Classifying Reddit user vs AI-generated responses

Alan Andrews DSIR-814 Project 3

Overview

1) Collect question answer pairs from Reddit.
2) Submit questions to OpenAI model.
3) Build models to predict human-generated text vs AI-generated text.

Data collection

Python Reddit API Wrapper - Praw

Subreddits: Ask, AskReddit, AskScience, AskHistorians, Ask_politics, AskCulinary

Additional subreddits: AskUK, AskStatistics, AskScitech

Created loop to collect attributes from 999 top comments in each subreddit.

Collected 5250 question-answer pairs.

Data collection

OpenAI API

Used backoff module to stay within API rate limit.

Submitted questions collected from Reddit in batches of 20.

Used the OpenAI Davinci Model.

$$$$

Data Cleaning

Removed [deleted], [removed] posts.

Removed '\n'

Removed '\\_'

Labeled answers as either AI generated (1) or not (0)

Models

10 Grid Searches

Countvectorizer

Multinomial Naive Bayes

Logistic Regression

Bernoulli Naive Bayes

Linear Support Vector Classification

TFID

Multinomial Naive Bayes

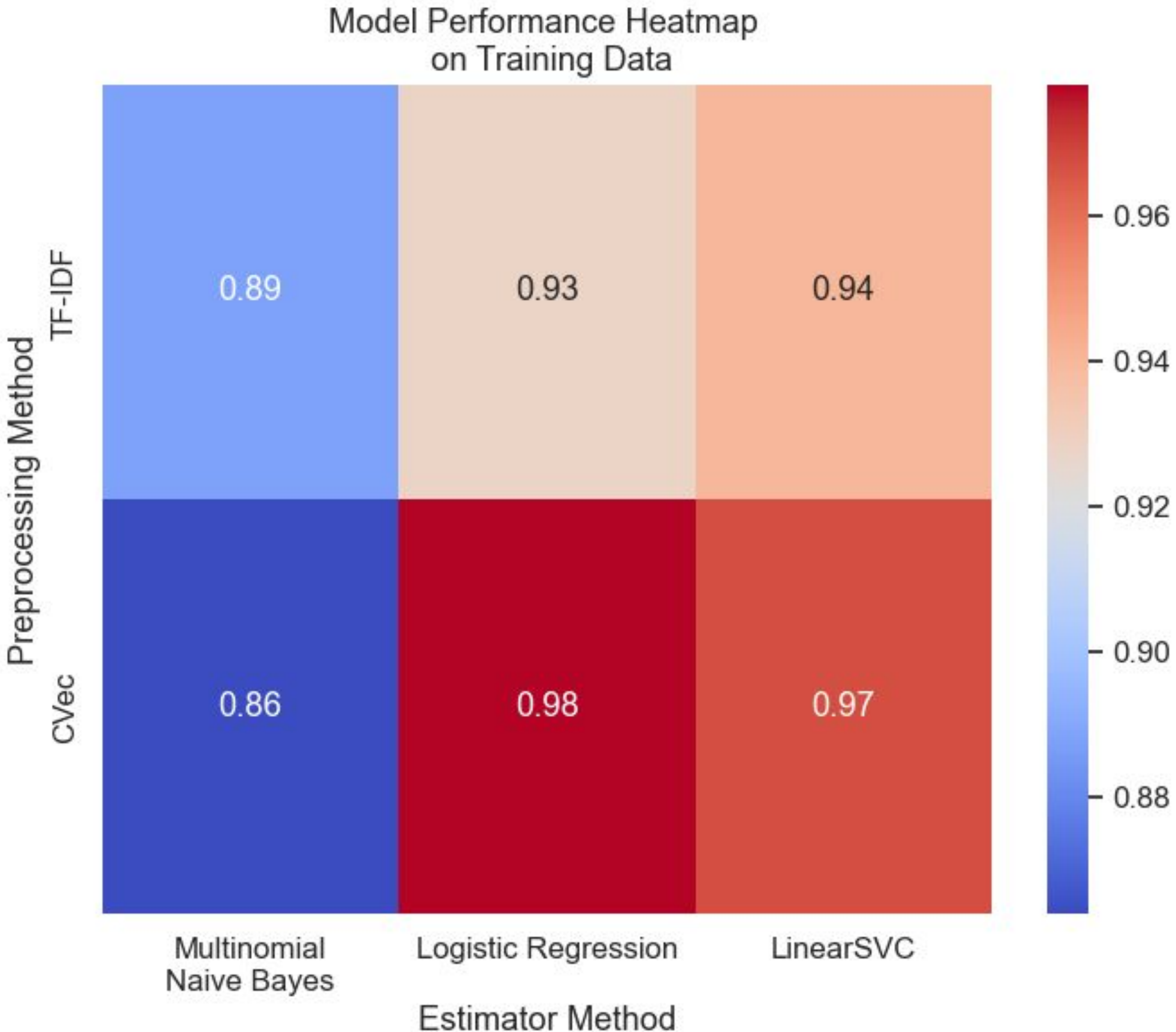
Logistic Regression

Gaussian Naive Bayes

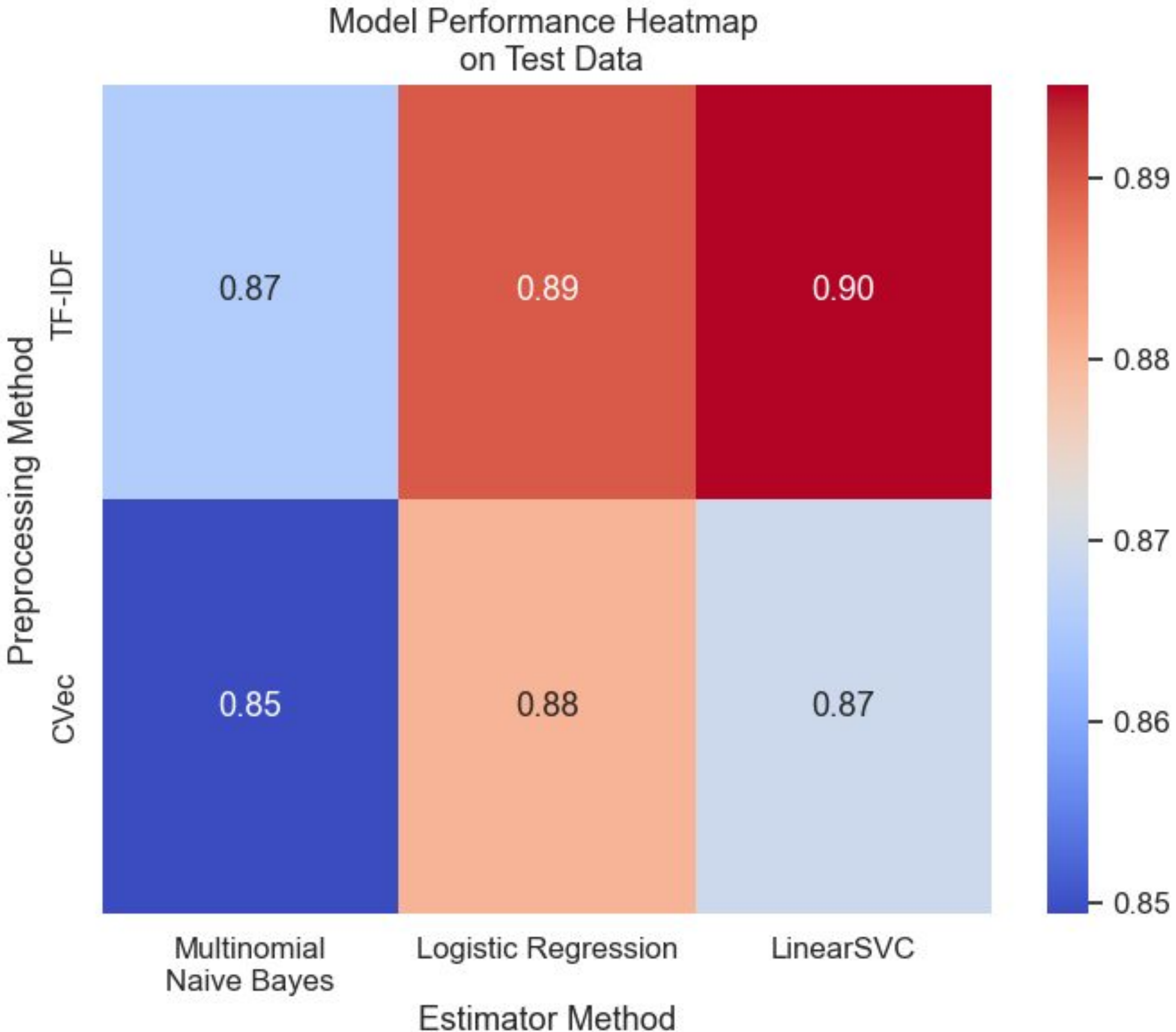
K-nearest neighbors

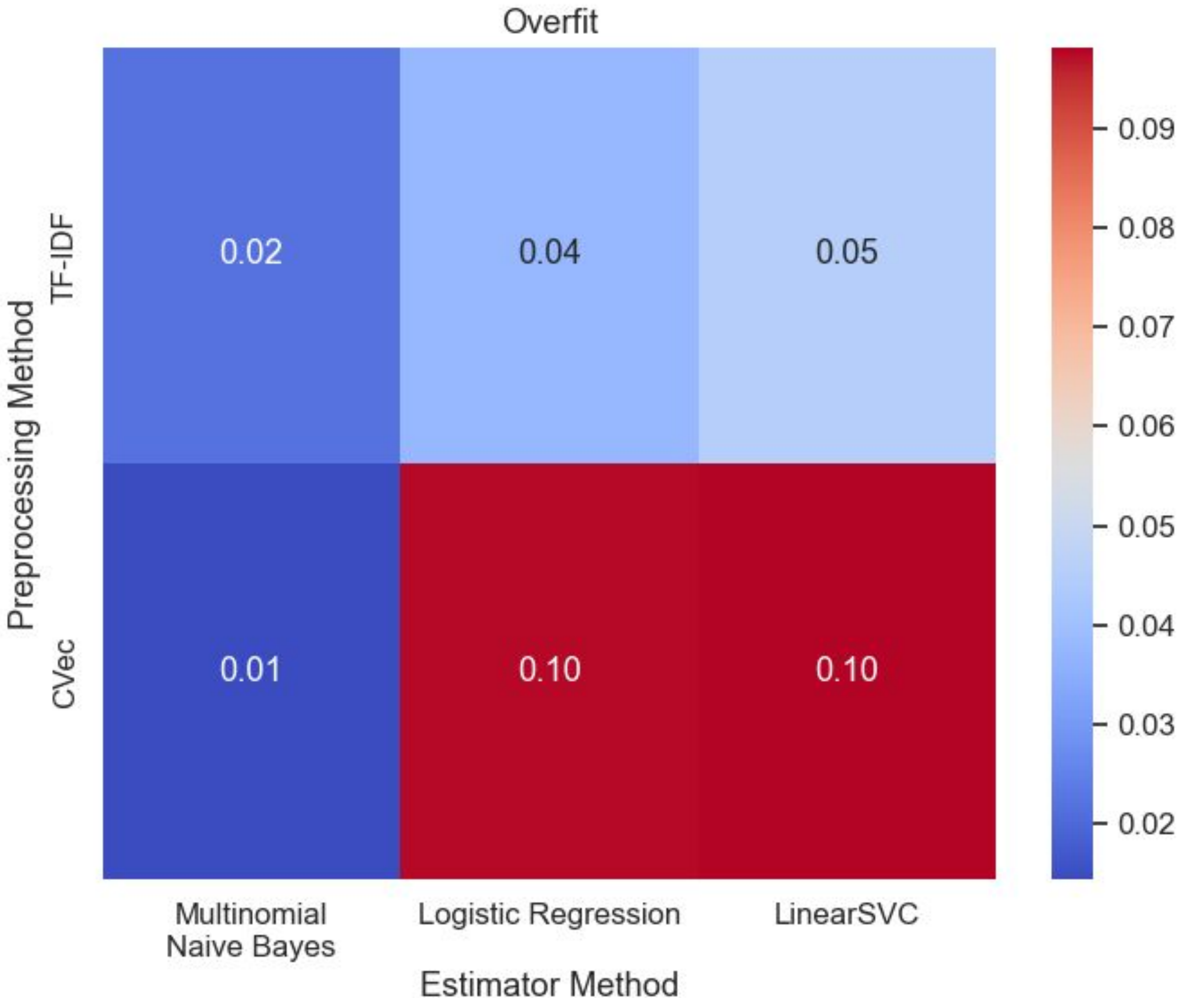
Random Forest

Linear Support Vector Classification



Model Performance Heatmap
on Training Data

