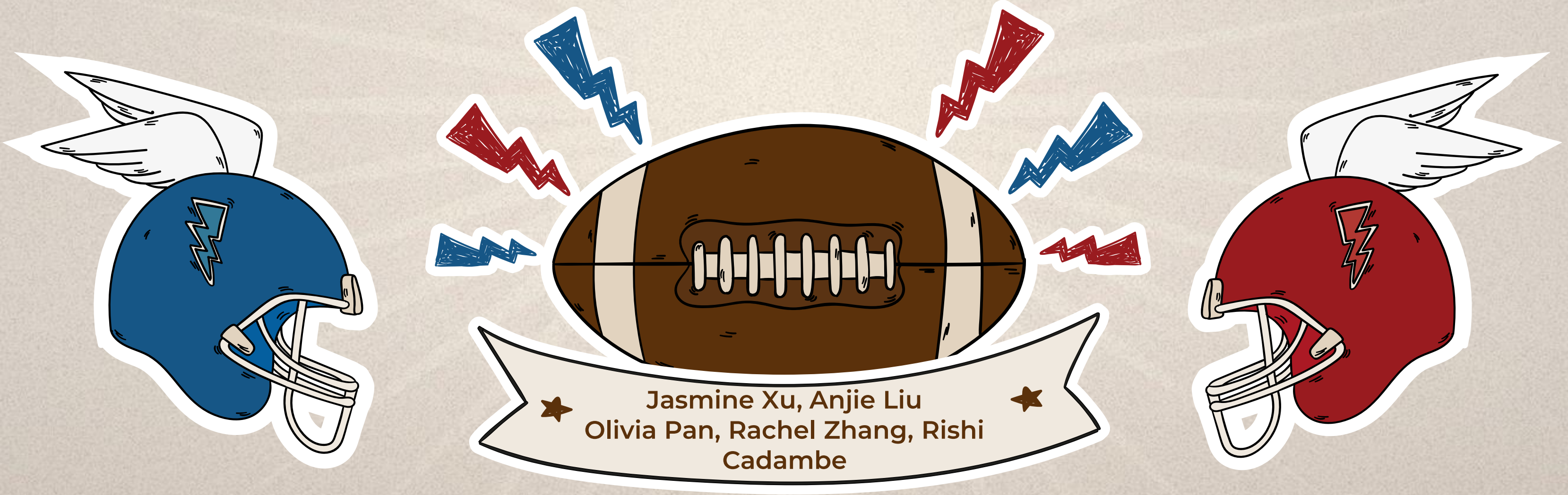
# Final Presentation



Jasmine Xu, Anjie Liu
Olivia Pan, Rachel Zhang, Rishi Cadambe


# What features contribute most significantly to comment engagement?

*What* or *how* should I write to attract more views?

# Data Description

**Features:**

- **win probability**
- **keyword densities**: the proportion of keywords within a comment
- **game time**: how far in the game was the comment made
- **comment length**
- **comment sentiment** (encoded via VADER) - 4 scores per comment: Positive, Negative, Neutral, and Compound
- **flair**: true or false
    - a user with a flair might add more credibility to the comment

**Target**

- **votes**: absolute value

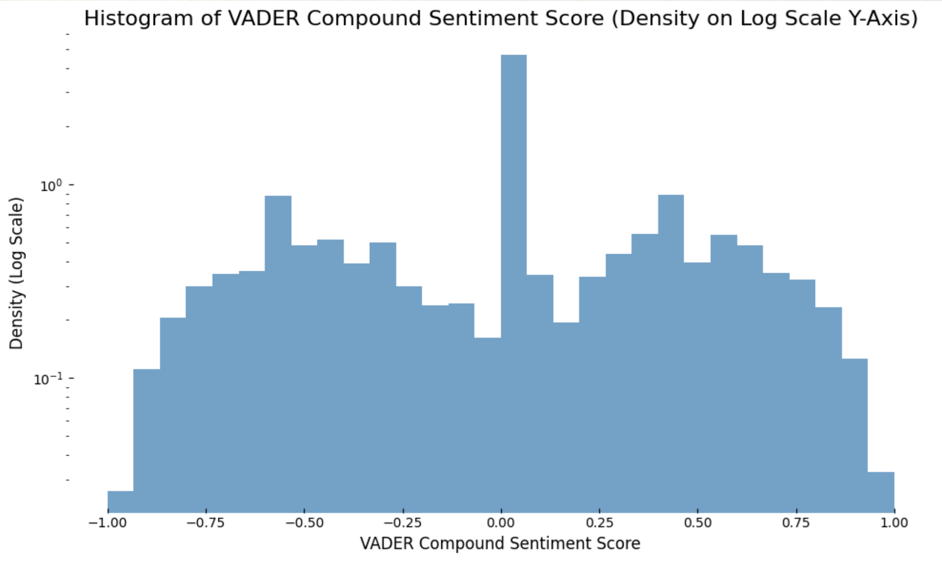
# Feature Engineering: Vader

**VADER** (Valence Aware Dictionary and sEntiment Reasoner)

- a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media (Hutto 2014).

