

Classifying r/History and r/AlternativeHistory posts

Leonardo Blas

Introduction

- I'm interested in classifying social media posts as “legit” or “fake”.
- I decided to scrape and classify r/History and r/AlternativeHistory posts.



Challenges and opportunities

- Different subreddits may have different submission schemes.
- Even if two subreddits' submissions have the same feature, that column's values can vary a lot.

```
print(history["removal_reason"].unique())  
print(alternative_history["removal_reason"].unique())
```

```
[nan]
```

```
[nan 'legal']
```

One approach

- Homogenize the features for both types of posts.
- Use as many reasonable features as possible to make good predictions.
- Mix both categories of posts into a dataset and chuck it into a model.

Decisions

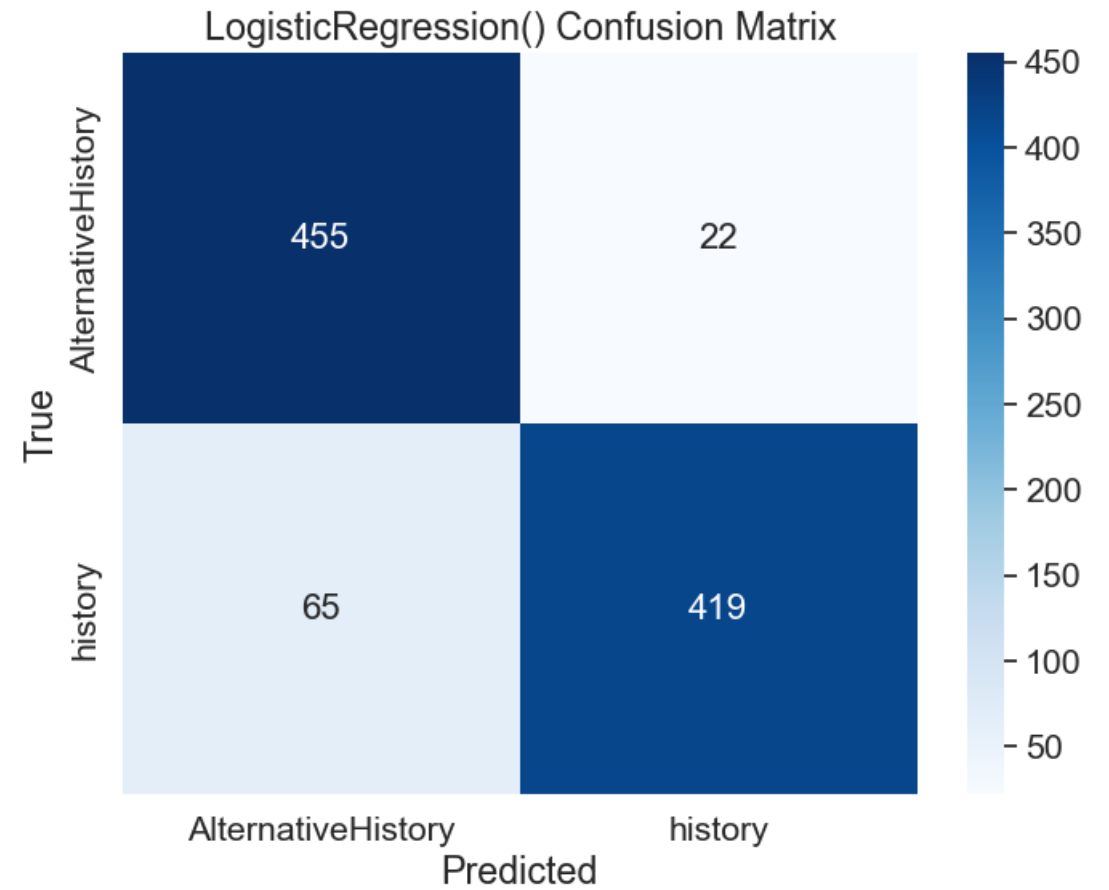
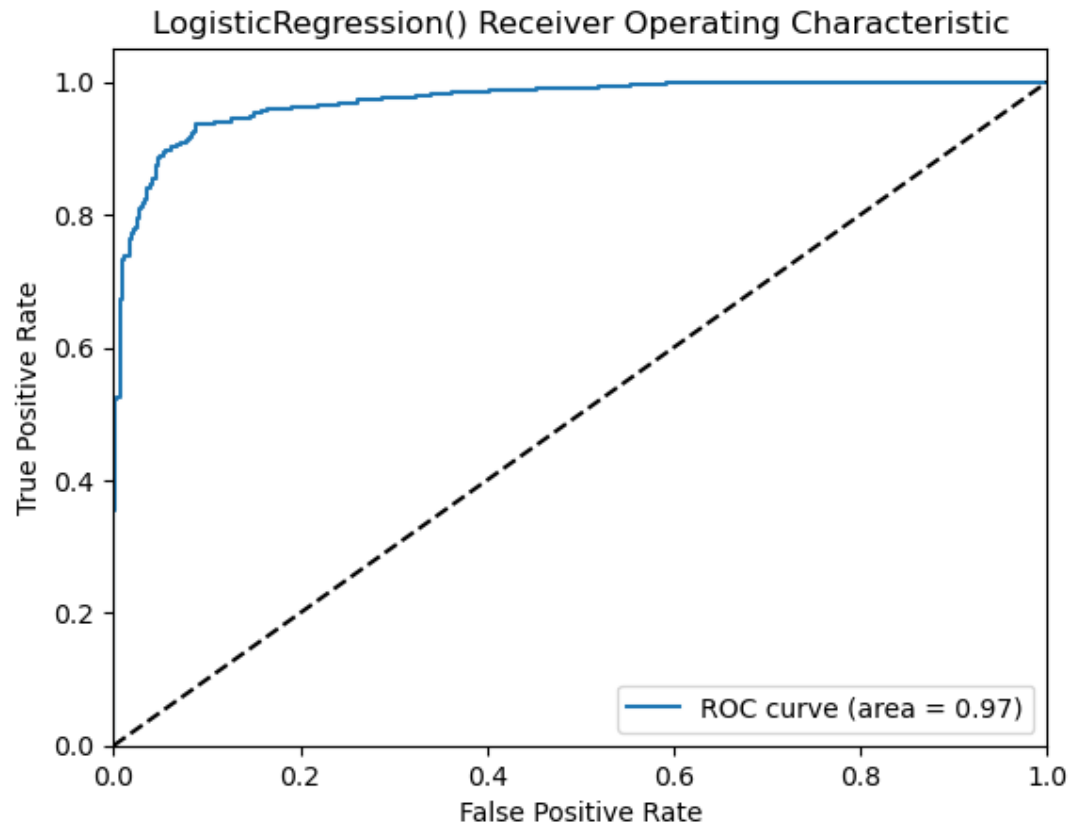
Chosen features

```
columns_to_keep = [  
    "subreddit",  
    "selftext",  
    "gilded",  
    "title",  
    "thumbnail_height",  
    "upvote_ratio",  
    "score",  
    "edited",  
    "is_self",  
    "created",  
    "domain",  
    "allow_live_comments",  
    "no_follow",  
    "locked",  
    "author",  
    "num_comments",  
    "send_replies",  
    "num_crossposts"  
]
```

