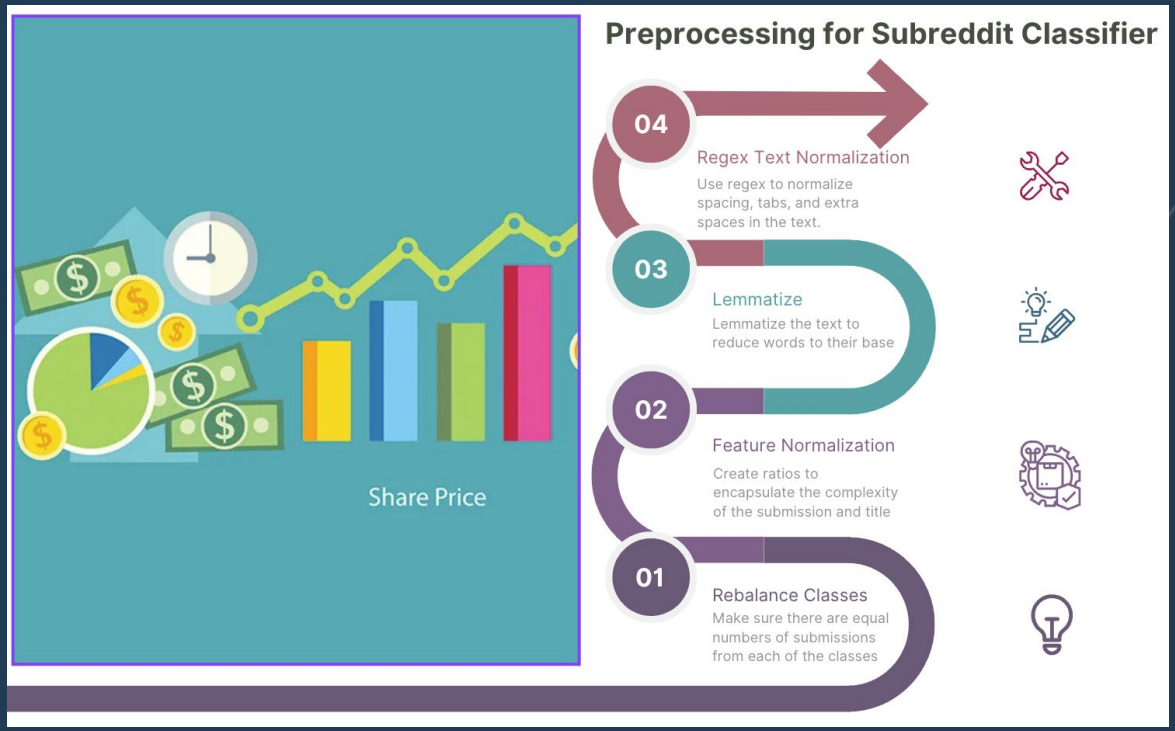
Subreddit Classification

Nicholas Sanso

Objective

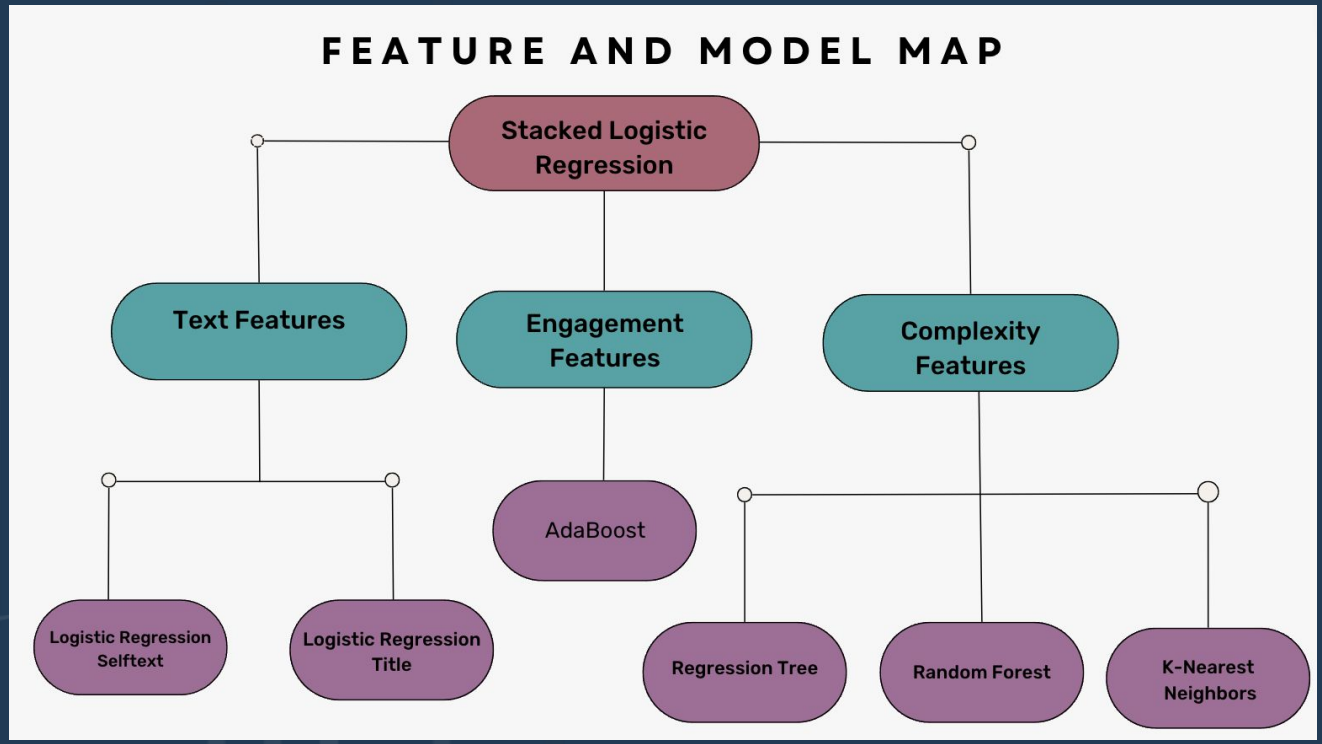
- Build a model that will use submission text complexity, submission engagement metrics, and submission term (word) frequency to distinguish which subreddit a submission was posted to (r_investing or r_stocks).

Preprocessing Workflow



Feature Mapping

- Features were grouped into sets seen in turquoise
- Models fitted on the feature sets



Engagement Metric Base Model Results

- Similar validation accuracies
- Consistent results in the cross validation folds

Metrics of Base Models	Knn	RF	Decision Tree
train accuracy	0.689	0.727	0.6811
validation accuracy	0.631	0.645	0.643
cross validation standard deviation	0.0136	0.0108	0.0090
cross validation mean	0.655	0.663	0.6626

Base Model Results

- Models built exclusively on title and selftext have nearly identical validation accuracy.
- Models built with TfidVectorizer are somewhat overfit.

Metrics of Base Models	Logistic selftext	Logistic title	AdaBoost
train accuracy	0.995	0.882	0.699
validation accuracy	0.772	0.743	0.729
cross validation standard deviation	0.0127	0.0142	0.0124
cross validation mean	0.761	0.730	0.654