We use VADER to get **4 types of scoring** for each comment:

- **compound**: [-1 (most extreme negative), +1 (most extreme positive)]
- **pos**, **neu**, and **neg**: ratios for proportions of text that fall in each category, [0, 1]
  - pos + neu + neg ~= 1



# Feature Engineering: Keywords

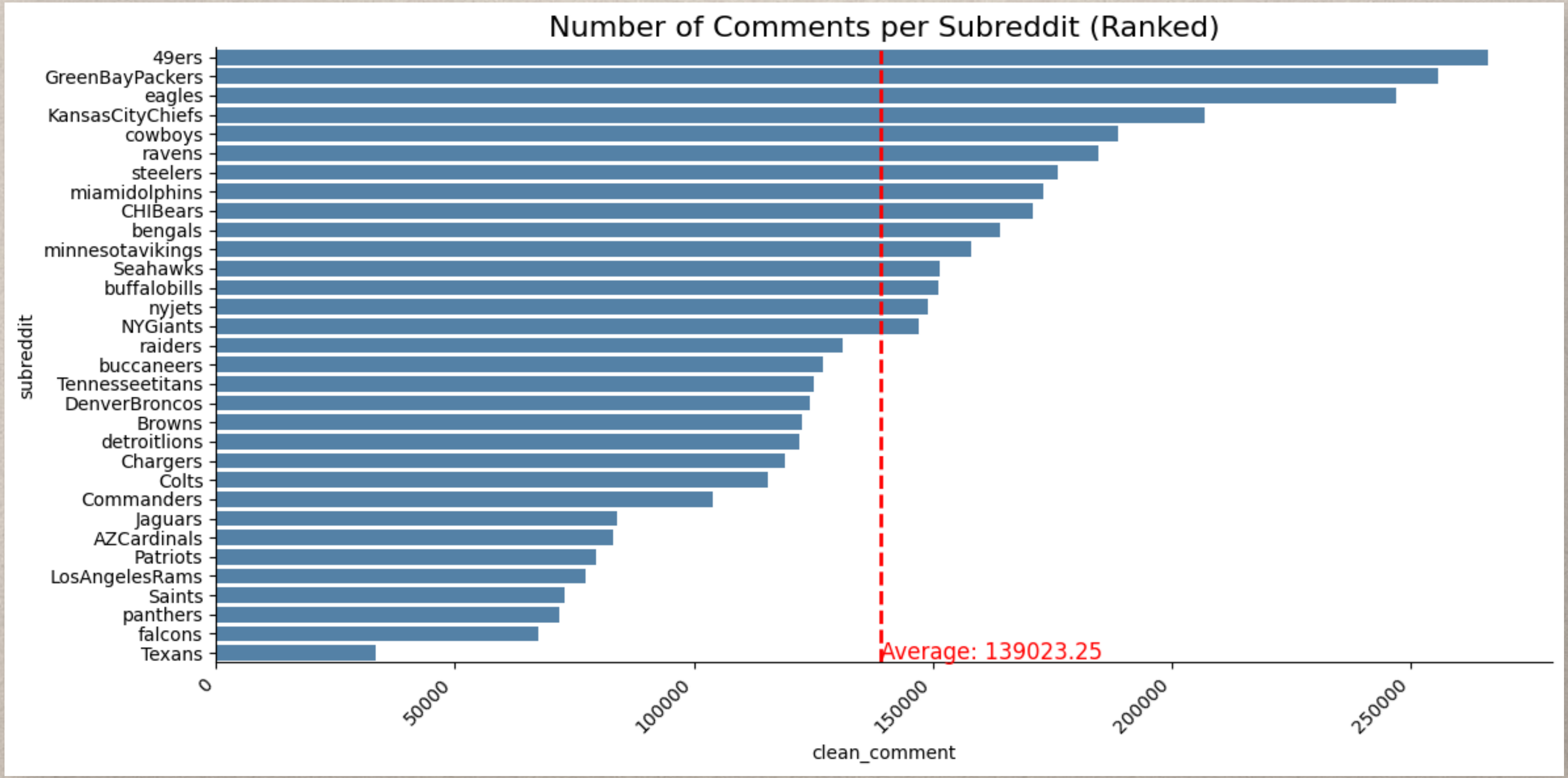
The presence of keywords may affect the number of upvotes a post receives. Keywords were selected by computing the correlation between their occurrence and the number of votes, and we computed keyword densities to use as features.

