

Classifying Reddit user vs AI-generated responses

Alan Andrews DSIR-814 Project 3





Data collection

Python Reddit API Wrapper - Praw

Subreddits: 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

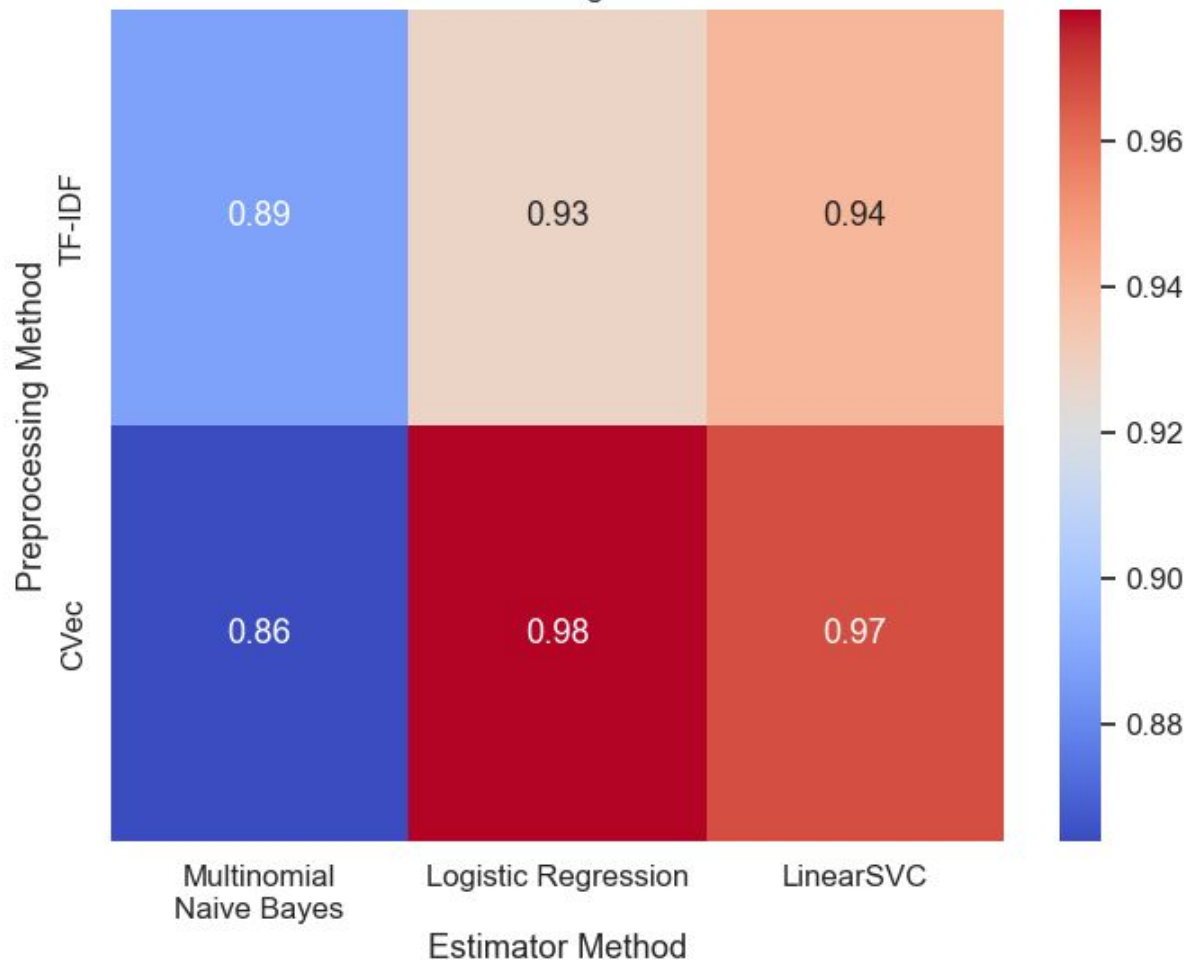
Logistic Regression

Gaussian Naive Bayes