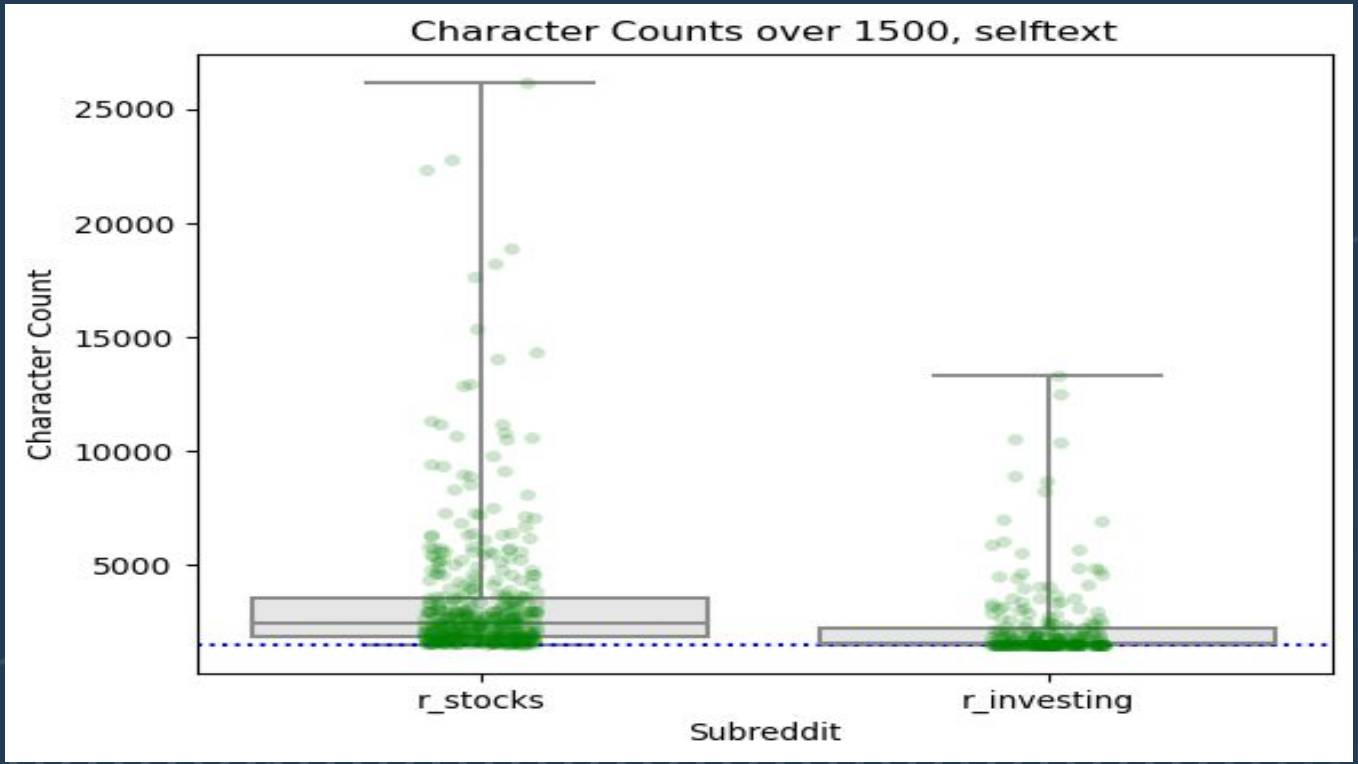
Stacked Model Results

- Both stacked models outperformed the base component models
- Stacked models showed no signs of overfitting

Metrics of Stacked Models	Stacked Logistic Regression	Stacked Adaboost
train accuracy	0.7763	0.7748
validation accuracy	0.7911	0.8022
cross validation standard deviation	0.0201	0.0195
cross validation mean	0.7770	0.7756