- **General keywords:** common across all subreddits
  - Game-related: “offense”, “defense”, “team”, “win”, “play”, “refs”, “holding”
  - Profanity: “fucking”, “fuck”, “bullshit”, “ass”
- **Names:** player names commonly referred to in a given subreddit’s comments
  - ["josh", "brock", "purdy", "jimmy", "kyle", "deebo", "lance", "aiyuk", "nick", "bosa", "trent", "mcglinchey", "brady", "trey", "williams", "shanahan", "shanny", "hufanga", "johnson", "greenlaw", "trey lance"]


Word cloud for the 49ers

# Total 4,448,744 -> 49ers 393,581 Comments

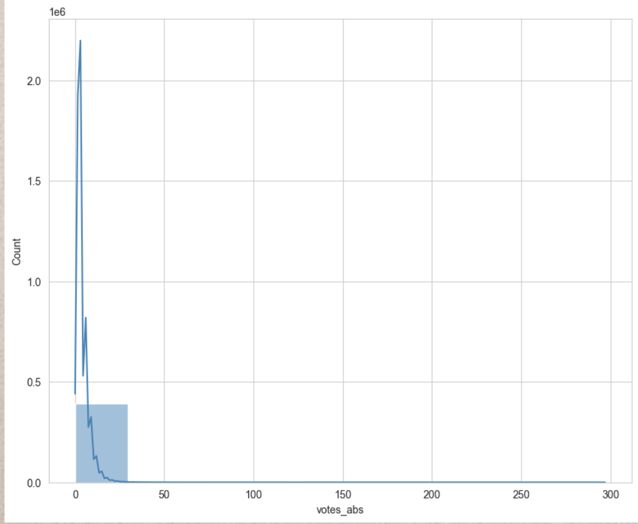
## Splitting by teams allows us to focus on comment engagement instead of group popularity



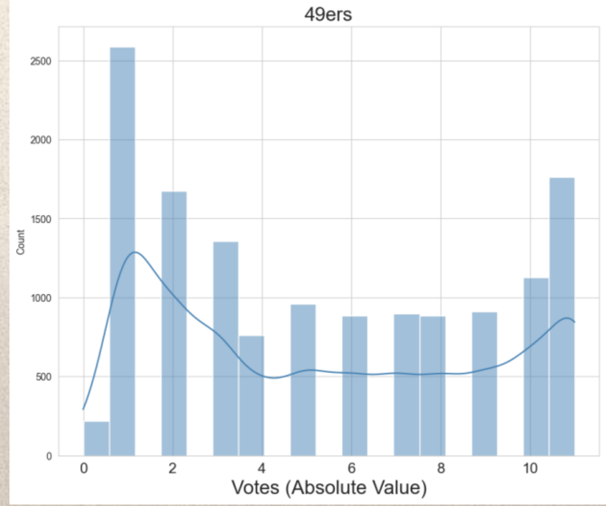
# Handling Outliers

- Looking at 49ers, we have an extremely skewed distribution of number of votes.
- This strongly influenced our model accuracy (r-square of 0.05), so we decided to remove outliers of votes_abs based on IQR
- Even though the distribution is still skewed, we chose a model that is robust to different distribution

