Project 3
Web scraping & classification

Zayn Melhem • 08.06.2022

Problem Statement

Scenario

To test and see what model will predict the best score.

- **I have chosen these models and will find the best score among them.**
- RandomForestClassifier
- KNeighborsClassifier
- LogisticRegressionCV
- GradientBoostingClassifier

- You're fresh out of your Data Science bootcamp and looking to break through in the world of freelance data journalism. Nate Silver and co. at FiveThirtyEight have agreed to hear your pitch for a story in two weeks!

- Your piece is going to be on how to create a Reddit post that will get the most engagement from Reddit users. Because this is FiveThirtyEight, you're going to have to get data and analyze it in order to make a compelling narrative.

- As you are well aware trying to make a hot post with the right characteristics on reddit is difficult with so many factors yet just for a simple post, If only there was a way to gather data and predict a potential hot post to make.

- Oh! Luckily I so happened to have the solution. An api scraper that feeds data from reddit into a few machine learning models I created that predict important key words and subreddits that are in trending hot posts!

How I tackled it

Stage 1

- I made a while/for loop script with API web scraper from praw to collect as many hot posts available every 3 hours.
- I import the scraped data into df's and perform cleaning (dropping emojis ect) & eda.
- I create my X['features'] for the model while lemmatizing, makes multiples words to the base word

Stage 2

- Instantiate Tfid Vectorizing: Common words are penalized
- Eliminate 'stop_words' to improve our analysis
- join dfs while dropping columns.

Stage 3

- Creating y['target'] baseline = 51.43%. y = num_comments + median of num_comments is above 51.43% is hot and below is not hot.
- My models:

- RandomForestClassifier
- KNeighborsClassifier
- LogisticRegressionCV
- GradientBoostingClassifier