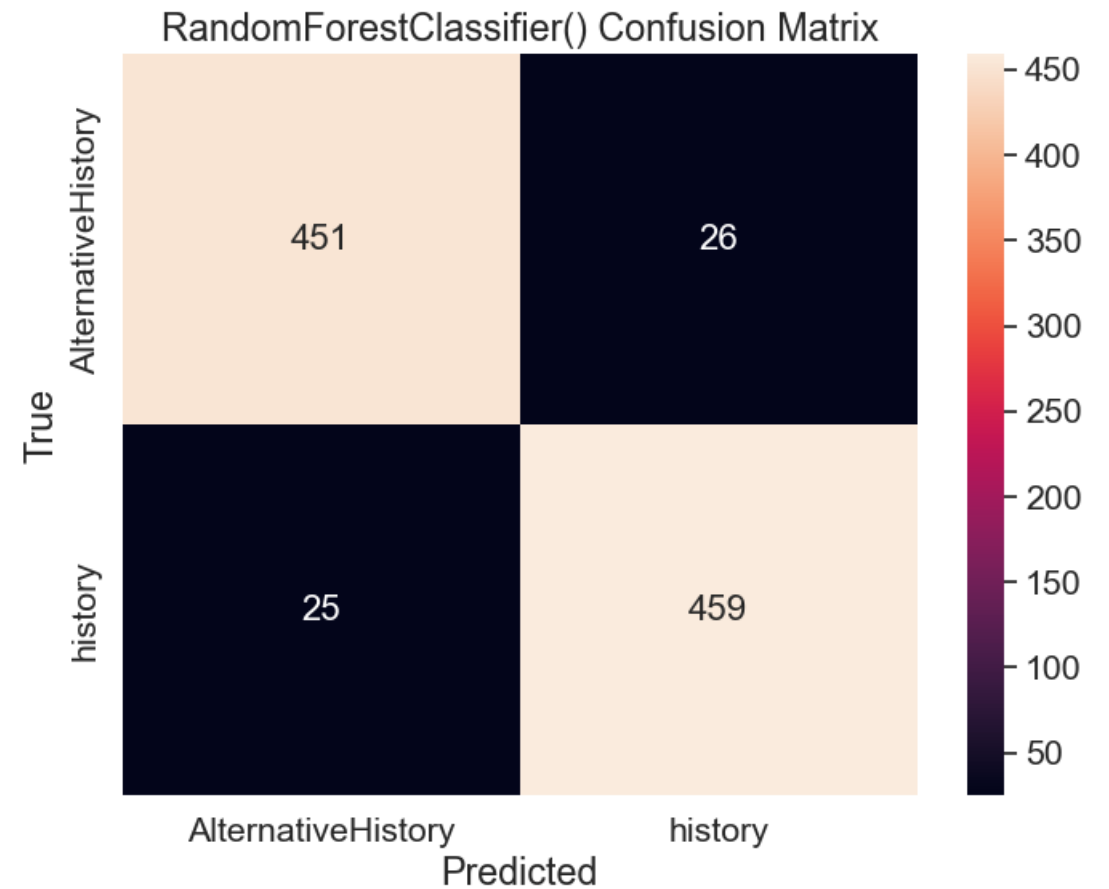
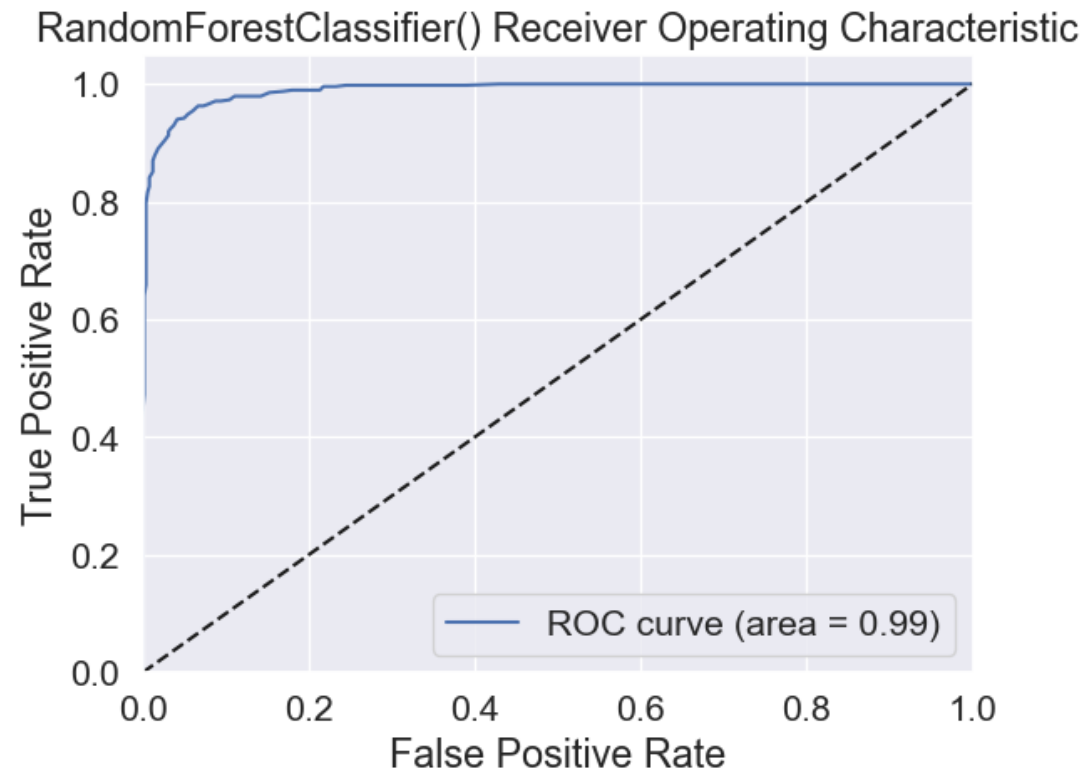
Chosen tools