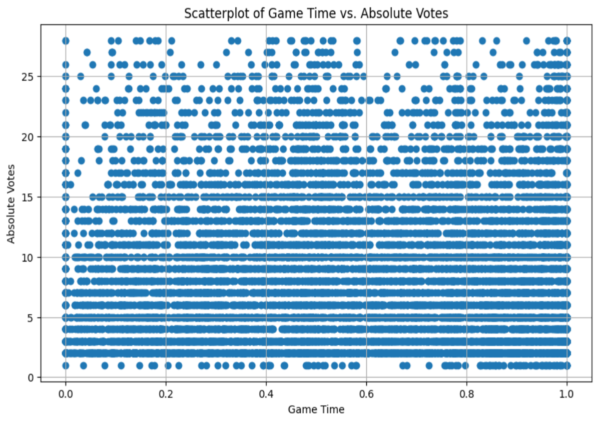
**before**

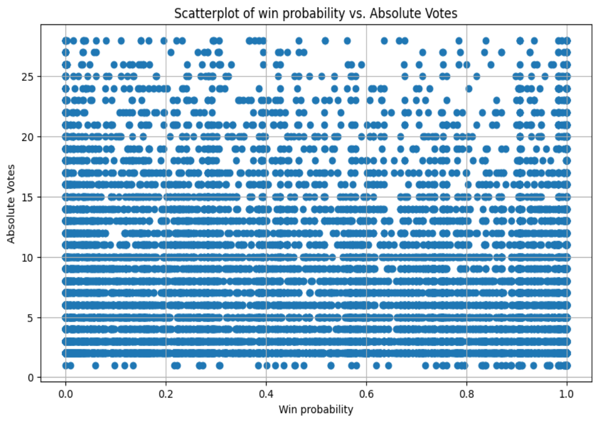


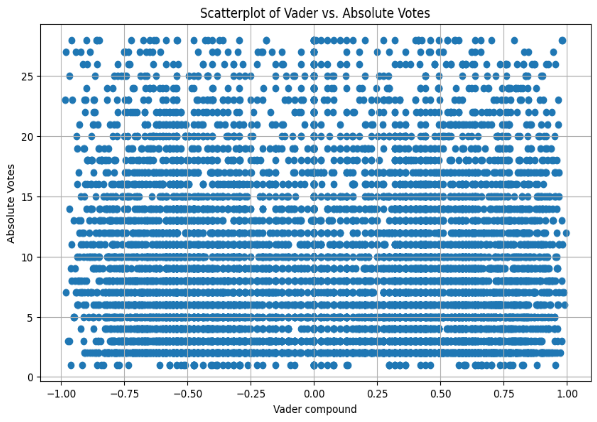

**after**

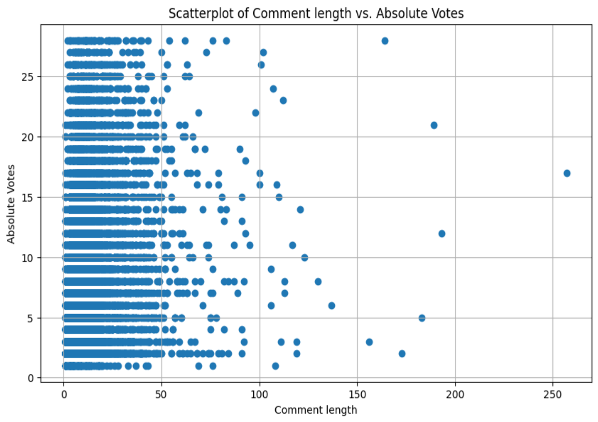


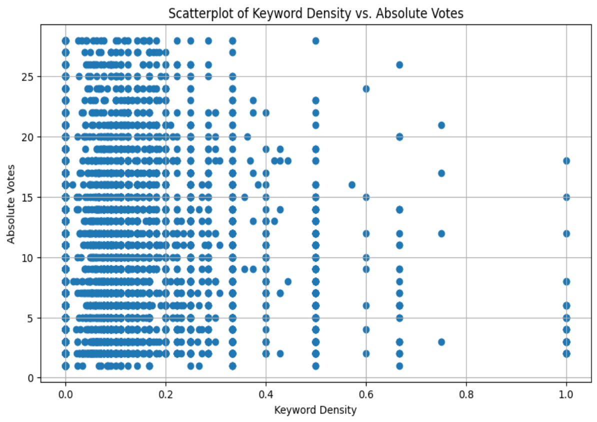
# Task difficulty



# Models: Pros vs Cons

| | | |
|---|---|---|
| **Random Forest Regression** | **- Handles non-linear relationships well.**<br>**- Robust to outliers and overfitting.**<br>**- Works well with high-dimensional data** | - Less interpretable than simpler models.<br>- May require hyperparameter tuning for optimal performance |
| **Neural Network** | - Capable of capturing complex, non-linear relationships.<br>- Scalable to large datasets with advanced architectures | **- Requires significant computational resources.**<br>- Prone to overfitting without proper regularization |
| **Logistic Regression** | - Simple, fast, and interpretable<br>- Requires fewer computational resources | - Assumes linearity between input features and log-odds.<br>**- Struggles with complex, non-linear relationships** |

## Modeling Approach

# Random Forest Regression

**Reasons**: does not assume sample distribution (i.e. linear relationship), can handle correlated features, less prone to overfitting, good for analyzing feature importance.

**I.I.D. Assumption:** data are drawn from the same probability distribution.
We trained the comments from 49er subreddits because it has the most comments in the dataset.

# Evaluation Strategy

1. **Train-Test Split**

   64/20/16

2. **Hyper-Parameter Tuning**

   Grid Search

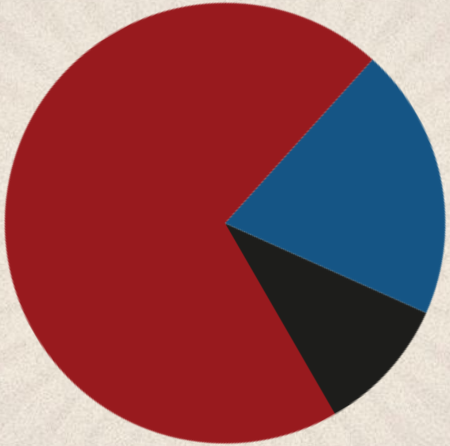
3. **Cross Validation**

   5-Fold Cross Validation

4. **Utility Maximization**

   MSE, MAE, R2

# Train-Test Split



**64%**

**Training Set**

Train the model and estimate model parameters

**20%**

**Testing Set**

Final evaluation

**16%**

**Validation Set**

(implicitly created in cross validation) Tune hyper-parameters

One user can post multiple comments in a subreddit. To avoid data leakage in the testing set, **we chose to divide the dataset by unique users instead of comments.** This prevents data leakage and aligns with our project goal.