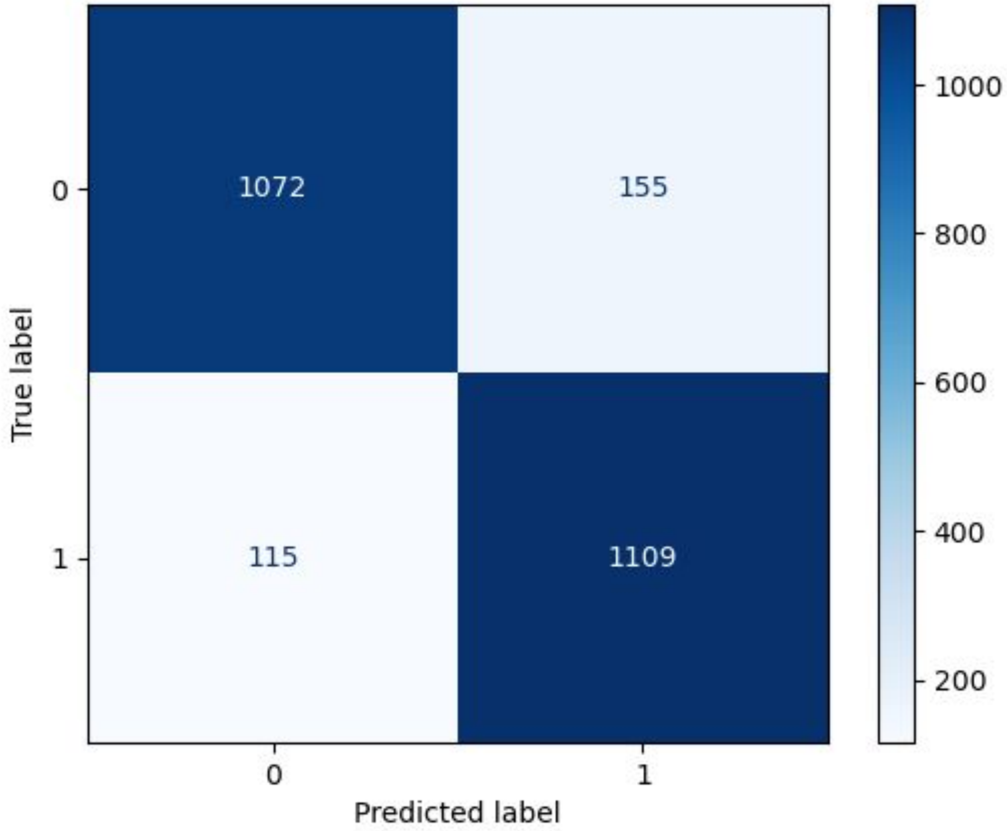

Model Performance Heatmap
on Test Data
Preprocessing Method
TF-IDF
CVec
0.87
0.89
0.90
0.85
0.88
0.87
Multinomial
Naive Bayes
Logistic Regression
LinearSVC
Estimator Method
0.89
0.88
0.87
0.86
0.85



Overfit

Preprocessing Method
TF-IDF
CVec
0.02
0.04
0.05
0.01
0.10
0.10
Multinomial Naive Bayes
Logistic Regression
LinearSVC
Estimator Method
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02