- LogisticRegression().
- RandomForestClassifier().
- Both vanilla because gridsearch was taking a long time.
- TfidfVectorizer, because we want each token to have a weight.

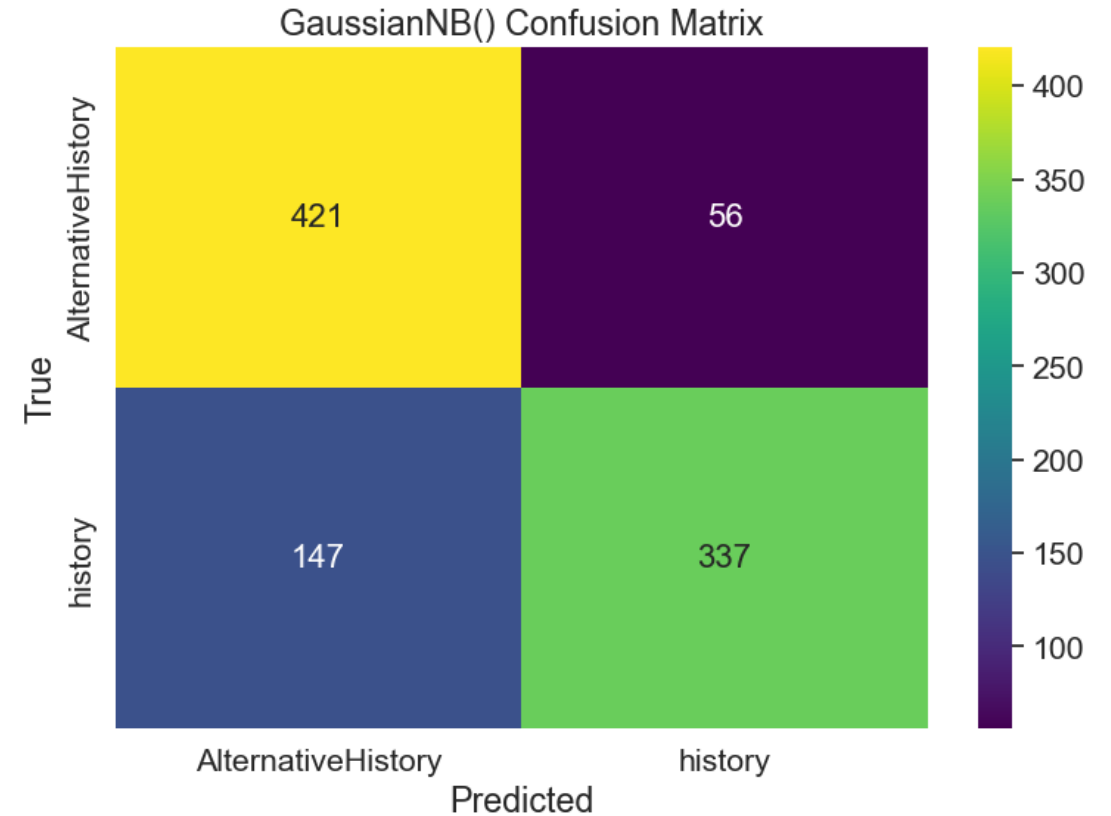
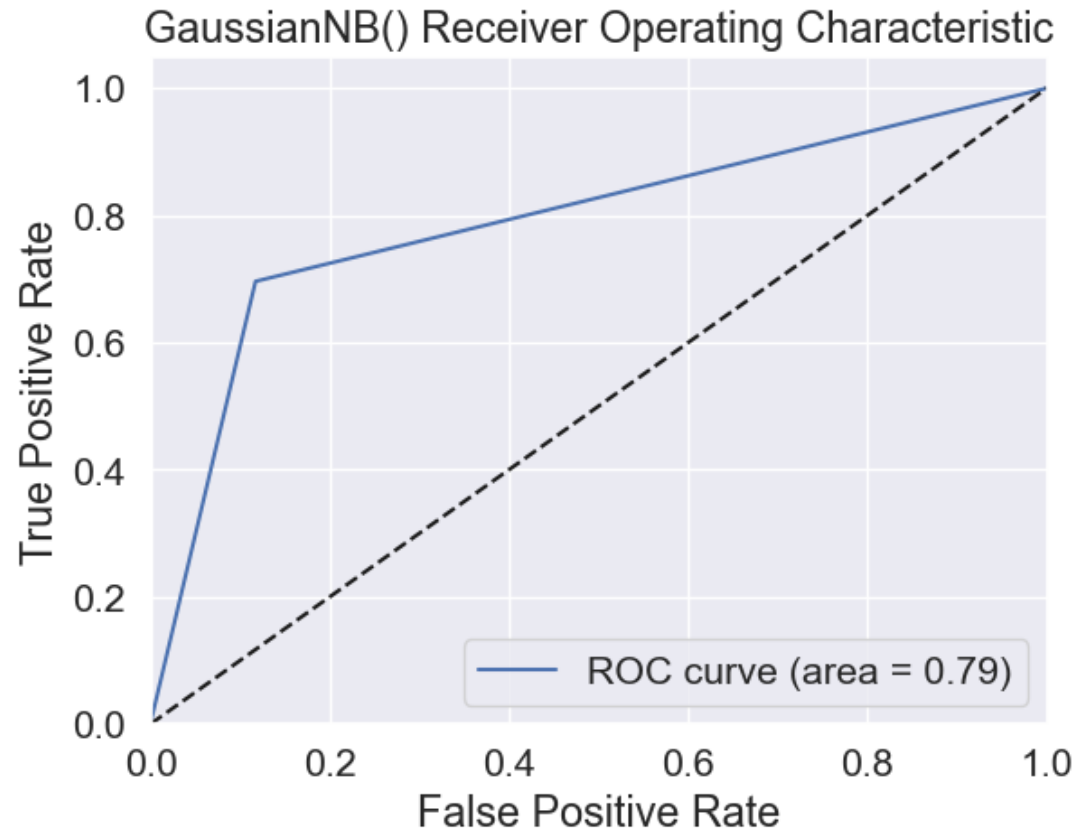
Logistic model