# Cross-Validated Grid Search

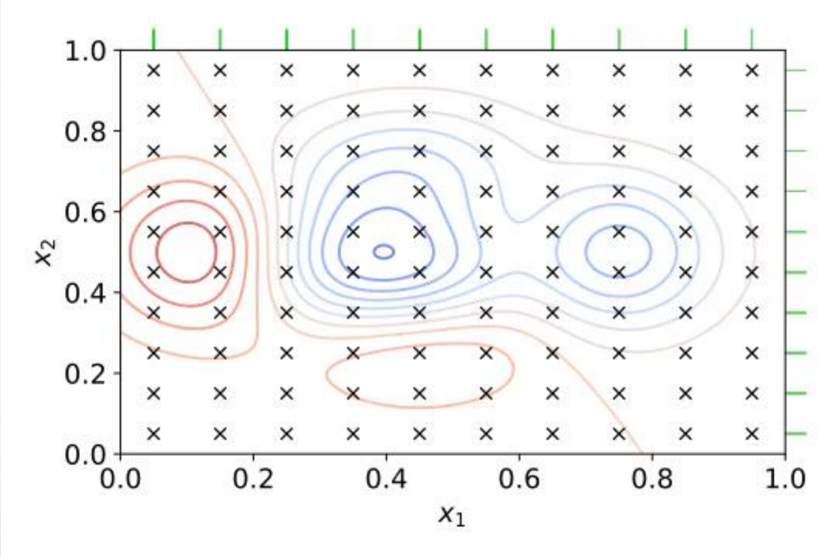
**Search over multiple combinations of different hyperparameters:**

- max_depth: [None, 10, 20]
- min_samples_leaf: [1, 2, 4]
- min_samples_split: [2, 5, 10]

**5-fold CV ensures results generalize across data:**

- Splits data into 5 parts
- Trains on 4, tests on 1

We tune hyperparameters because Random Forest models are often sensitive to parameters



# Utility Maximization

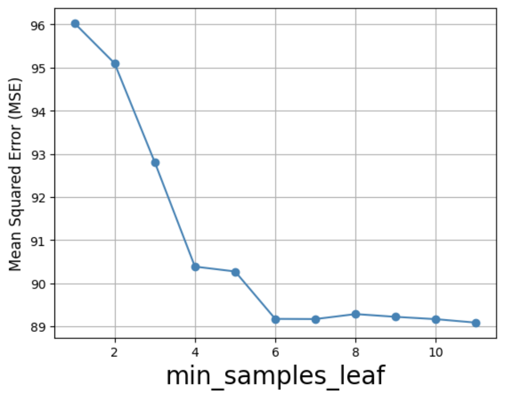
We focused on **minimizing MSE** when tuning the hyper-parameters.