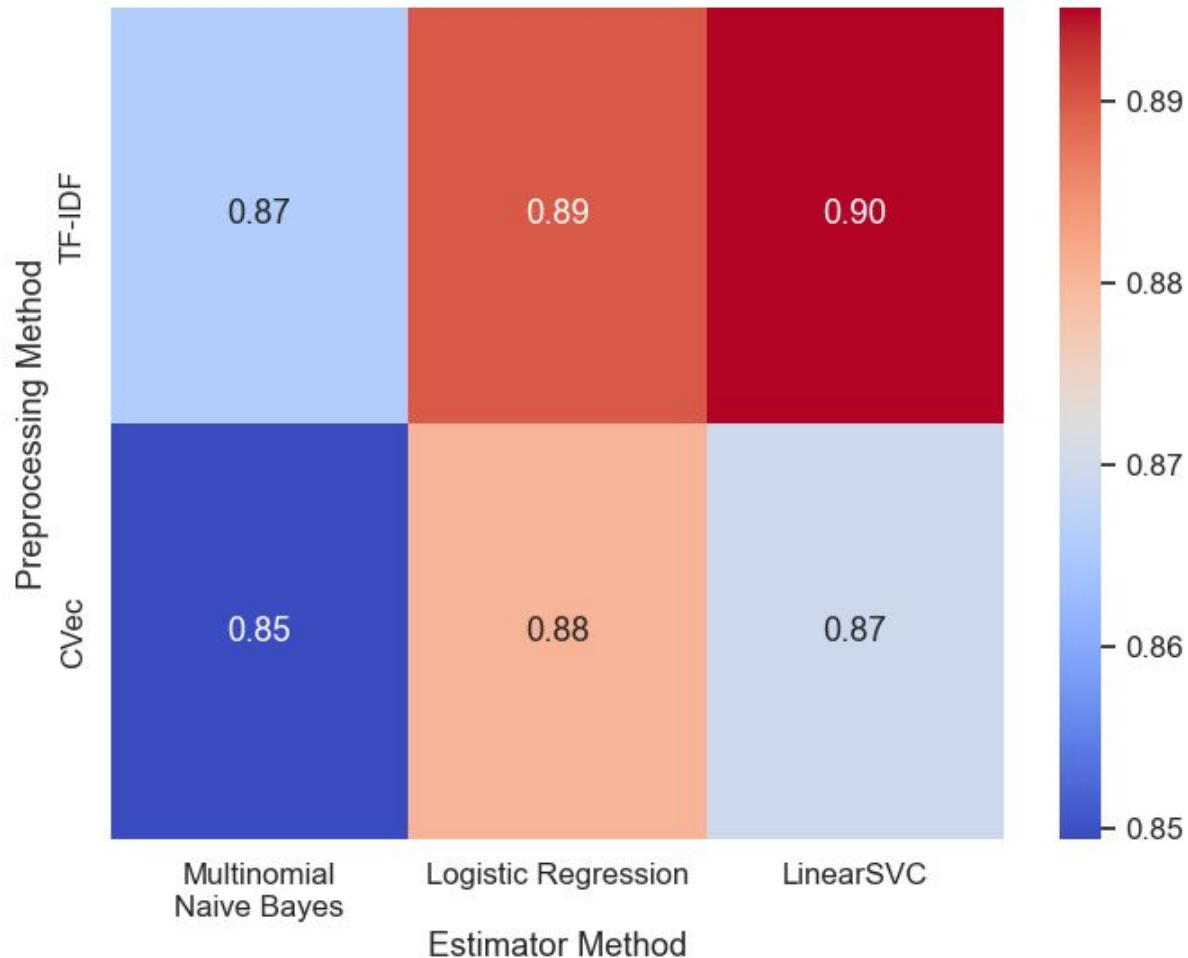
K-nearest neighbors

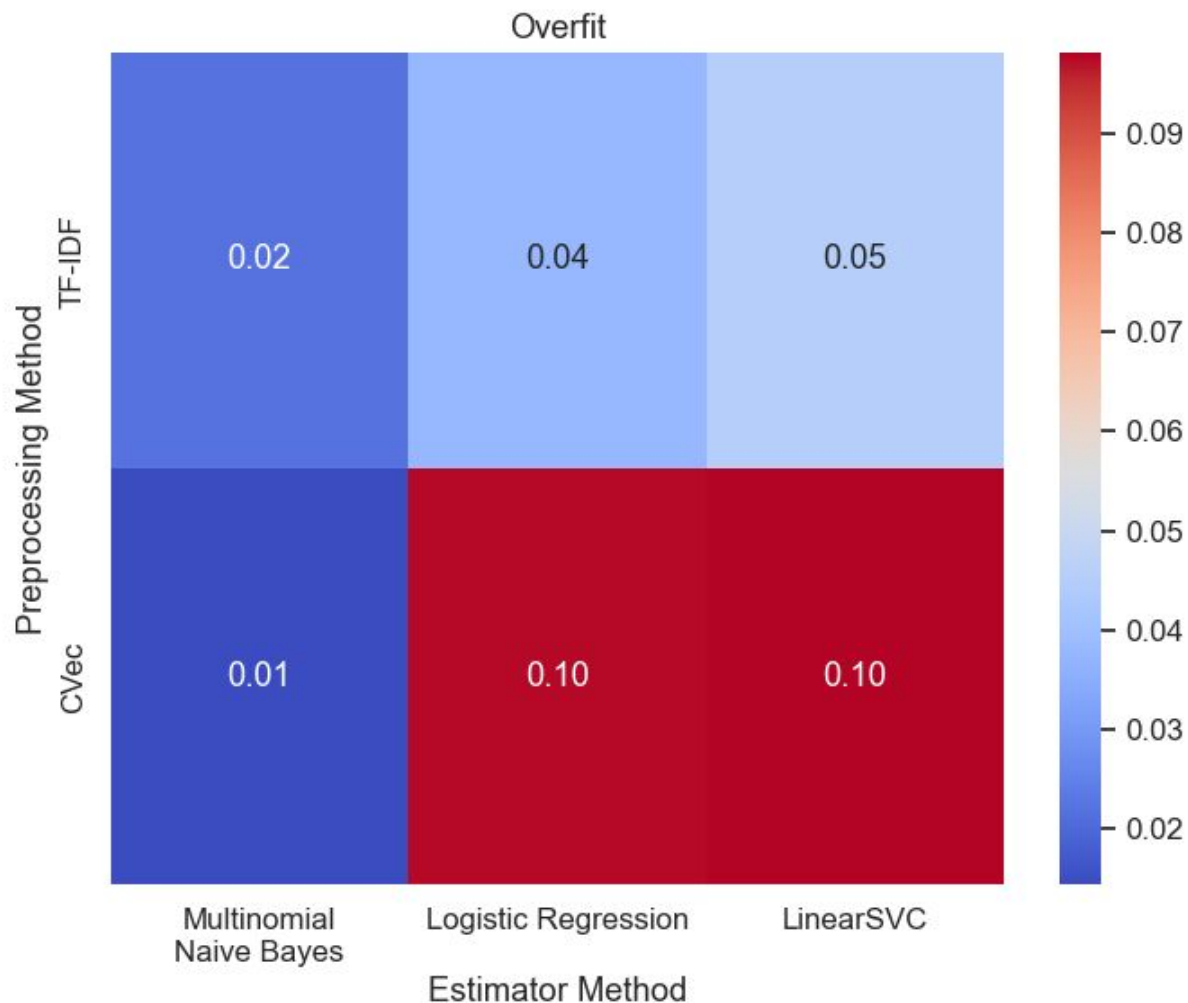
Random Forest

Model Performance Heatmap
on Training Data