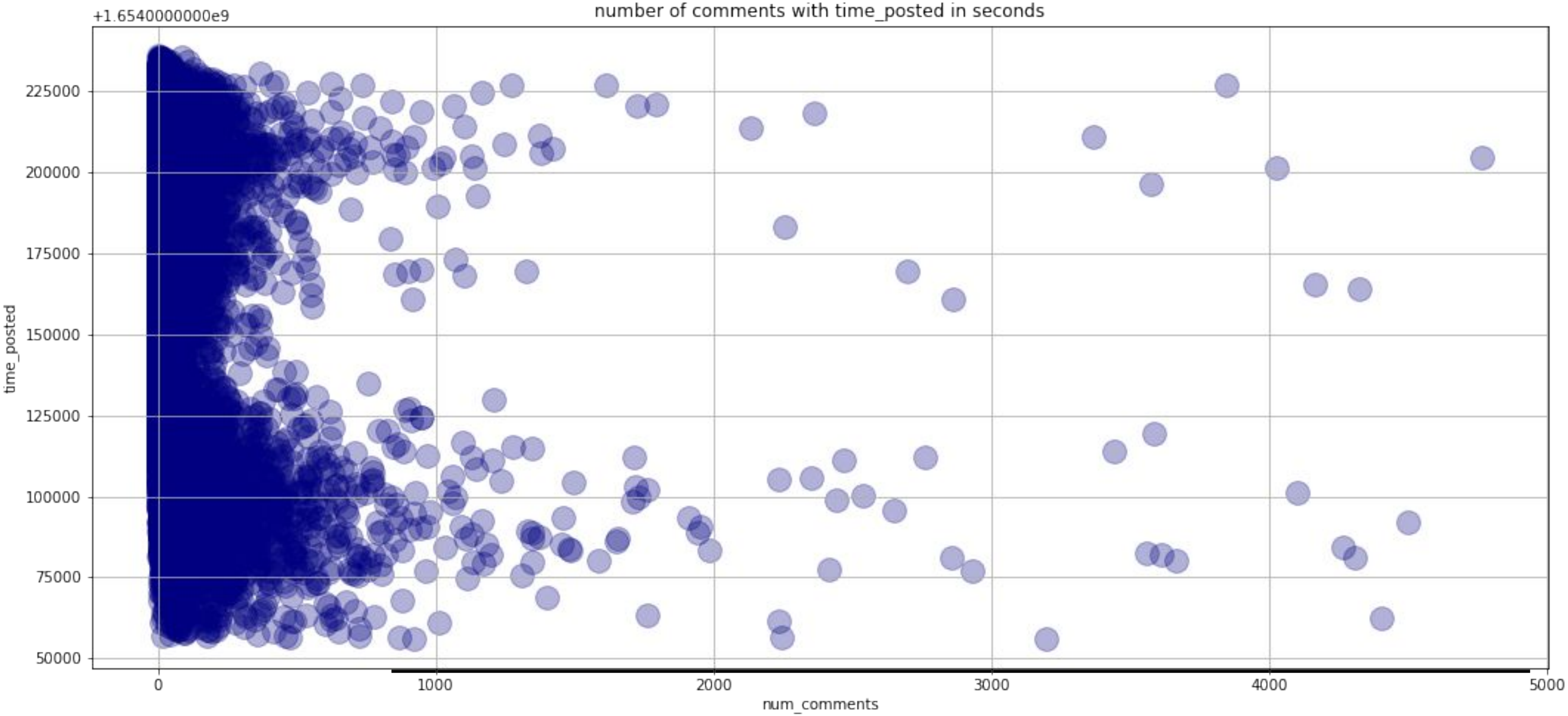
Overall scores of each model

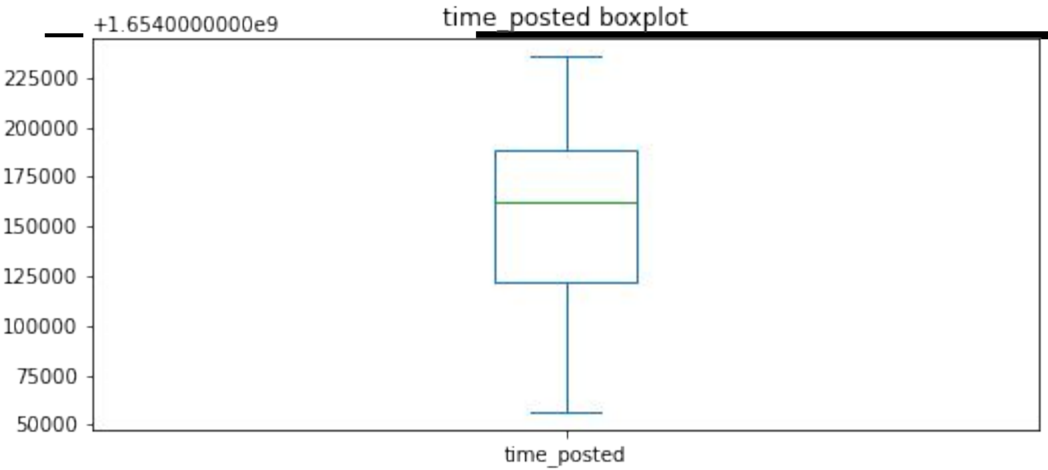
Overall scores

- RandomForest model correctly predicted 0.73% overall score of posts.
- GradientBoostingClassifier model correctly predicted 0.67% overall score of posts.
- KNeighborsClassifier model correctly predicted 0.71% overall score of posts.
- LogisticRegressionCV model correctly predicted 0.63% overall score of posts.

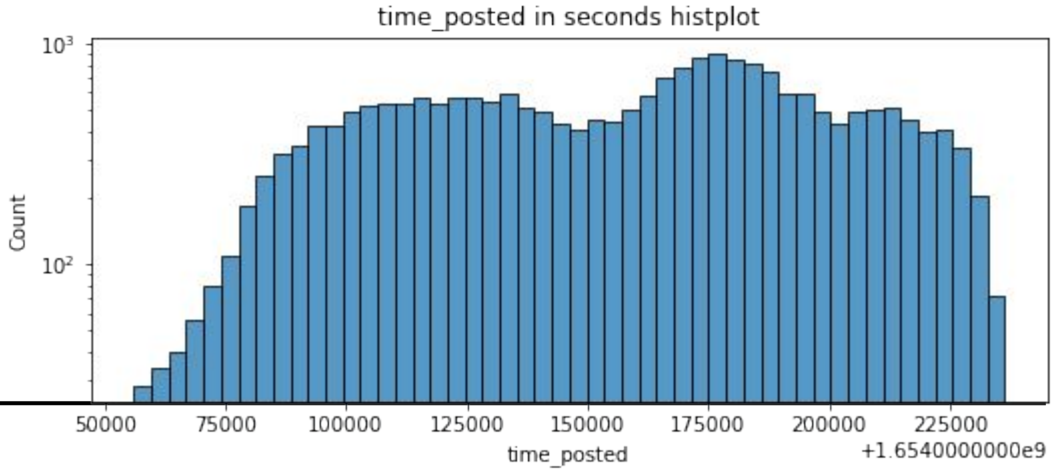
Visualizations and Interpretations

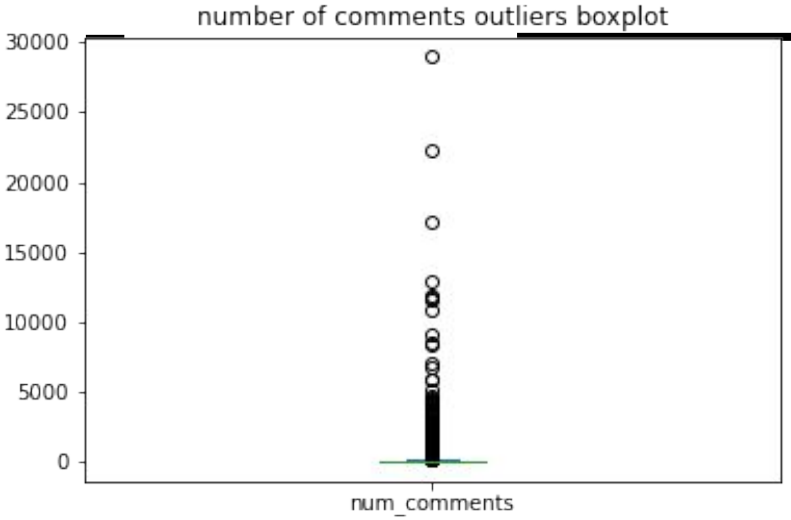
- The time_posted display number of seconds elapsed in unix epoch.
- It appears that time_posted correlation with num_comments, yet it appears that the longer a post has been up does not affect the increase of num_comments after a certain amount of time elapsed.