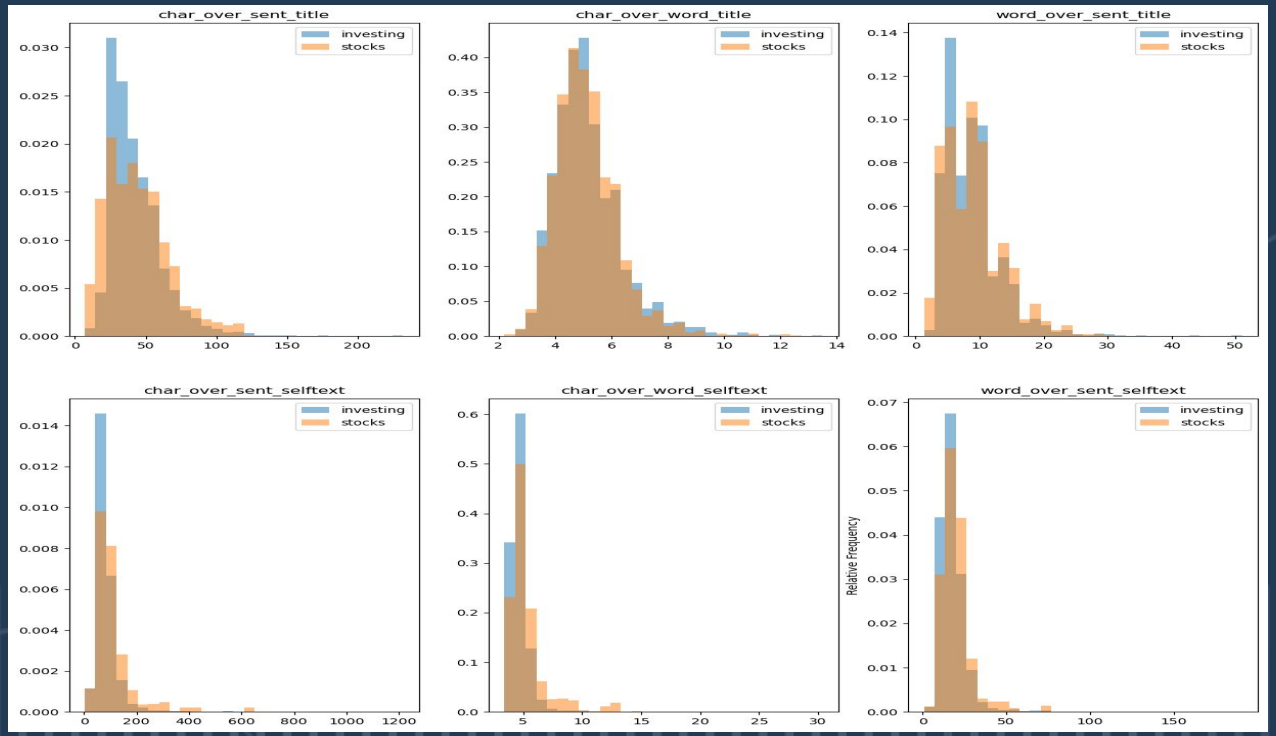
Submission Complexity

- r_stocks has more submissions with character counts in excess of 1500 submission than r_investing.
- First and third quartile of the r_investing box and whisker plot lies closer to the 1500 character count dotted line.



