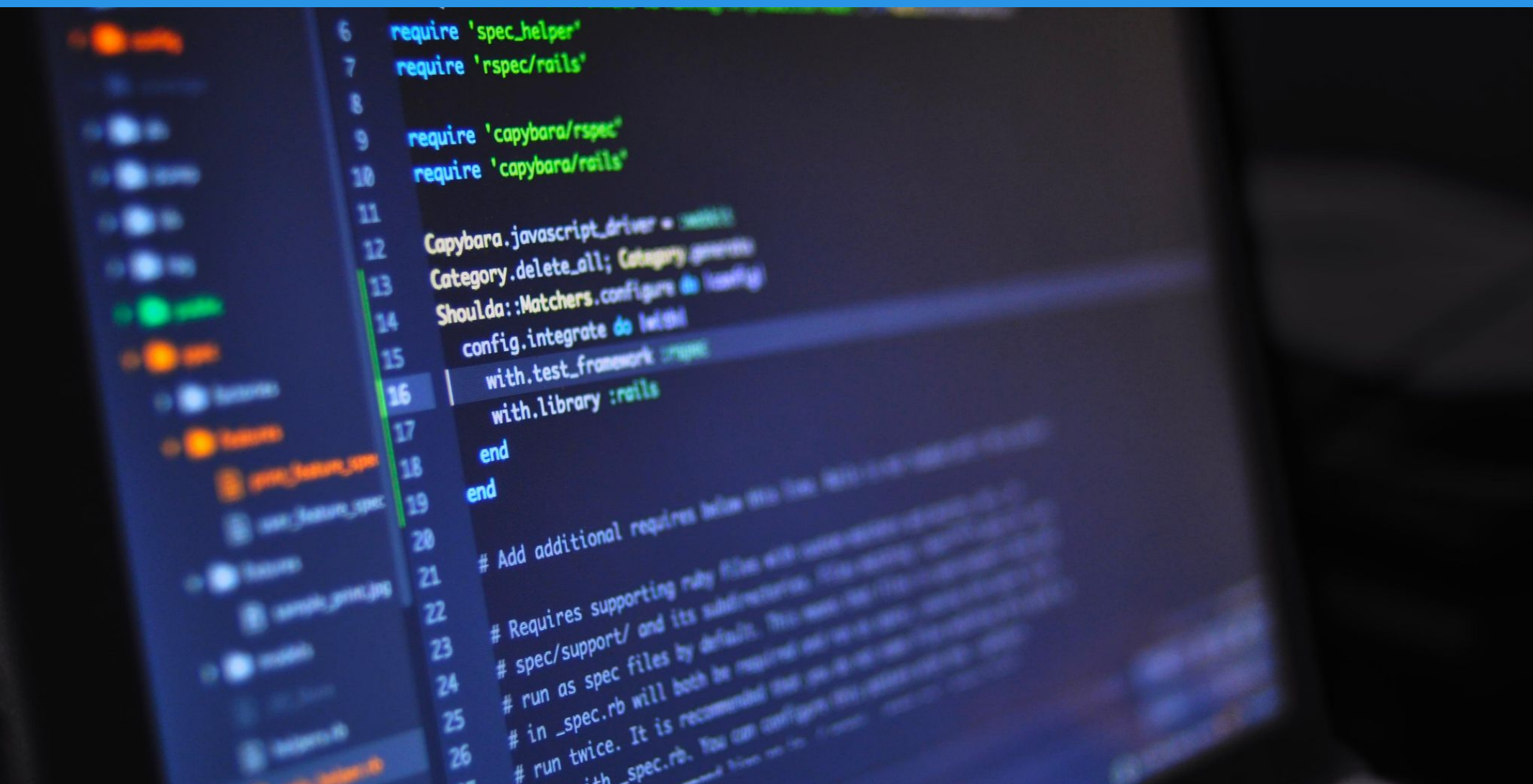
The Ballon D'or is an annual soccer award given to the player chosen by a majority vote from FIFA officials, coaches and captains of national teams

POTENTIAL CANDIDACY

Given the nature of how the Ballon D'or is awarded, we believe that researching public approval and disapproval of popular Ballon D'or candidates this year provides insight into who might win the award.

OUR IMPLEMENTATION

Take a quantitative approach towards who the Ballon D'or could potentially be awarded to, by analyzing public sentiment via Twitter.



The background of the left slide is a large, stylized Twitter logo in white on a blue background. The logo is partially obscured by a dark blue rectangular box containing text.