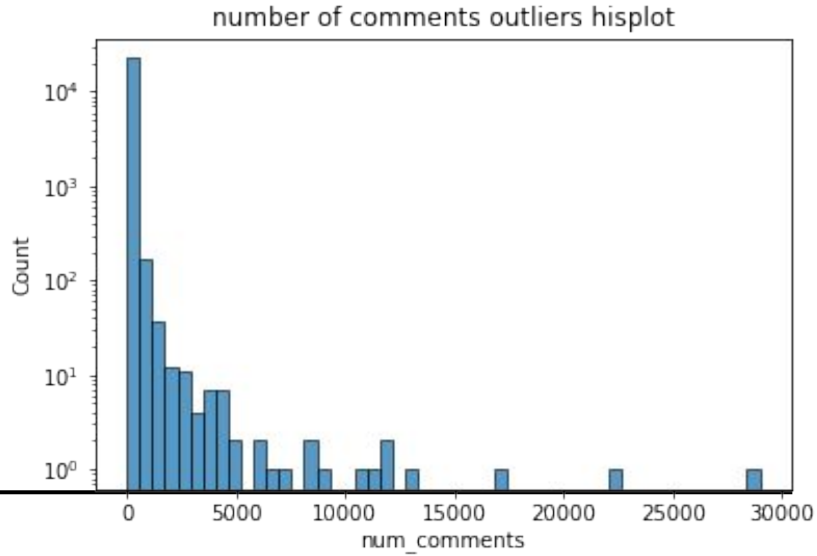


Distribution has definite changes with less occurrences. The shape of the distributions have changed as well while still keeping a near consistent shape pattern. There is a big spike around 175000 seconds which is around 48 hours after the post has been up they get more comments around the 2 day mark.