Random forest classifier model

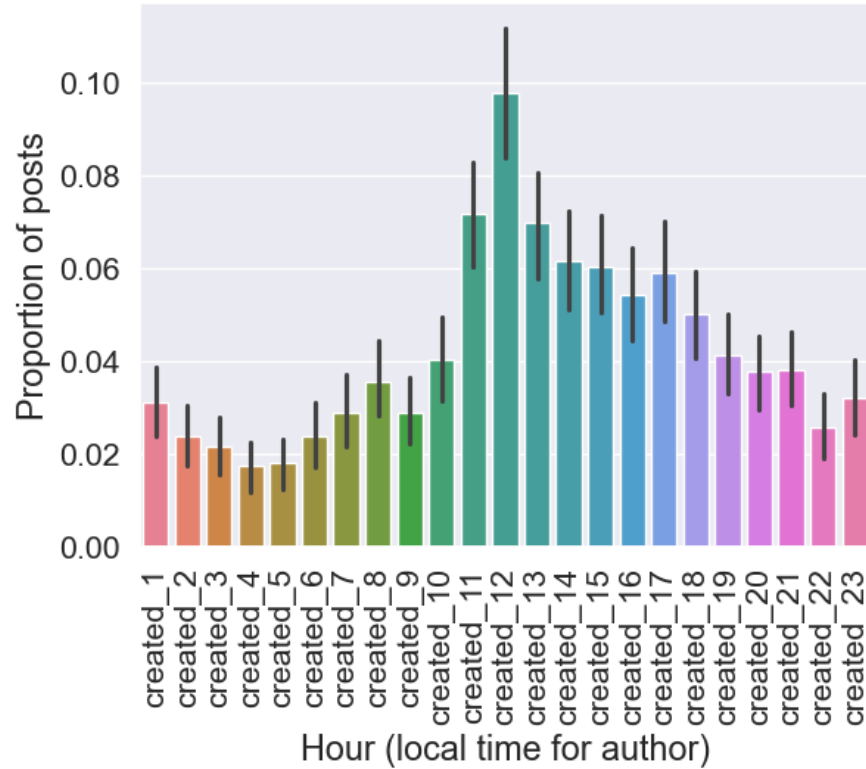


Naïve bayes model

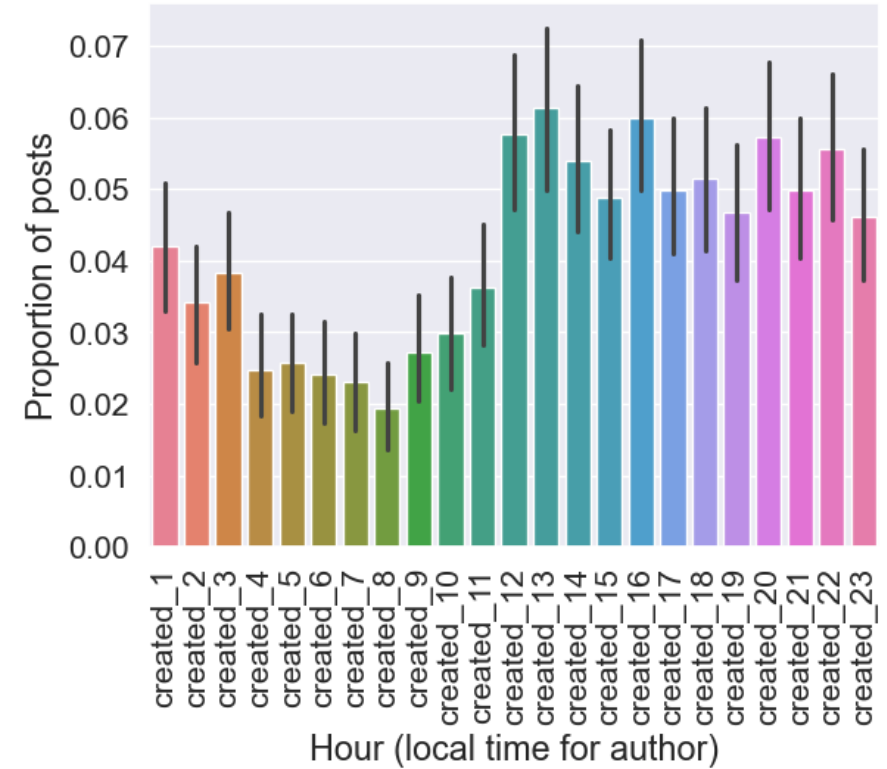


Interesting: Time of posting