Mission

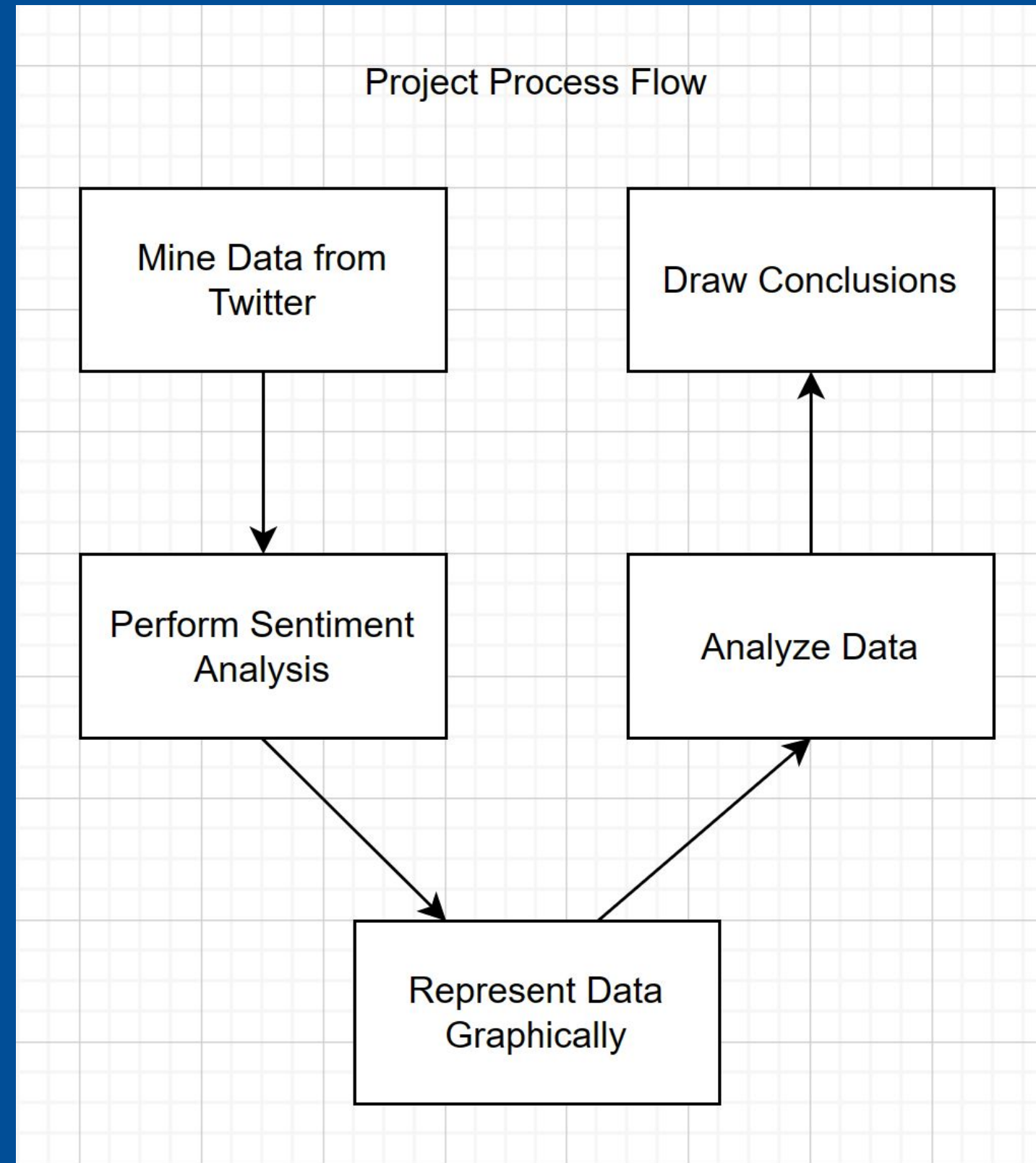
Use Sentiment Analysis to predict the finalists for the ballon d'or

The background of the right slide is a close-up of an Oscar statuette, showing its head and upper body. The statuette is gold and set against a dark background. A dark blue rectangular box containing text is overlaid on the lower part of the image.