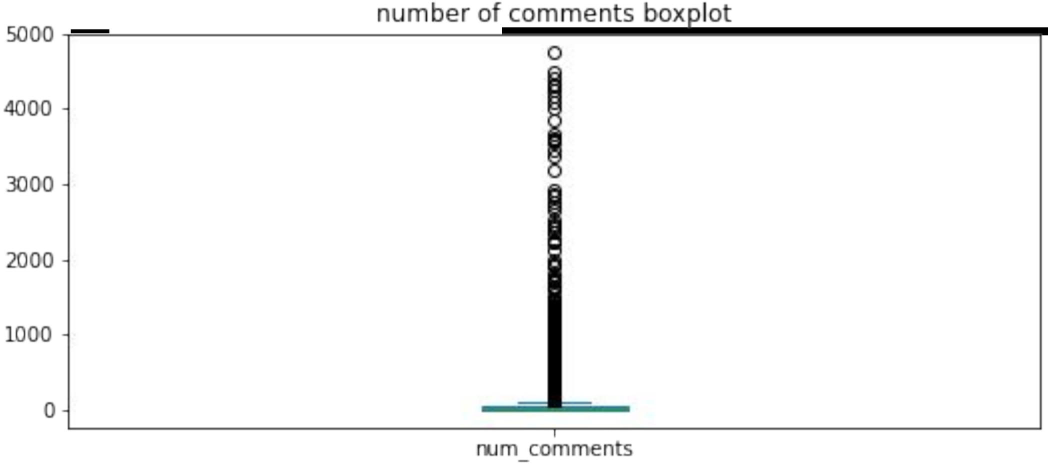
number of comments outliers boxplot

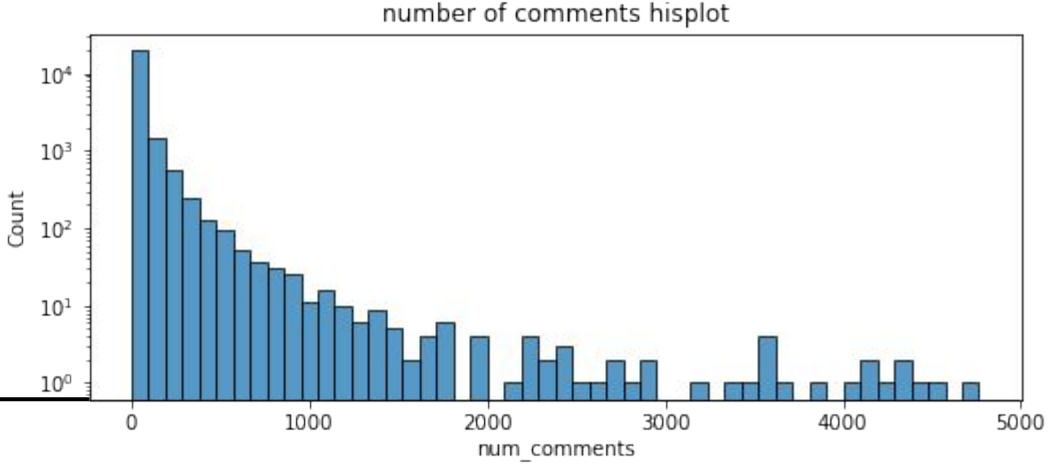


number of comments outliers hisplot

Visuals for number of comments with outliers/

5000+ equals outliers .

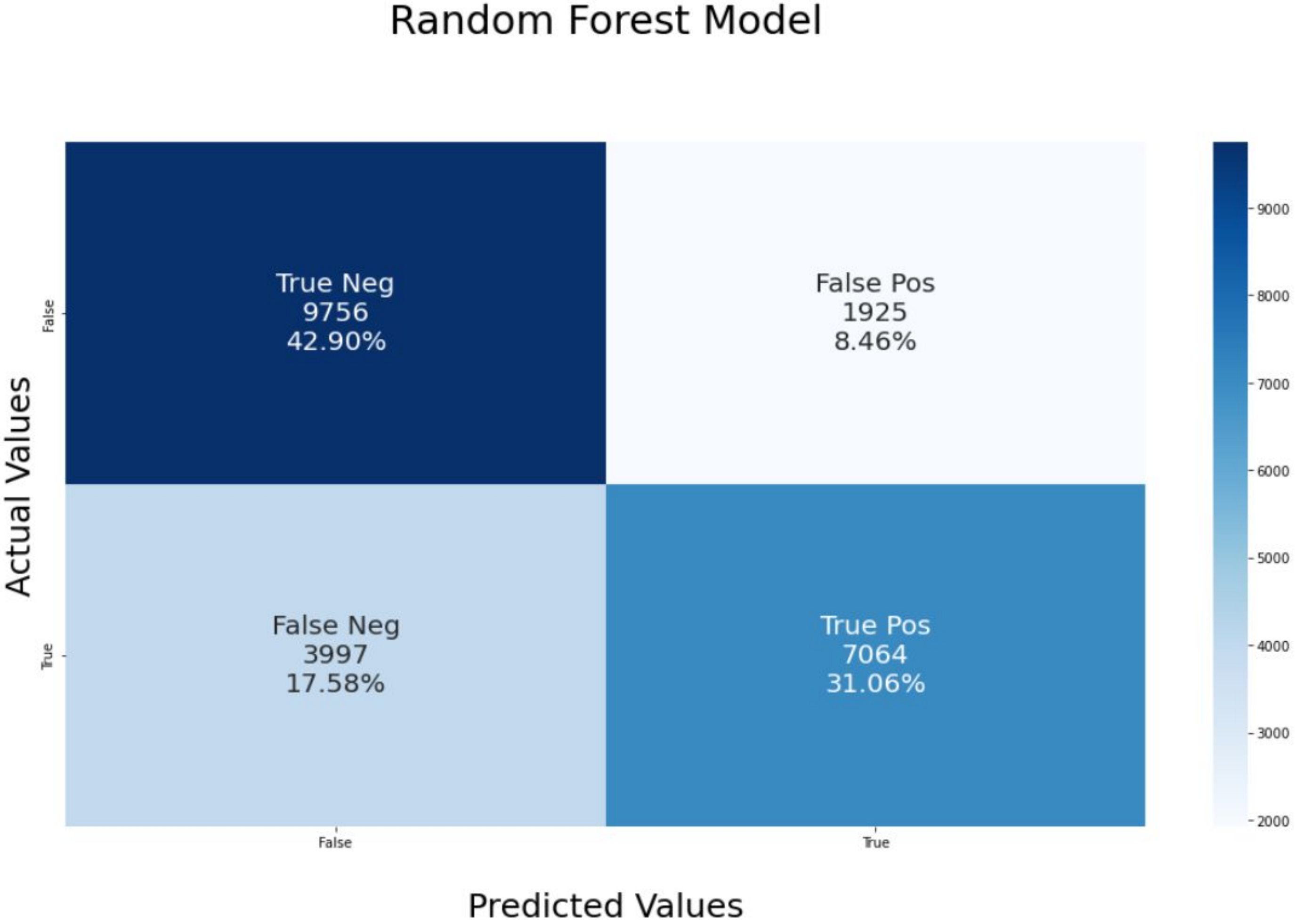




Visuals for number of comments with outliers filtered out.