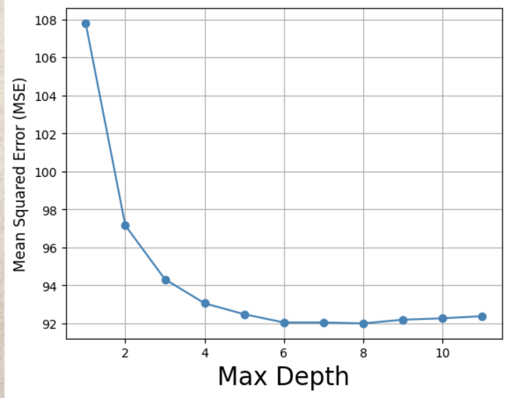
max_depth=10

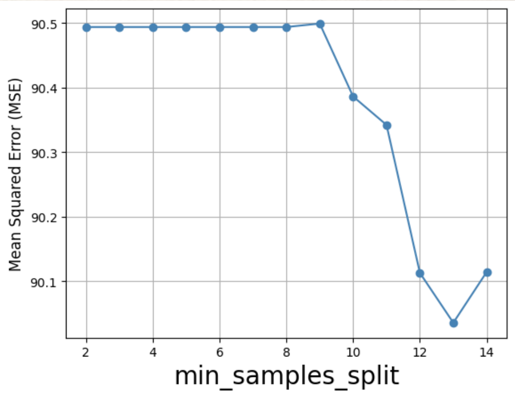
min_samples_split=5

min_samples_leaf=2

n_estimators=200



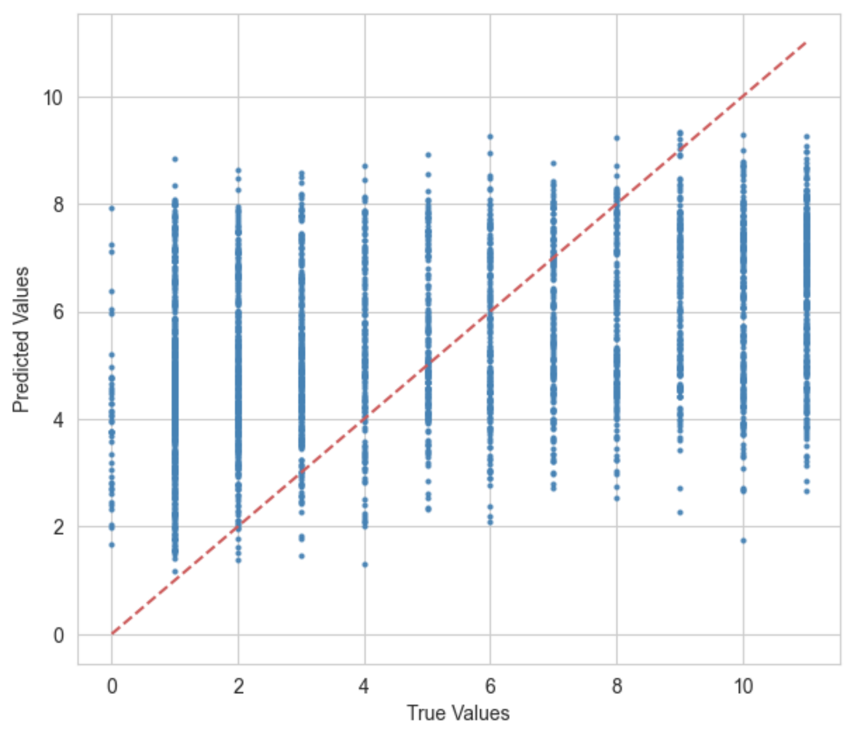
# Results

MSE = 85.62

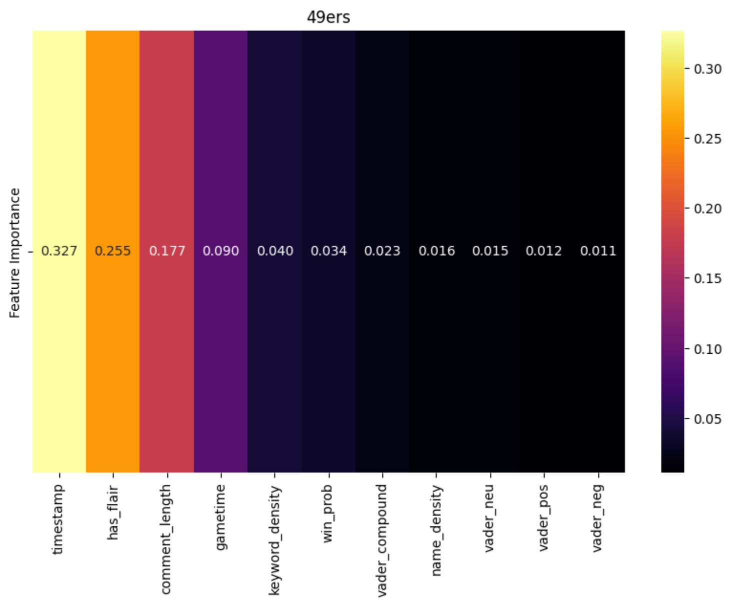
RMSE = 9.25

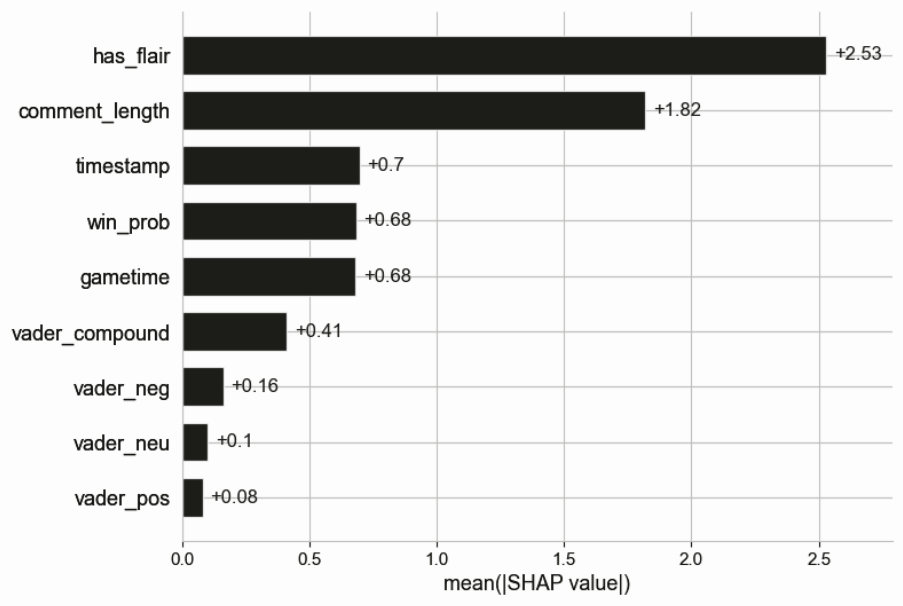