Vision

Use how people are talking about a topic to predict the results

Process Flow



Data Gathering

```
# Includes modified code from the Twitter Cookbook #1
# This function takes in a list of search strings for each player, and outputs 300 tweets per player
# to the playerTweets.txt file.
def getPlayerTweets(playerNameList):      # playerNameList is a list of the search keyword for each player

    file = open("playerTweets.txt", "w")

    for playerQuery in playerNameList:

        q = playerQuery
        count = 100

        search_results = twitter_api.search.tweets(q=q, count=count)
        statuses = search_results['statuses']

        # Iterate through 2 more batches of results by following the cursor, so 300 tweets are retrieved for the cur
        for _ in range(2):
            print('Length of statuses', len(statuses))
            try:
                next_results = search_results['search_metadata']['next_results']

            except KeyError as e: # No more results when next_results doesn't exist
                break

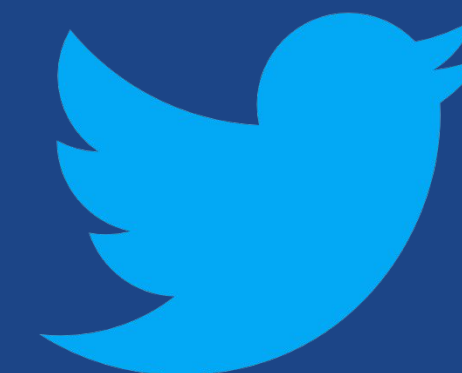
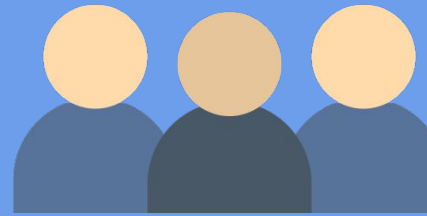
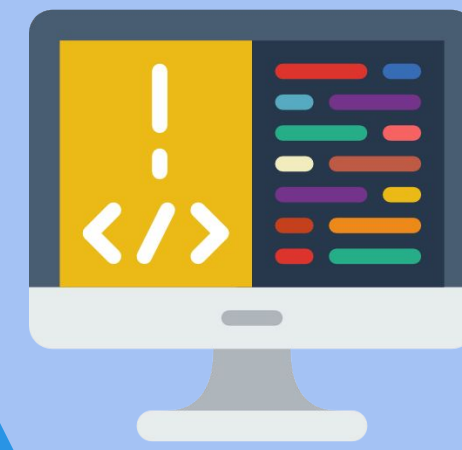
            # Create a dictionary from next_results, which has the following form:
            # ?max_id=847960489447628799&q=%23RIPSelena&count=100&include_entities=1
            kwargs = dict([ kv.split('=') for kv in unquote(next_results[1:]).split("&") ])

            search_results = twitter_api.search.tweets(**kwargs)
            statuses += search_results['statuses']

        print('Length of statuses', len(statuses))

        file.write("\n\n**Tweets for searching: " + playerQuery + "**")
        # This is a way of accessing/printing each tweet text
        for i in range(len(statuses)):
            file.write("\n\n")
            file.write(statuses[i]['text'])

    # close file for writing
    file.close()
```



Part 1

- Receive the tweets, and use Regex to separate them into discrete sections
- Preprocess them by removing stopwords and lemmatizing with NLTK

Part 2

- Take the preprocessed tweets and run sentiment analysis on each one
- Put the sentiment scores into an array for simple categorization and graphical analysis

Data Analysis

```
lemmatizer = WordNetLemmatizer()
preTweets = []
fiveSections = re.split(tweets, "\\*\\*([A-Za-z]+([A-Za-z]+)+): ([A-Za-z]+([A-Za-z]+)+)\\*\\*")
for section in fiveSections:

    sectionTweets = re.split(section, "RT @[A-Za-z0-9]+ [A-Za-z0-9]+|@[A-Za-z0-9]+ [A-Za-z0-9]+")
    sectionPreTweets = []
```

```
analyzer = SentimentIntensityAnalyzer()

sentiments = []
for section in tweets:

    sentimentSection = []
    for tweet in section:

        scores = analyzer.polarity_scores(tweet)
        sentimentSection.append(scores['pos'])
```



Data Visualization

Using Matplotlib and NumPy


Pie chart for each player showing positive vs negative tweets

Pie chart for all players showing who had the highest percent of positive tweets

These graphs can show:

- **expected winner**
- **ratio of +/- tweets**



- 
- An aerial photograph of a large soccer stadium, likely Wembley, filled with a massive crowd of spectators. The pitch is green, and the goalposts are visible. The image is partially obscured by a blue geometric overlay on the right side.
- Provides insight into the opinions of soccer fans on Twitter
 - Can replicate this project on other forms of data to make informed decisions

Conclusion