TVEC LOGR



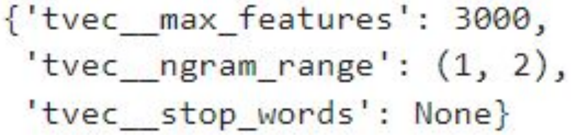
Accuracy: 0.8898

Precision: 0.8773

Recall: 0.9060

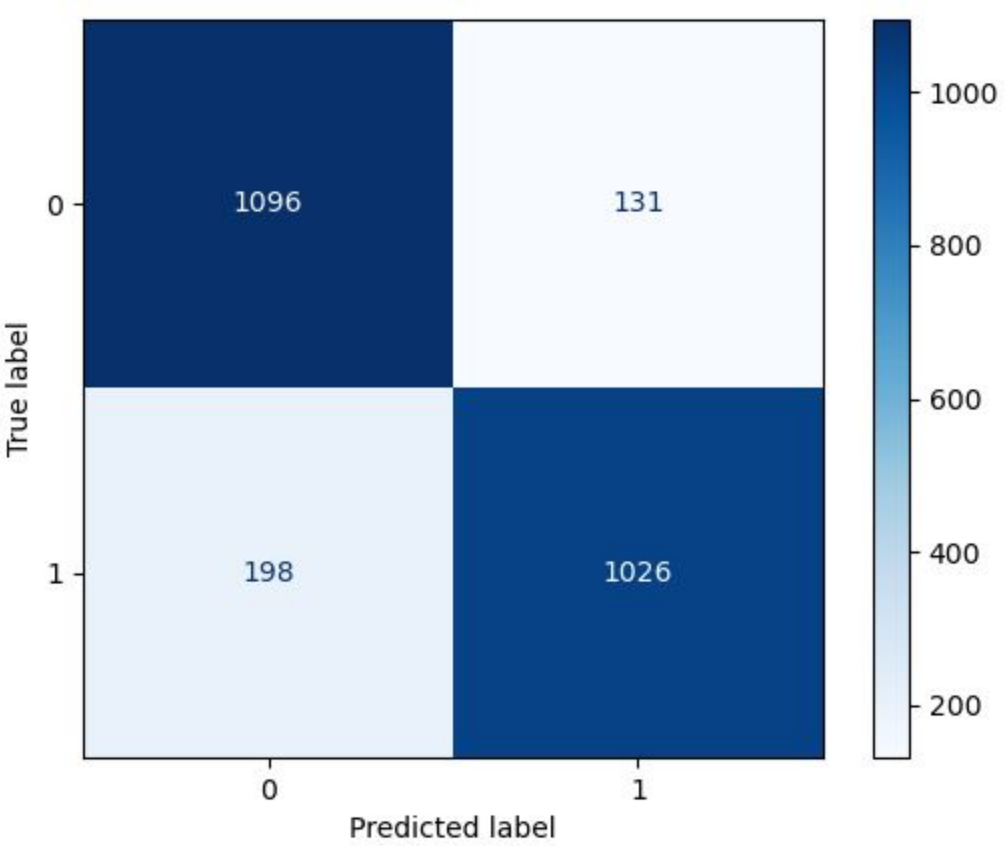
Specificity: 0.8736

F1 Score: 0.8914

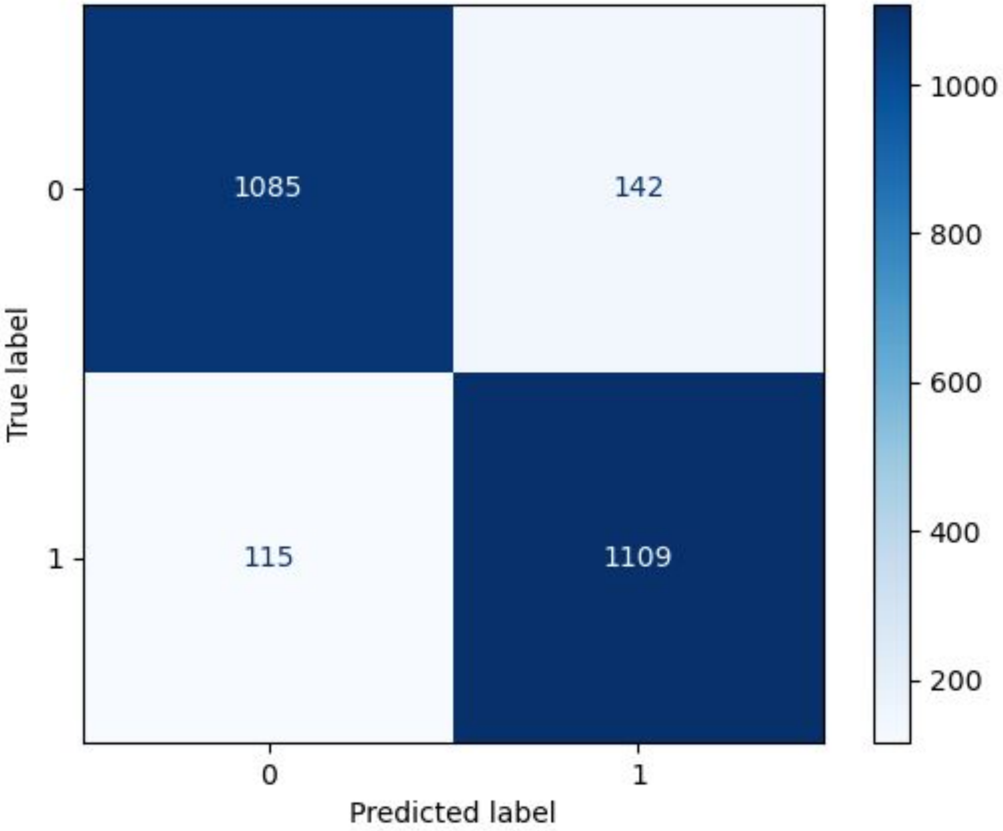


```
{'tvec__max_features': 3000,
 'tvec__ngram_range': (1, 2),
 'tvec__stop_words': None}
```