5000 or less is non outliers.

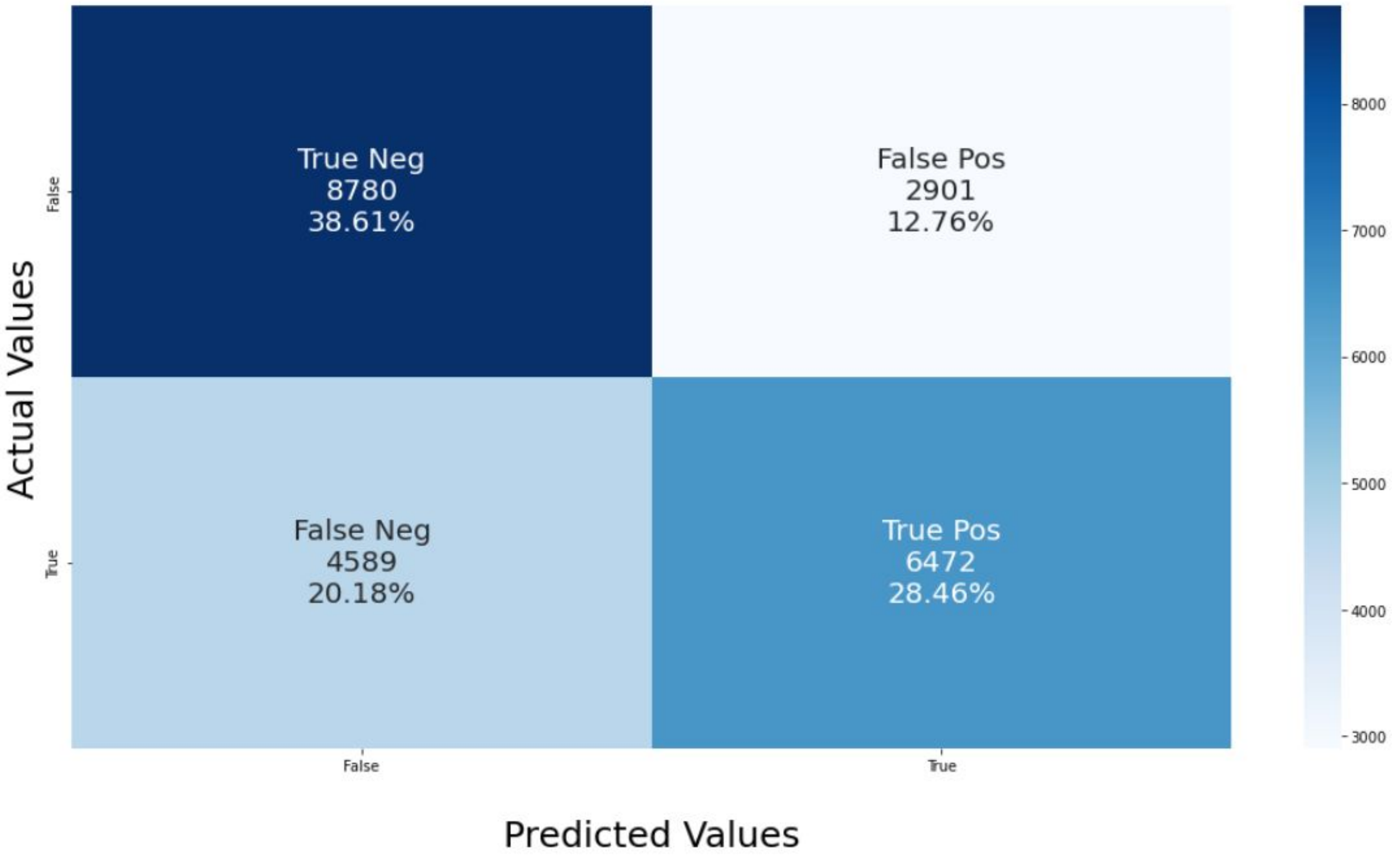
Confusion matrix and scores



Random Forest

- Our RandomForest model correctly predicted 84% of the not-hot posts.
- Our RandomForest model correctly predicted 62% of the actual accuracy of hot posts.
- Our RandomForest model correctly predicted 73% overall score of posts.