MAE = 5.78

$R^2$ = 0.29



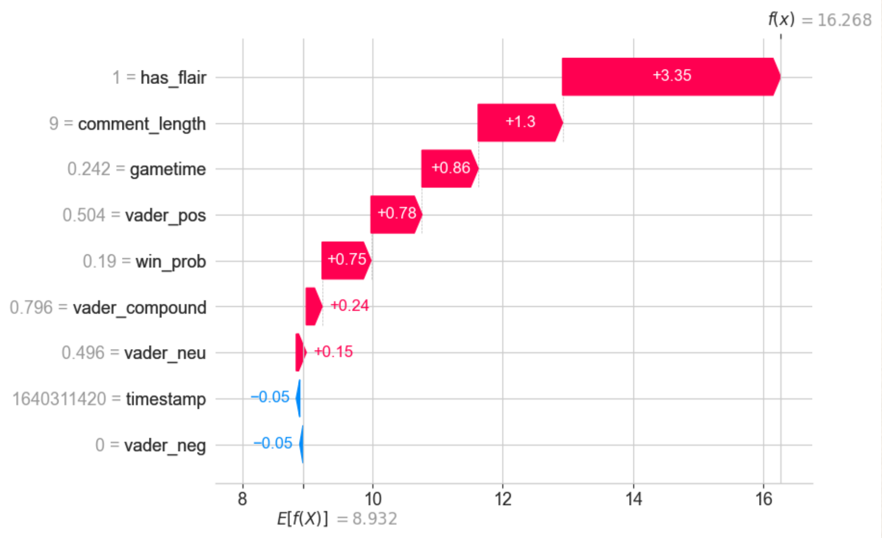
# Results



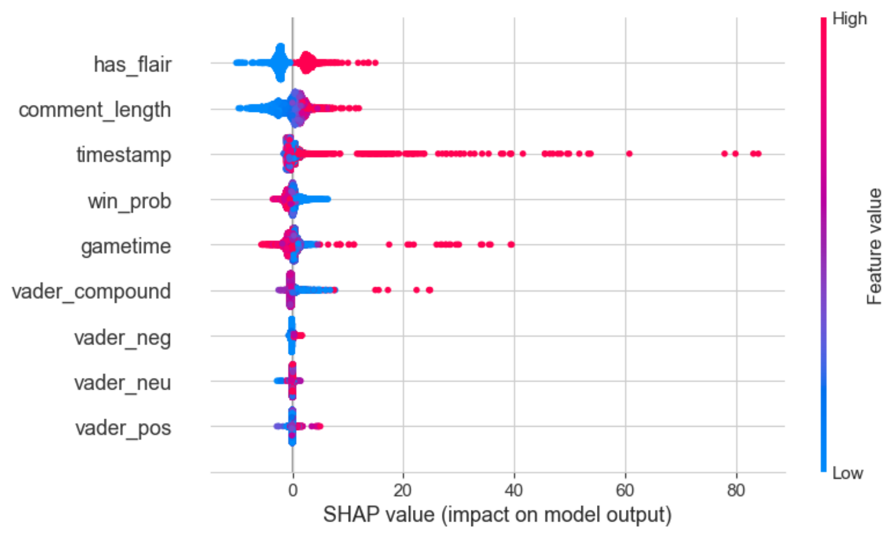




Feature importance and the mean SHAP value share similar results: timestamp, gametime, and comment length are viewed as more important than VADER scores.