TVEC MNB


1096
131
198
1026
0
1
True label
0
1
Predicted label
1000
800
600
400
200

TVEC LSVC



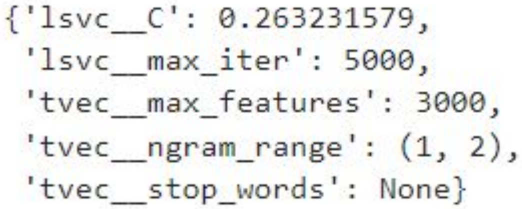
Accuracy: 0.8694

Precision: 0.8779

Recall: 0.8578

Specificity: 0.8810

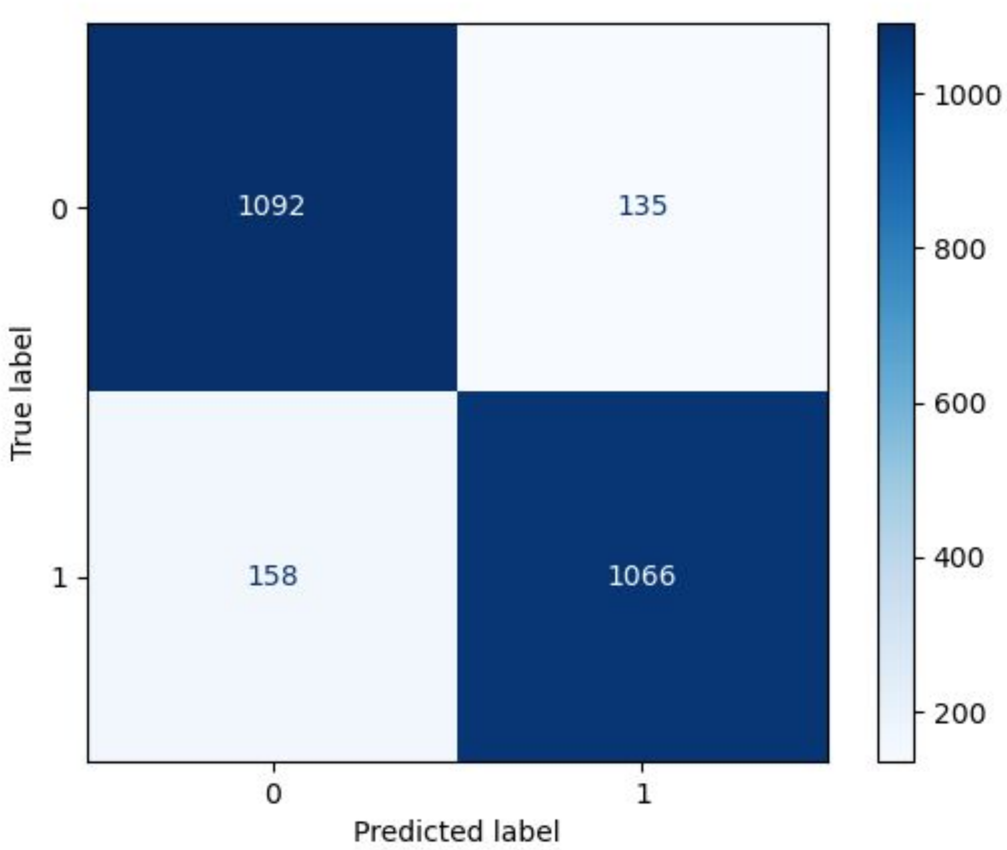
F1 Score: 0.8810



```
{'lsvc__C': 0.263231579,
 'lsvc__max_iter': 5000,
 'tvec__max_features': 3000,
 'tvec__ngram_range': (1, 2),
 'tvec__stop_words': None}
```