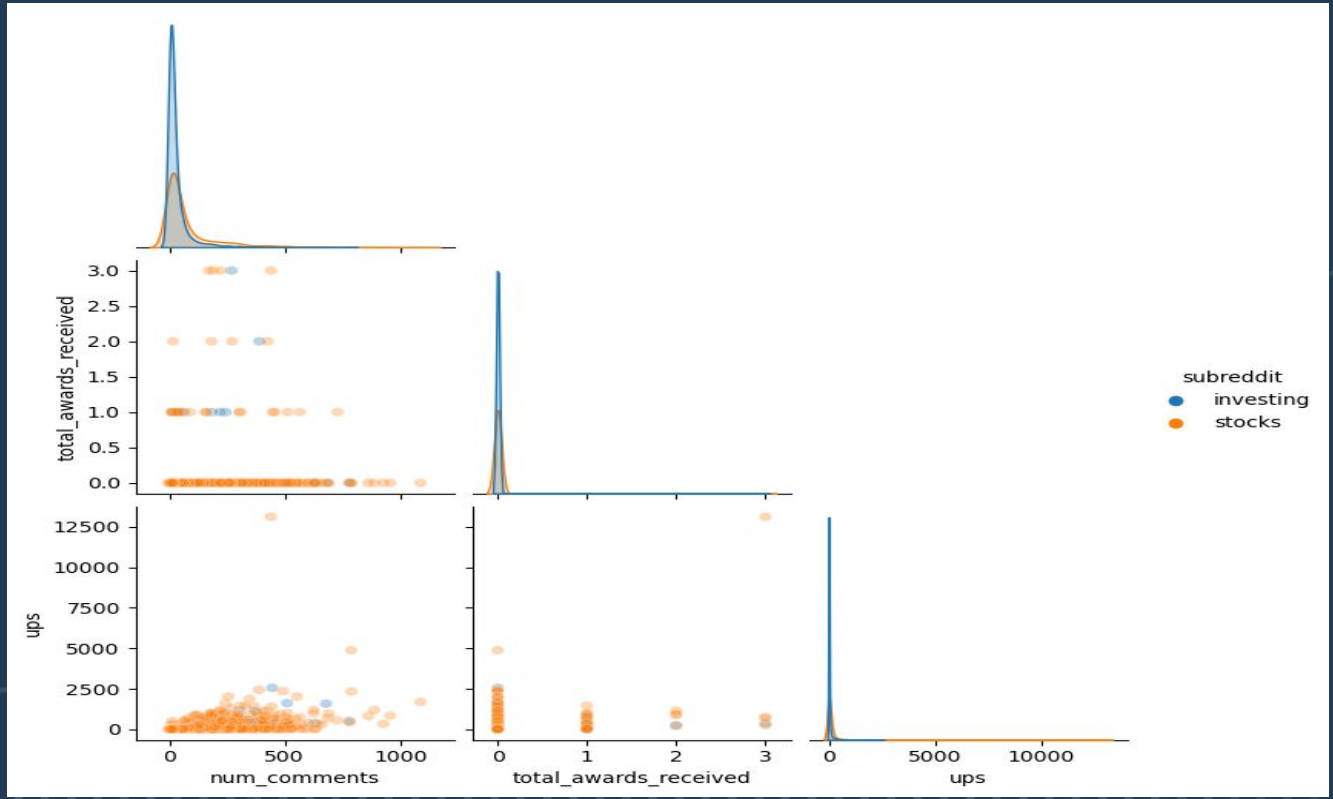
Submission Complexity

- r_stocks has a greater diversity of sentence complexity, word complexity, and paragraph structure.



Submission Engagement

- r_investing has more engagement than r_stocks, as the heightened curtosis of the r_investing distributions shows.



Primary Conclusions

- Stacked models outperformed all their component base models in the validation accuracy and cross validation mean metrics.
- Term frequency was the most predictive of the feature groupings
- The logistic regression models trained on submission text performed almost identically to the estimator trained exclusively on the submission text.
- Validation accuracy of AdaBoost model which used only metrics representing the complexity of the language of the features was 72.9%, showing the complexity and length of the words, sentences and document itself are indicative of which subreddit the document belongs to.
- Validation accuracy of the Knn, RF, and Decision Tree models show engagement on the submission is indicative of which subreddit the post belongs to.

Next Steps

1) Feature engineer more specific grammatical features like:
    a) Coordinating conjunctions ("for, "and", "nor", "but", "or" "yet"), followed by a comma allows you to join two independent clauses together, creating compound sentences.
    b) Commas not associated with a coordinating conjunctions to count the number of complex sentences
    c) The number of sentences with high word counts and a coordinating conjunction but no comma in order to try to gauge how many submissions might be forgetting to use commas.
    d) The number of sentences with very high comma counts to gauge how many submissions might be misusing commas.
    e) The submissions with forgotten commas with the submissions with misused commas to see if user grammar (or incorrect grammar) is itself a feature.

2) Apply Random Forest to the length and ratio features. Random Forests are good with multicollinearity and unscaled data. Also good at incorporating uneven predictive power of different features in the feature set.