# SHAP



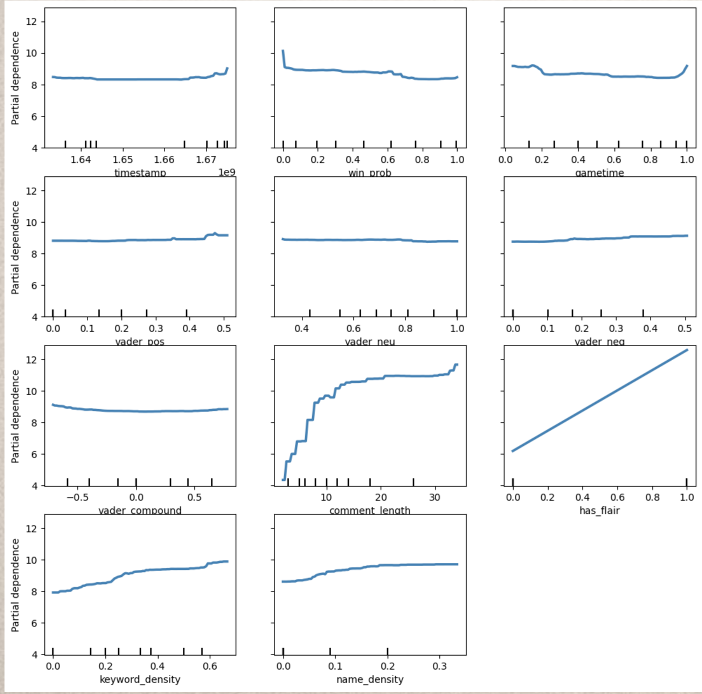
f(x) = 16.268
1 = has_flair
9 = comment_length
0.242 = gametime
0.504 = vader_pos
0.19 = win_prob
0.796 = vader_compound
0.496 = vader_neu
1640311420 = timestamp
0 = vader_neg
+3.35
+1.3
+0.86
+0.78
+0.75
+0.24
+0.15
−0.05
−0.05
8
10
12
14
16
E[f(X)] = 8.932

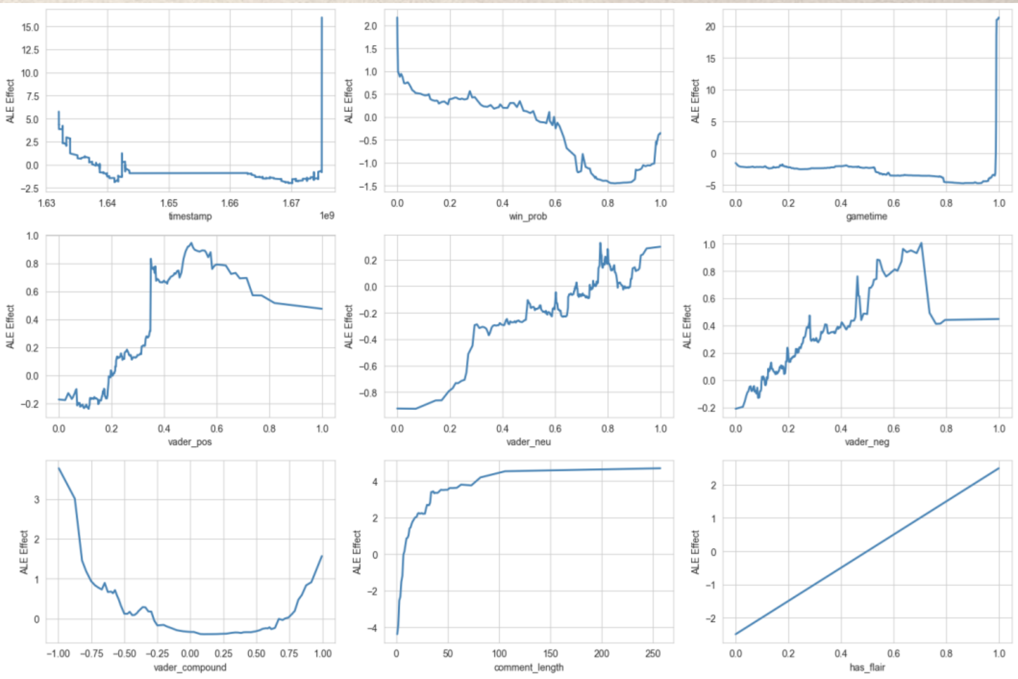



has_flair
comment_length
timestamp
win_prob
gametime
vader_compound
vader_neg
vader_neu
vader_pos
0
20
40
60
80
SHAP value (impact on model output)
High
Low
Feature value

# Interpretation

- **PDP and ALE plots** showing the **marginal effect** of a feature on the model's prediction.



# Significance

- Even though R2 increases after feature engineering, the results are still not great.
  - Our variables (time, sentiments, common length) might not be the best indicators of whether a post will go viral.
  - There might be other variables that have a strong influence on number of votes that we didn't consider: ex. user popularity
- Looking at feature importance, still, **our variables explain the result to some extent**
  - Whether you post at the right time and whether you have a flair (tag) **tend to be more significant than the exact contents of your post**

# Future Applications

- We have **32 datasets, one for each subreddit (team)**
- **Each subreddit has a different data distribution** (recall that last time when we were presenting feature importance, subreddits have the highest contributions)
- We could find we want to find a way to generalize our result to all subreddits, or **find features that are important across all subreddits**
- To do this, we could create **32 different models**, and we will collect the feature importance from each model and present a more generalized solution to our inference problem

# Thank you!