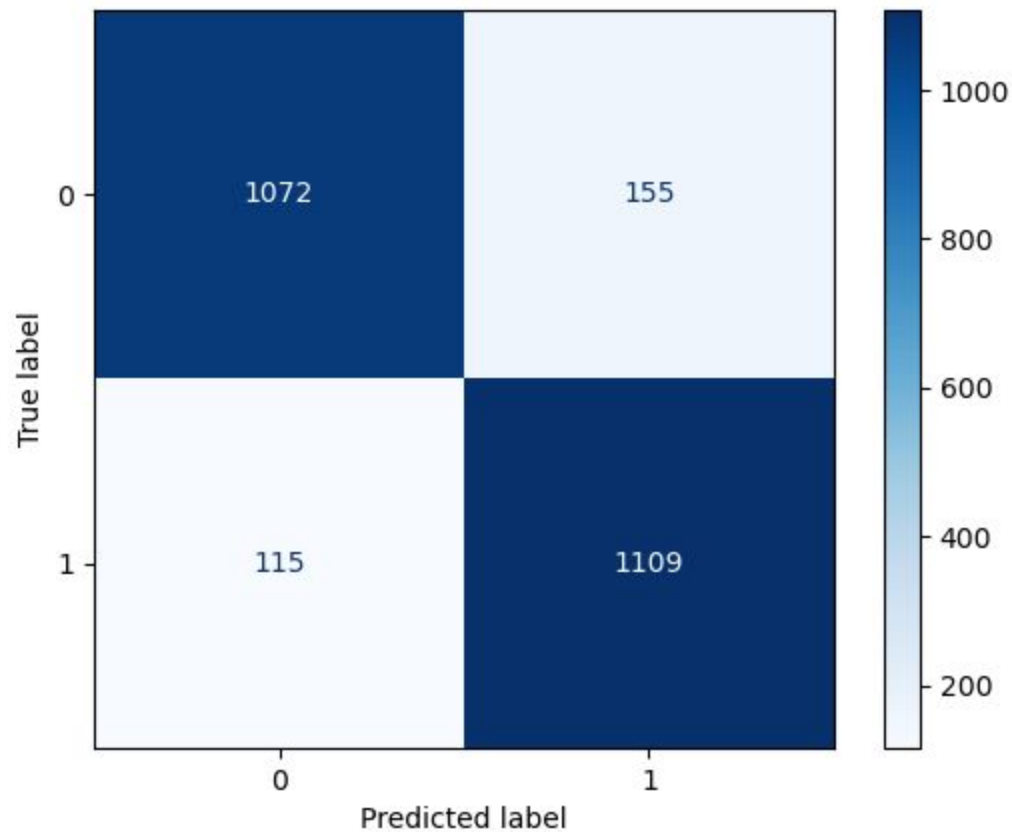


Model Performance Heatmap
on Test Data





TVEC LOGR



Accuracy: 0.8898

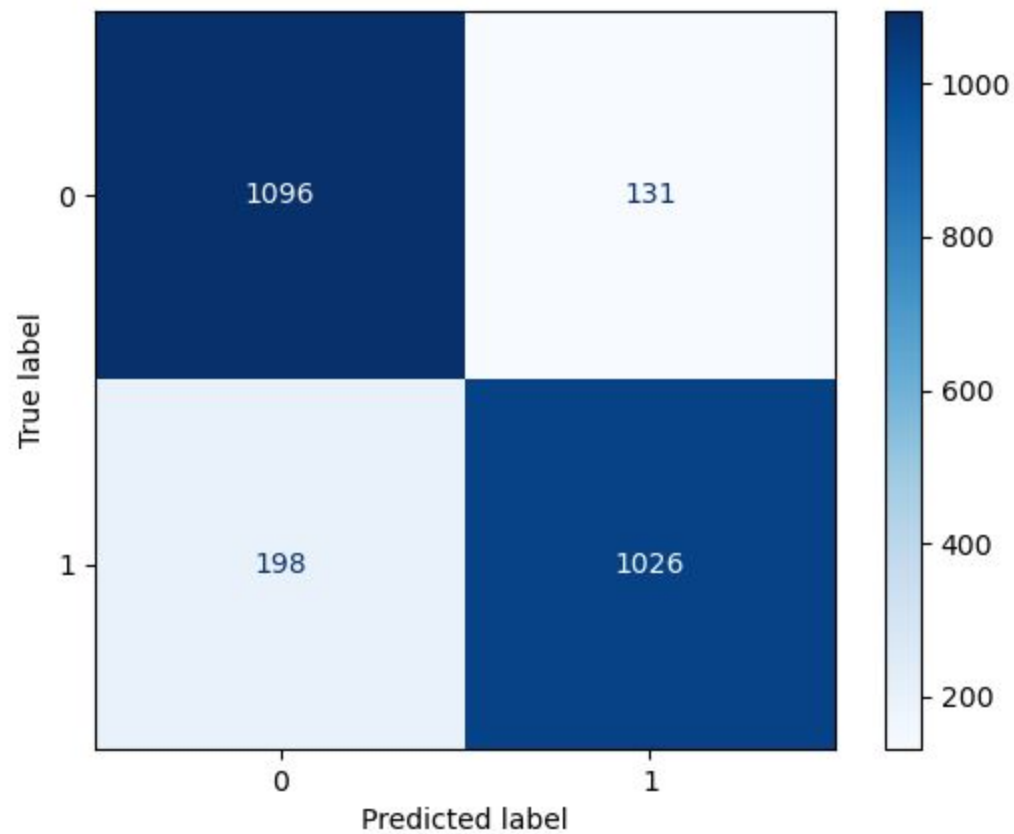