Gradient Boosting Classifier Model



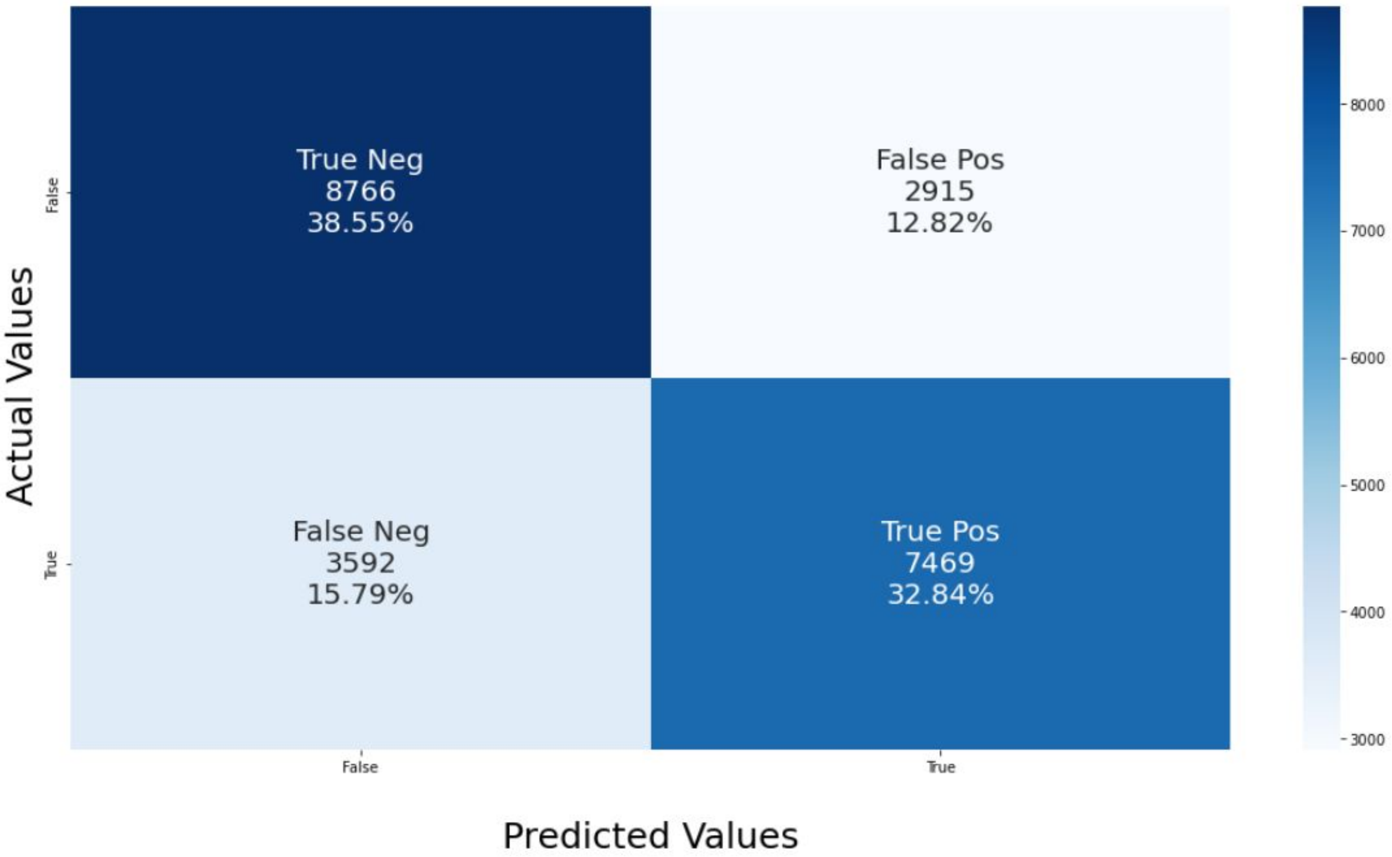
Gradient Boosting

- Our GradientBoostingClassifier model correctly predicted 77% of the not-hot posts.

- Our GradientBoostingClassifier model correctly predicted 57% of the actual accuracy of hot posts.

- Our GradientBoostingClassifier model correctly predicted 67% overall score of posts.

