A description of the data pipeline and procedures I’ve executed. The following will be organized according to the data order of operations outlined by high level descriptions of python scripts, data inputs and data outputs. Important datasets will appear emboldened. I understand this is not how methods and procedures will be presented in the final paper.

DATA SOURCES

* **In-game betting odds, 20-milliseconds basis**
* **Tweets per game (two data sets, i.e. per home team and away team), millisecond basis**

EVENTS

* **In-game events per game, per second**

Script(s)

* ws\_get\_links.py
  + scrape match urls for each season
* ws\_scrape\_links.py
  + with match urls, scrape text commentaries per match
* ws\_process\_xml.py
  + transform text commentaries into timestamped xml files
* get\_teams.py
  + traverse xml files, replace team names to match with odds dataset

SENTIMENT

* **In-game tweet sentiment, millisecond basis**

Script(s)

* nlp.py
  + predict sentiment of tweet per match using a random forest regressor trained on data from the Stanford Sentiment Tree Bank
  + output is added as a column named ‘prediction’ to existing sentiment score files

AGGREGATE

* **In-game betting odds, events, and tweet sentiment aggregated by minute**

Script(s)

* aggregate\_min.py
  + merging and aggregation of all datasets on match minute
  + betting odds matches are aggregated in 3 ways… mean, median, and effective
    - effective odds are calculated using the total volumes matched multiplied by match price
  + parse xml event files, give comma separated list of events occurring in chronological order in every given minute
  + twitter sentiment columns given include
    - mean of sentiment scores per minute
    - mean of sentiment scores weighted by retweets per minute
      * retweets treated as individual tweets. The original tweet sentiment is duplicated per retweet and added to the mean

REFORMAT

* **Dataset final form: Wide, in-game betting odds, rescaled probabilities, events, and tweet sentiment aggregated by minute**

Script(s)

* reformat\_dataset.py
  + make dataset wider by merging away, draw, and home into one observation
* rescale.py
  + create outcome probability columns with 1 / odds
  + rescale probabilities to remove overround

SURPRISE & SHOCK

* **Dataset final form with surprise and shock calculated**

Script(s)

* calculate\_metrics.py
  + surprise - the square root of the sum of the squared differences in outcome probabilities and outcome probabilities in the previous time period
  + shock - the square root of the sum of the squared differences in outcome probabilities and outcome probabilities pre match

CLEAN UP

Script(s)

* cleanup\_data.py
  + remove duplicate rows
  + change incorrect in-play values
  + subtract twitter data timestamps by 1 hour
  + remove twitter draw columns

SUSPENSE

To calculate suspense, I referenced Tim’s instructions (attached at the end), along with the Buraimo et al. paper.

* **Dataset final form complete with surprise, shock, sentiment, AND suspense**

Script(s)

* estimate\_scoring\_rates.py
  + Assume independently Poisson distributed number of goals scored by home and away
  + Using closing odds on O/U-totals averaged over bookmakers, estimate scoring rates by minimizing the squared difference between the bookmaker implied probabilities and the outcome probabilities from the in-play model
* get\_scoring\_distrib.py
  + use event data to count the scoring distribution across minutes for the season
  + calculate a moving average over 15
  + backward interpolate the missing values at the beginning of each series
* simulate\_goals.py
  + using the previous ingredients: scoring rates and scoring distribution, simulation every minute of play 100k times, record results. Do this for every match
* calculate\_suspense.py
  + calculate hypothetical probabilities
    - iterate through each minute
      * get the probabilities for H-win, draw and A-win given a home score (use simulation values added to current home score)
        + square the difference between these and the given probabilities for H-win, draw and A-win for the minute
        + multiply that squared difference by the probability of scoring in the next minute
        + sum all of these values
      * get the probabilities for H-win, draw and A-win given an away score (use simulation values added to current away score)
        + square the difference between these and the given probabilities for H-win, draw and A-win for the minute
        + multiply that squared difference by the probability of scoring in the next minute
        + sum all of these values
      * sum the sums of values

TO DO

* Isolate matches with red cards. Re-calculate suspense/run simulations for said matches with red card penalties.

SUSPENSE INSTRUCTIONS

1) Score lines per match

- Objective: estimate the scoring rates of each team

- Ingredients:

o Closing odds on the result averaged over bookmakers

o Closing odds on O/U-totals averaged over bookmakers

- Procedure:

o Assuming independently Poisson distributed number of goals scored b H and A

o Predict the scoreline probabilities for each match (match-level)

o Sum all predicted scoreline probabilities to calculate the probabilities of potential match outcomes

o Estimate scoring rates by minimizing the squared difference between the bookmaker implied probabilities

and the outcome probabilities from the in-play model

2) Score lines per minute

- Objective: distributing scoring rates across the minutes

- Ingredients: use scores dataset

- Procedure:

o Assume the average amount of injury time to evenly share out the inflated scoring rates in the

45th and 90th min across these extra minutes

o Presume all matches to be 93 min long

o Calculate a moving average over 15 min to smooth the relative frequency distributions

o Backward interpolate the missing values at the beginning of each series (first 15 min)

o calculate the density function of goals scored per minute

- Problem:

o average scoring behavior might not adequately represent a team’s scoring pattern

o eventually use team-specific empirical goal distributions (though then we might have several zeros since not all teams score in all minutes)

3) Number of goals per minute

- Simulate the number of goals in each minute using the score lines per minute

- Sum up score line and record result, repeat 100,000 times per min => 9 Mio times per match

4) Calculate hypothetical probabilities

- what are the probabilities for H-win, draw, A-win if either H or A do score in the next min

- to calculate suspense

- iterate through each minute

- get the probabilities for H-win, draw and A-win given a home score (use simulation values added to current home score)

- square the difference between these and the given probabilities for H-win, draw and A-win for the minute

- multiply that squared difference by the probability of scoring in the next minute

- sum all of these values

- get the probabilities for H-win, draw and A-win given an away score (use simulation values added to current away score)

- square the difference between these and the given probabilities for H-win, draw and A-win for the minute

- multiply that squared difference by the probability of scoring in the next minute

- sum all of these values

- sum the sum of values

- Account for red cards by modifying scoring rates