

David Payare, Justin Pham, Vincent Phan, Keaton Ylanan

Professor Parolin

CS 4375.501

11 May 2025

## Project Report

### **Introduction:**

For our project, we chose to do Option 2, where we worked with text classifiers. We decided to perform sentiment analysis on our datasets, classifying each instance as either “positive” or “negative”, without including a neutral class, for simplicity. We used four diverse datasets, and applied different classifiers to each, including Logistic Regression, Linear SVC, SGD, and Naive Bayes. Hyperparameter tuning was applied to each classifier, to see the results of selecting different parameters. Additionally, we used Transformers (specifically BERT) from Hugging Face in order to implement a deep learning model.

### **Methodology:**

For the methodology, we first selected from a number of diverse datasets containing bodies of text associated with a rating of some kind. Then, for each dataset, we imported and applied preprocessing to them, which included the removal of punctuation and changing of all letters to lowercase, where appropriate. Additionally, stop words were used to remove “fluff” words that had no bearing on the positivity or negativity of a given review. A count vectorizer was used to extract features from the dataset, and then the results were used in a TF-IDF transformer, so that TF-IDF could be used for the datasets, instead of Bag-of-Words. From there, the different classifiers were fit and then tested on sample datasets in order to evaluate their

accuracy. Next, hyperparameter tuning was performed with Grid Search, so that optimal hyperparameters could be found for each classifier, and then their new parameter values were tested on sample data. Finally, sklearn's `classification_report` was used to evaluate each model based on several performance metrics.

### **Description of Datasets:**

In total, four datasets were used for this project. The first dataset, Amazon Fine Food Reviews, consists of a list of customer reviews from Amazon of various food items, as well as ratings from 1 to 5. We decided to exclude any neutral reviews.

The second dataset, Steam Reviews, is a list of player reviews on Steam for the games Battlefield 2042, The Binding of Isaac: Rebirth, Counter Strike 2, Monster Hunter Wilds, and Stardew Valley. These reviews are categorized as either being positive or negative reviews.

The third dataset, RateMyProfessor Reviews, is a collection of reviews on the website RateMyProfessor from between 2000 and 2018. The reviews contain a score from from 1 to 5. The website describes 3.4 as an average score, so all reviews ranked 3.4 or higher are classified as “positive”, while all other reviews are classified as “negative.”

The fourth dataset, Tweets, is a collection of posts from x.com with an attached “polarity” from 0 to 4, indicating the overall “positivity” of the tweet.

### **Results:**

## Amazon Fine Food Reviews dataset

```
--- Multinomial Naive Bayes Evaluation Results ---
Accuracy: 0.8862

Classification Report:
              precision    recall  f1-score   support

     0           0.90       0.31       0.46       12306
     1           0.89       0.99       0.94       66567

 accuracy          0.89          0.89          0.89       78873
 macro avg         0.89          0.65          0.70       78873
 weighted avg      0.89          0.89          0.86       78873
```

```
Logistic Regression      Train Accuracy: 0.92344      Test Accuracy: 0.90200
Classification Report:
              precision    recall  f1-score   support

     0           0.89       0.43       0.58         629
     1           0.90       0.99       0.94       3371

 accuracy          0.90          0.90          0.90       4000
 macro avg         0.89          0.71          0.76       4000
 weighted avg      0.90          0.90          0.89       4000
```

```
Linear SVC      Train Accuracy: 0.99150      Test Accuracy: 0.91750
Classification Report:
              precision    recall  f1-score   support

     0           0.80       0.63       0.71         629
     1           0.93       0.97       0.95       3371

 accuracy          0.92          0.92          0.92       4000
 macro avg         0.87          0.80          0.83       4000
 weighted avg      0.91          0.92          0.91       4000
```


```
Support Vector Machine   Train Accuracy: 0.94819           Test Accuracy: 0.90650
Classification Report:
              precision    recall  f1-score   support

     0           0.83       0.51       0.63         629
     1           0.91       0.98       0.95        3371

 accuracy                   0.91         4000
 macro avg              0.87       0.74       0.79         4000
weighted avg              0.90       0.91       0.90         4000
```

```
--- Transformer Test Results ---
Loss:      0.0908
Accuracy:   0.9725
Precision:  0.9832
Recall:     0.9842
F1 Score:   0.9837
Runtime:    47.62 seconds
Samples/sec:1656.46
Steps/sec:  103.54
--- Test Evaluation Complete ---
```

## Steam Reviews Dataset

	Logistic Regression		Train Accuracy: 0.88650		Test Accuracy: 0.75800
	Classification Report:				
		precision	recall	f1-score	support
	0	0.77	0.74	0.75	500
	1	0.75	0.78	0.76	500
	accuracy			0.76	1000
	macro avg	0.76	0.76	0.76	1000
	weighted avg	0.76	0.76	0.76	1000

```
-----
Logistic Regression:
max_iter: 500
solver: 'newton-cg'
-----
Logistic Regression      Train Accuracy: 0.88650      Test Accuracy: 0.75800
Classification Report:
      precision    recall  f1-score   support

     0       0.77       0.74       0.75        500
     1       0.75       0.78       0.76        500

 accuracy                   0.76        1000
 macro avg       0.76       0.76       0.76        1000
 weighted avg    0.76       0.76       0.76        1000
-----
```

```

→ Linear SVC      Train Accuracy: 0.95550      Test Accuracy: 0.73800
Classification Report:
              precision    recall  f1-score   support

         0           0.75       0.71       0.73         500
         1           0.73       0.76       0.74         500

    accuracy          0.74          1000
   macro avg          0.74          1000
  weighted avg          0.74          1000