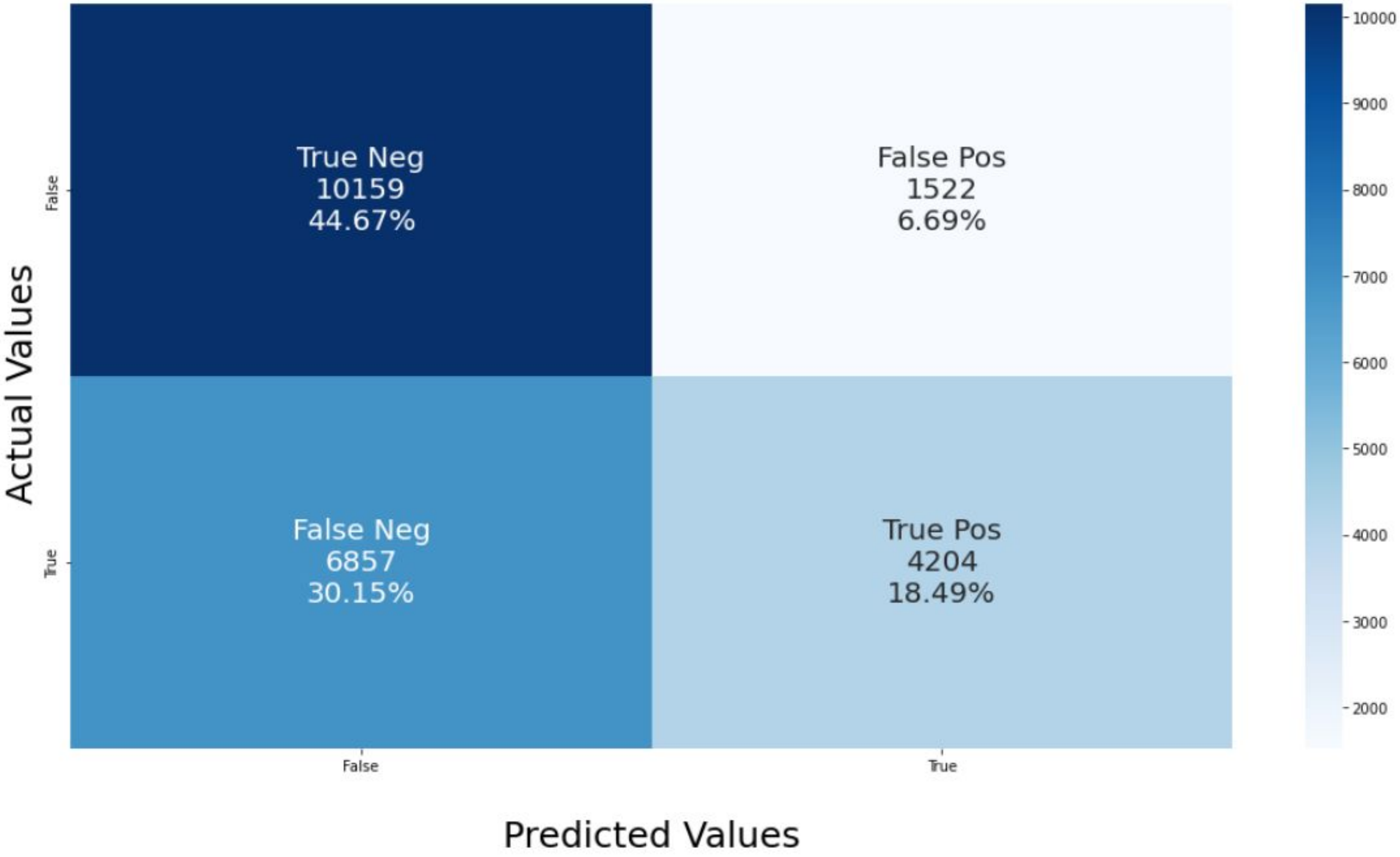
KNeighbors Classifier Model



KNeighborsClassifier

- Our KNeighborsClassifier model correctly predicted 75%of the not-hot posts.
- Our KNeighborsClassifier model correctly predicted 67% of the actual accuracy of hot posts.
- Our KNeighborsClassifier model correctly predicted 71% overall score of posts.

Logistic Regression Model



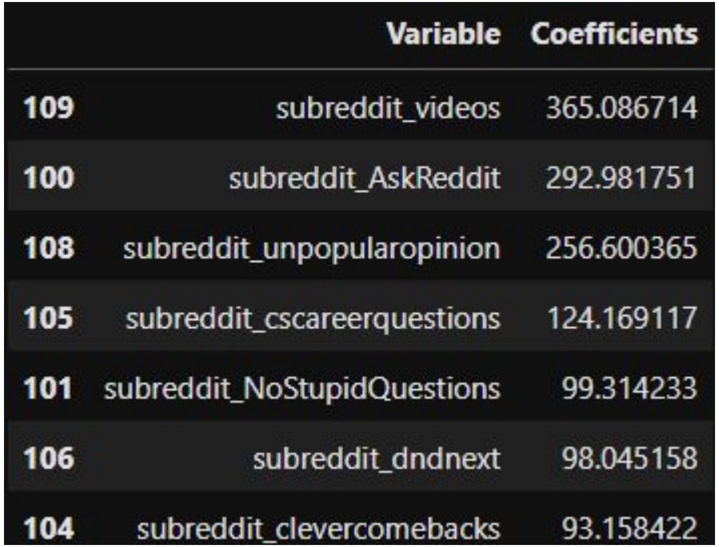
Logistic RegressionCV

- Our LogisticRegressionCV model correctly predicted 90%of the not-hot posts.
- Our LogisticRegressionCV model correctly predicted 34% of the actual accuracy of hot posts.
- Our LogisticRegressionCV model correctly predicted 63% overall score of posts.

The best among all 4 models is... Gradient Boosting Model! It has predicted the most accurate score.

- **LogisticRegressionCV seem to have over fit & Gradient Boosting curbs overfitting & trains faster.**
- Our GradientBoostingClassifier model correctly predicted 76% of the not-hot posts.
- Our GradientBoostingClassifier model correctly predicted 58% of the actual accuracy of hot posts.
- Our GradientBoostingClassifier model correctly predicted 67% overall score of posts.Our GradientBoostingClassifier model correctly predicted 76% of the not-hot posts.