r/History posts time of creation (one-hot encoded and dropped time 0)

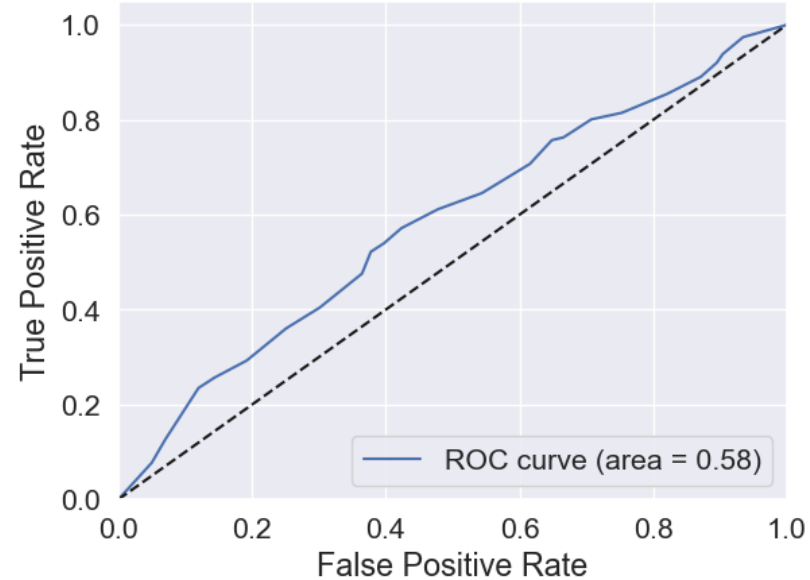


r/AlternativeHistory posts time of creation (one-hot encoded and dropped time 0)