Precision: 0.8773

Recall: 0.9060

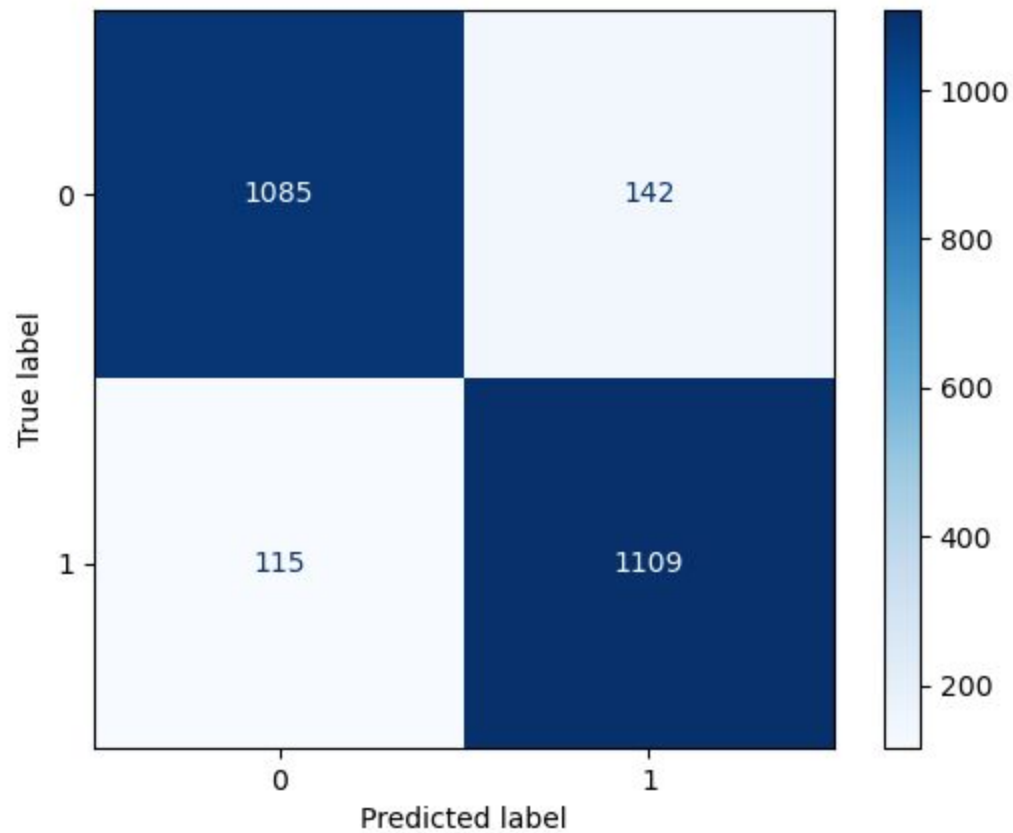
Specificity: 0.8736

F1 Score: 0.8914

TVEC MNB



TVEC LSV