Importance variables in Logistic Regression



	Variable	Coefficients
109	subreddit_videos	365.086714
100	subreddit_AskReddit	292.981751
108	subreddit_unpopularopinion	256.600365
105	subreddit_cscareerquestions	124.169117
101	subreddit_NoStupidQuestions	99.314233
106	subreddit_dndnext	98.045158
104	subreddit_clevercomebacks	93.158422

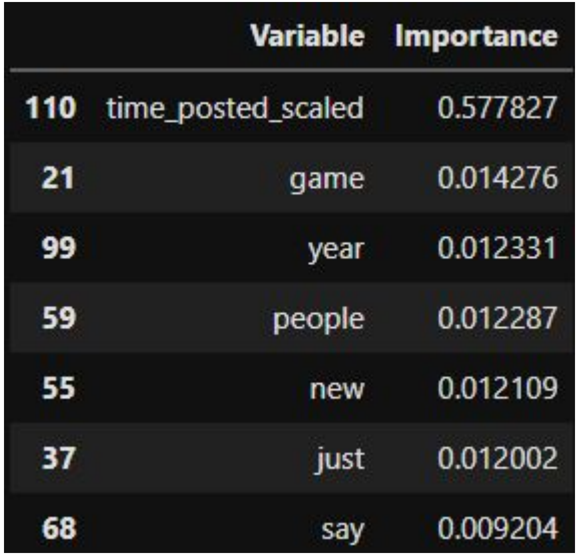
LOGREG interpretation

- The top coefs push you to the most POSITIVE CLASS not the most powerful(importance)

- For each unit change in the variable, the positive class is that times more likely.

- The subreddit_title are more important than words in titles.

Importance variables in Random Forest

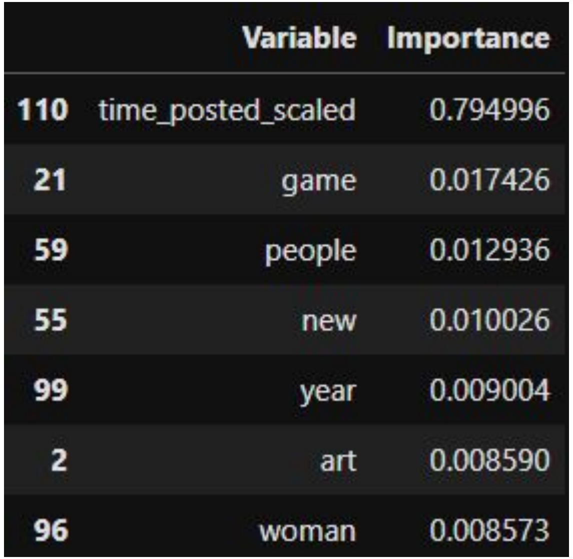


	Variable	Importance
110	time_posted_scaled	0.577827
21	game	0.014276
99	year	0.012331
59	people	0.012287
55	new	0.012109
37	just	0.012002
68	say	0.009204

RF interpretation

- We just know they were important in making the decision.
- These features may have importance related to a game going on.
- The key words game & people are one of the top features trending the day we scraped for posts, there was an NBA game and other sporting events on while I was only scraping data for that one day.
- time_posted seems to be correlated towards the time a post was posted during trending subreddit such as the words people & games, as an NBA game possibly was on during this time period.
- Man, year, say & new was also important key words that were in subreddits hot posts I assume they could be correlated to all sporting events that happened the day we scraped.

Importance variables in Gradient Boosting Classifier



	Variable	Importance
110	time_posted_scaled	0.794996
21	game	0.017426
59	people	0.012936
55	new	0.010026
99	year	0.009004
2	art	0.008590
96	woman	0.008573

GB interpretation

- Like Random forest these are importances not like the Coefficients.
- time_posted is once again the most important variable in correlation to game & possibly correlation to people.
- Sporting events were trending around the same time so it is correlated to game, people, woman & time_post when I was scraping the data.

Conclusion & Recommendations

- Based on my problem statement and work, I have concluded, That my model predicts what words make a post that get above the baseline of comments which are hot subreddit posts.
- For future iteration I will run Sigma(my data scrape script) alot longer as 1 day of scraping data isn't enough to use to train the model for scores.
- The model score of Not Hot post are more accurate than the Hot post score, I think I need more data in order to improve the models overall.
- time_posted doesn't seem to affect the num_comments a post have from the data collected and interpretations.
- Hot post typically trends for around a day then gets pushed down by new hot posts which seems to not effect num_comments in contrary to age of post.
- With the findings I have with my model scores and trend can determine which subreddit post company's can template and use to make FiveThirtyEight get more views and interaction on each new post.

- The models show promise with the data collected in such a very short period, with more time and data collected I can tune the models and make them much more efficient and accurate for hot posts.
- To better journalism spreading the news and reach as many people as possible. If your ready to increase your engagement and views of your media content then this is the product for you!