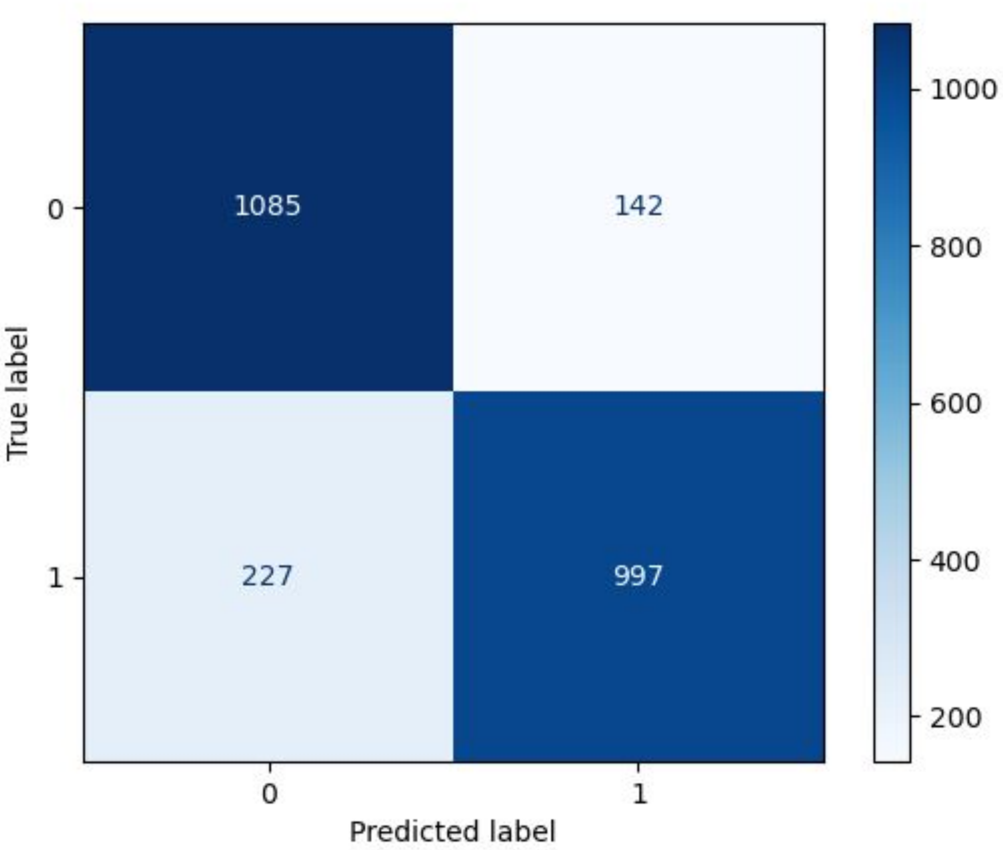
CVEC LOGR

E


1092
135
158
1066
True label
0
1
Predicted label
0
1
1000
800
600
400
200

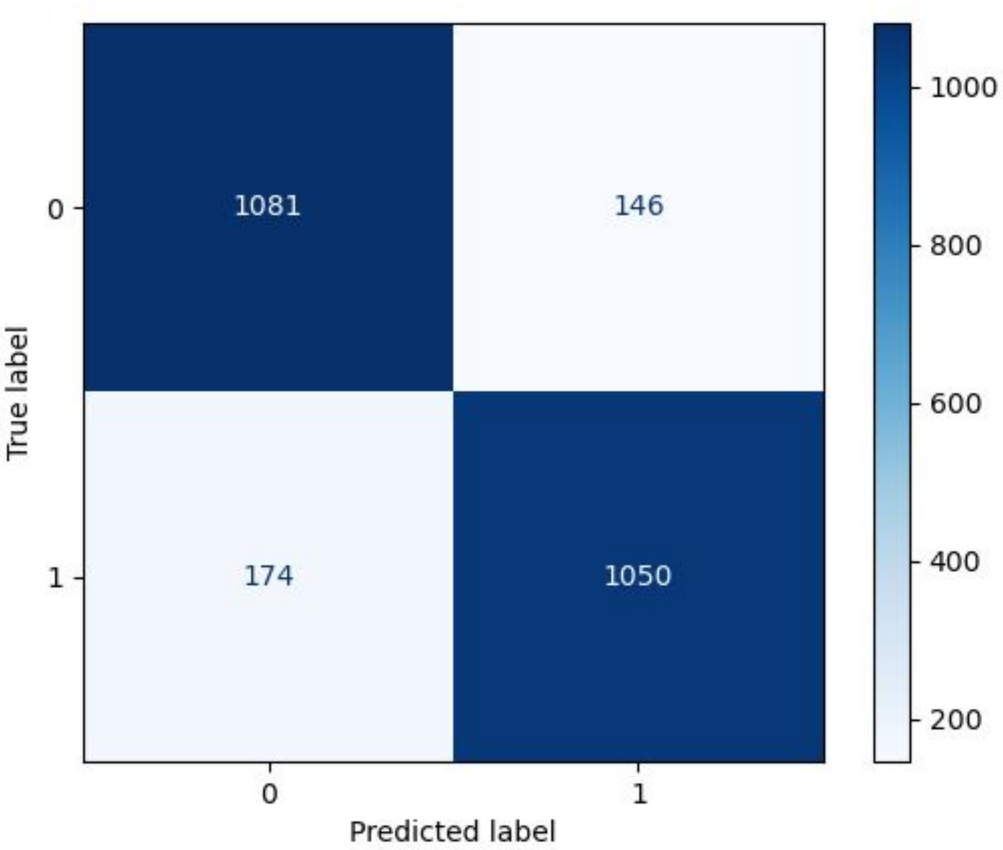
CVEC MNB

E


1085
142
227
997
0
1
True label
0
1
Predicted label
1000
800
600
400
200

CVEC LSVC

E


1081
146
174
1050
True label
0
1
Predicted label
0
1
1000
800
600
400
200

Future steps

Continue fitting models, such as XGBoost.

Continue adding data.

Continue data cleaning.

Conclusions

The best performing models were the pipelines with Logistic Regression and LSVC estimators using TFID vectorizer preprocessing.

They performed well and generalized on unseen data.