Accuracy: 0.8694

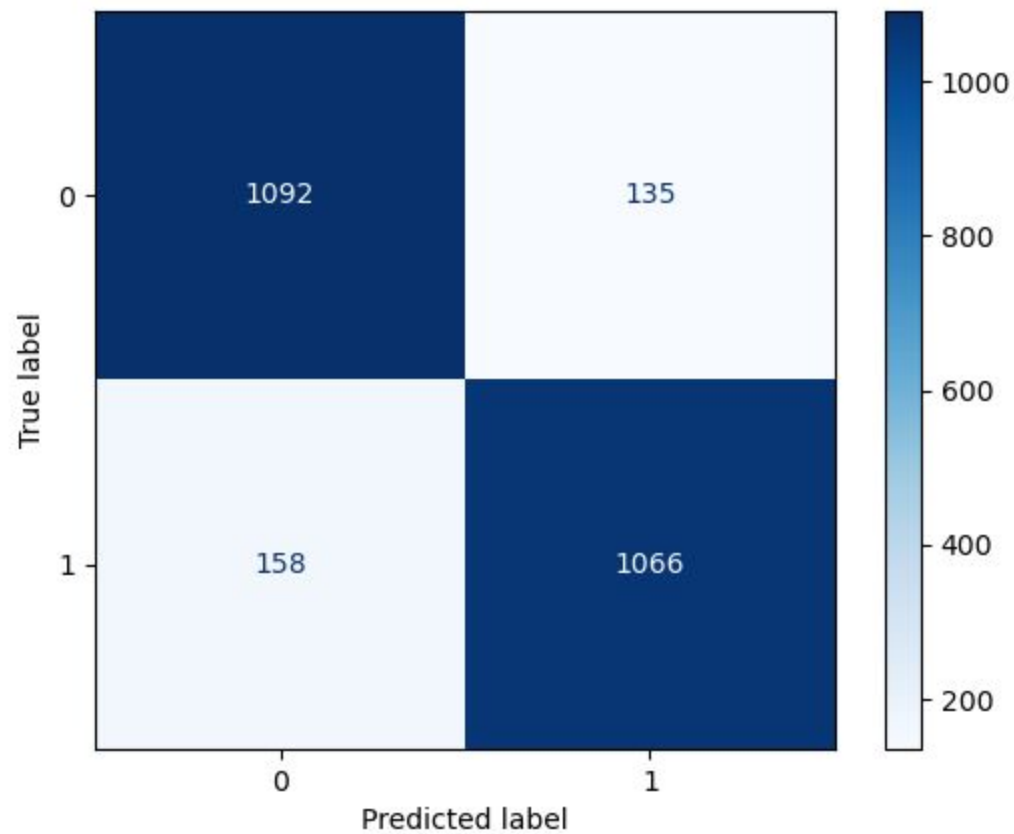
Precision: 0.8779

Recall: 0.8578

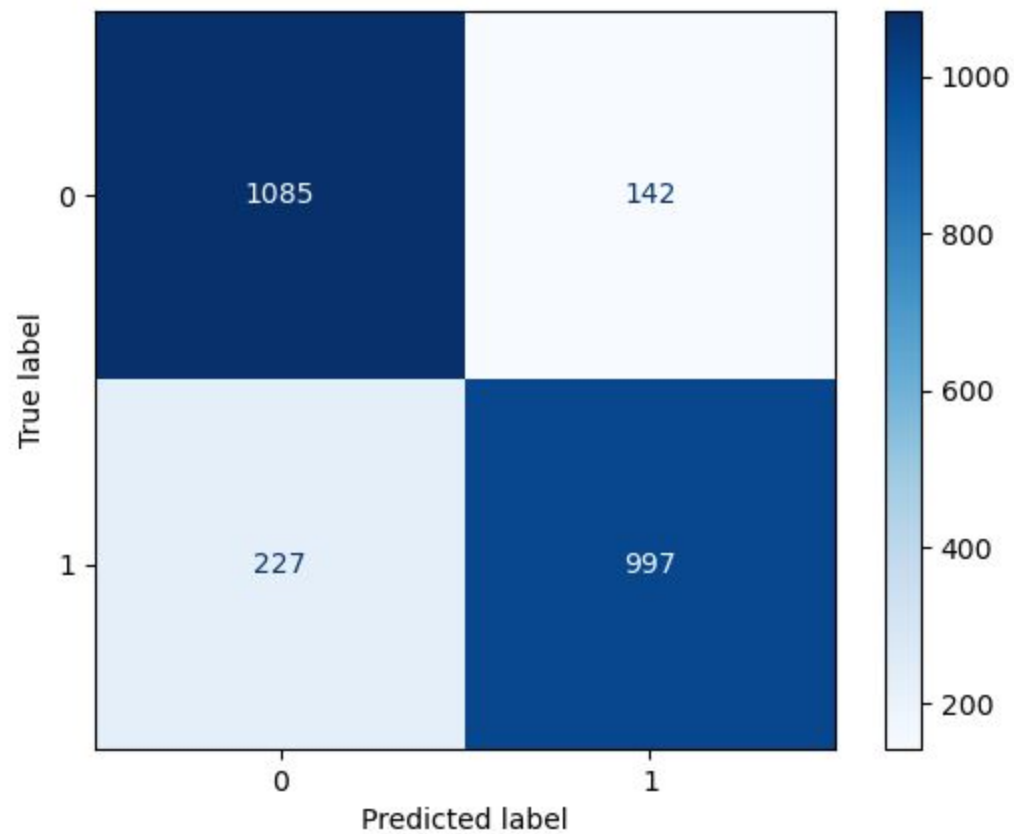
Specificity: 0.8810