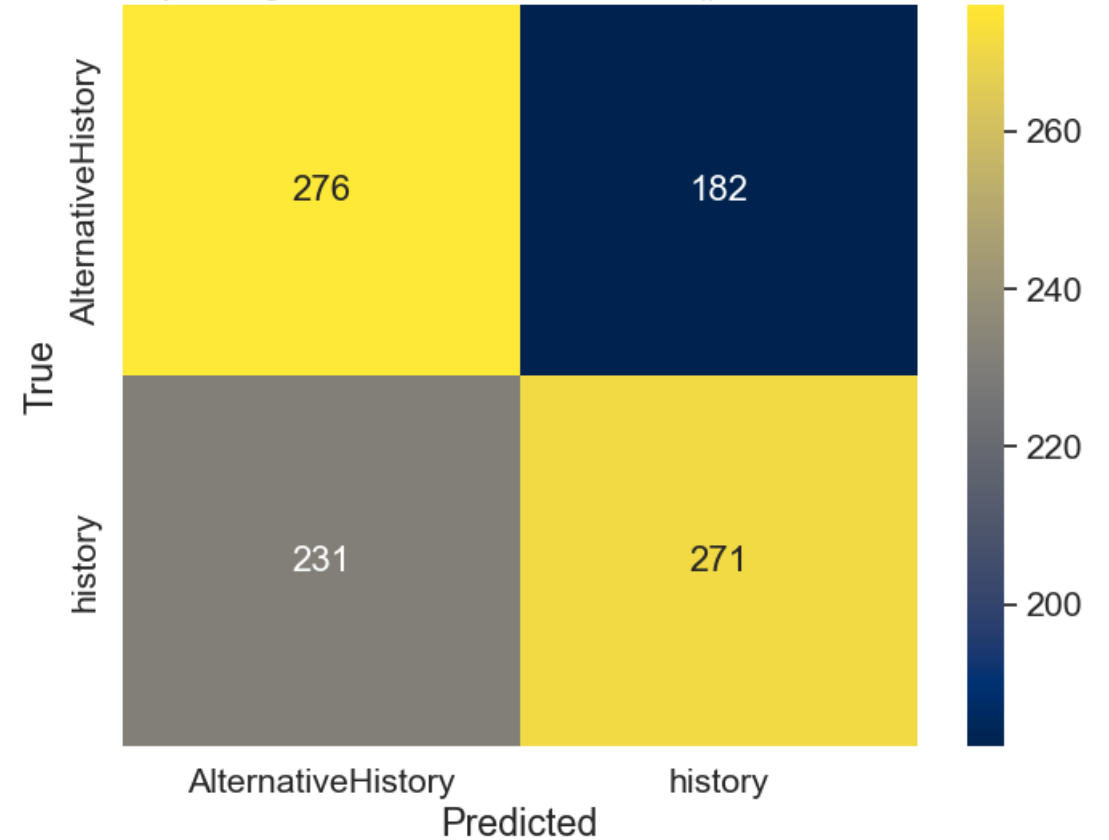


Interesting: Time of posting as predictor

Time of posting RandomForestClassifier() Receiver Operating Characteristic



Time of posting RandomForestClassifier() Confusion Matrix



Conclusion

- Vanilla classifiers are surprisingly good at classifying posts from closely related topics.
 - Logistic regressor: ~ 0.95 train, ~ 0.90 test.
 - Random forest classifier: 1.0 train, ~ 0.95 test.
- Posts are more than text. Other features, like the number of crossposts (how many times a post was featured in other unique subreddits) proved to be helpful.
- Interesting: The time of posting achieved ~ 0.56 accuracy in a random forest classifier model.