F1 Score: 0.8810

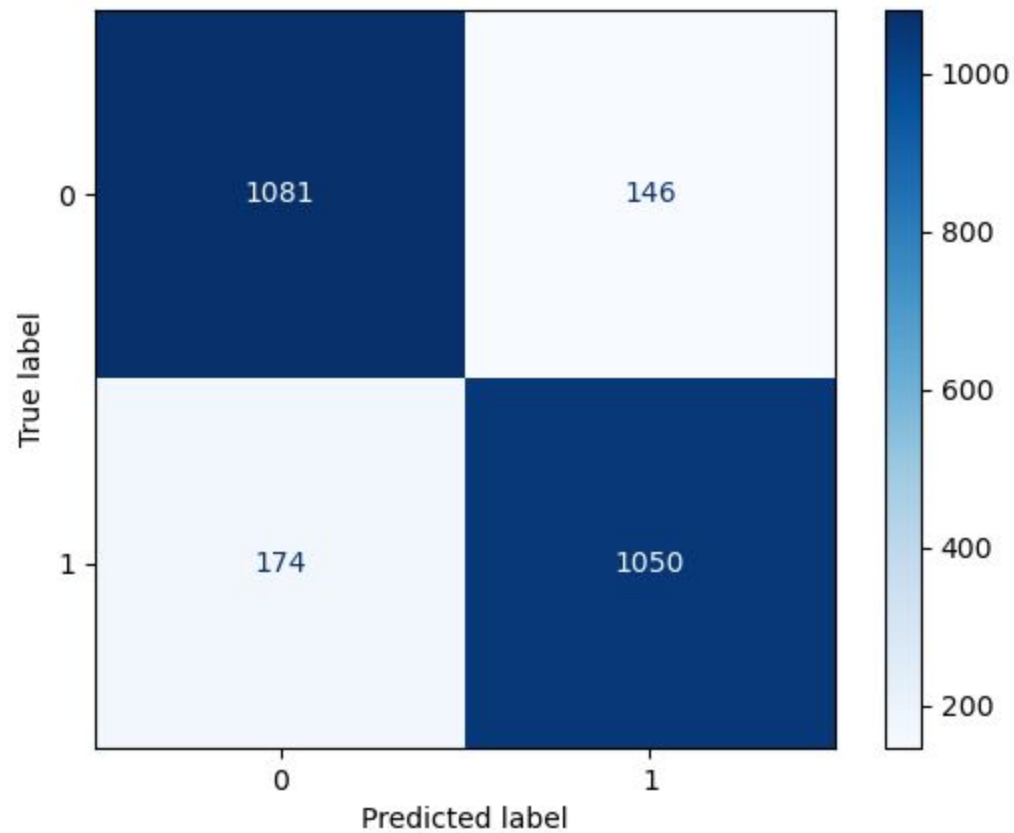
CVEC LOGR



CVEC MNB



CVEC LSV





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.