```

```

-----
LinearSVC:
C: 0.1
max_iter: 500
-----
Linear SVC      Train Accuracy: 0.88325      Test Accuracy: 0.75500
Classification Report:
              precision    recall  f1-score   support

         0           0.77       0.73       0.75         500
         1           0.74       0.78       0.76         500

    accuracy          0.76          1000
   macro avg          0.76          1000
  weighted avg          0.76          1000
-----

```

```

→ SGDC Classifier Train Accuracy: 0.94575 Test Accuracy: 0.72400
Classification Report:
      precision    recall  f1-score   support

     0       0.74      0.69      0.71      500
     1       0.71      0.76      0.73      500

 accuracy          0.72      1000
 macro avg         0.73      0.72      0.72      1000
 weighted avg      0.73      0.72      0.72      1000

```

```

SGDCClassifier:
alpha: 0.001
loss: 'squared_error'
max_iter: 1500
-----
SGDC Classifier Train Accuracy: 0.89500 Test Accuracy: 0.75400
Classification Report:
      precision    recall  f1-score   support

     0       0.77      0.73      0.75      500
     1       0.74      0.78      0.76      500

 accuracy          0.75      1000
 macro avg         0.75      0.75      0.75      1000
 weighted avg      0.75      0.75      0.75      1000
-----

```

```

→ Naive Bayes Classifier Train Accuracy: 0.88700 Test Accuracy: 0.74000
Classification Report:
      precision    recall  f1-score   support

     0       0.72      0.78      0.75      500
     1       0.76      0.70      0.73      500

 accuracy          0.74      1000
 macro avg         0.74      0.74      0.74      1000
 weighted avg      0.74      0.74      0.74      1000

```

```

-----
Naive Bayes:
alpha: 0.8
fit_prior: True
force_alpha: True
-----
Naive Bayes Classifier  Train Accuracy: 0.89000      Test Accuracy: 0.74300
Classification Report:
              precision    recall  f1-score   support

     0           0.73       0.78       0.75         500
     1           0.76       0.71       0.73         500

 accuracy          0.74       0.74       0.74        1000
 macro avg         0.74       0.74       0.74        1000
 weighted avg      0.74       0.74       0.74        1000

```

```

--- Transformer Test Results ---
Loss:      0.5446
Accuracy:  0.7512
Precision:  0.7506
Recall:     0.7525
F1 Score:   0.7516
Runtime:    185.42 seconds
Samples/sec:4.32
Steps/sec:  0.27
--- Test Evaluation Complete ---

```

RateMyProfessor Reviews dataset

```

Classifier      Feature Extr.  Train Acc.  Test Acc.
-----
Naive Bayes     TF-IDF         0.81144     0.76425
-----
Classification Report:
              precision    recall  f1-score   support

    -1.0       0.95       0.35       0.51        1398
     1.0       0.74       0.99       0.85        2602

 accuracy          0.76       0.76       0.76        4000
 macro avg         0.84       0.67       0.68        4000
 weighted avg      0.81       0.76       0.73        4000

```

w/ Hyperparams

```
alpha: 0.6
```



```
fit_prior: False
```

```
force_alpha: True
```

Classifier	Train Acc.	Test Acc.		
Naive Bayes	0.84937	0.79725		
-----				
Classification Report:				
	precision	recall	f1-score	support
-1.0	0.92	0.45	0.61	1380
1.0	0.77	0.98	0.86	2620
accuracy			0.80	4000
macro avg	0.85	0.72	0.73	4000
weighted avg	0.82	0.80	0.77	4000

Classifier	Feature Extr.	Train Acc.	Test Acc.	
SVM	TF-IDF	0.89556	0.84025	
Classification Report:				
	precision	recall	f1-score	support
-1.0	0.81	0.71	0.75	1380
1.0	0.86	0.91	0.88	2620
accuracy			0.84	4000
macro avg	0.83	0.81	0.82	4000
weighted avg	0.84	0.84	0.84	4000

w/ Hyperparams

```
alpha: 0.001
```

```
loss: 'squared_error'
```

```
max_iter: 1500
```

Classifier	Train Acc.	Test Acc.		
SVM	0.85156	0.82850		
Classification Report:				
	precision	recall	f1-score	support
-1.0	0.85	0.61	0.71	1380
1.0	0.82	0.94	0.88	2620
accuracy			0.83	4000
macro avg	0.83	0.78	0.79	4000
weighted avg	0.83	0.83	0.82	4000

Classifier	Feature Extr.	Train Acc.	Test Acc.	
Log Regression	TF-IDF	0.88100	0.84000	
Classification Report:				
	precision	recall	f1-score	support
-1.0	0.82	0.69	0.75	1380
1.0	0.85	0.92	0.88	2620
accuracy			0.84	4000
macro avg	0.83	0.80	0.82	4000
weighted avg	0.84	0.84	0.84	4000

w/ Hyperparams

```
max_iter: 500
```

```
solver: 'liblinear'
```

Classifier	Train Acc.	Test Acc.			
Log Regression	0.88144	0.83950			
Classification Report:					
	precision	recall	f1-score	support	
-1.0	0.82	0.69	0.75	1380	
1.0	0.85	0.92	0.88	2620	
accuracy			0.84	4000	
macro avg	0.83	0.80	0.81	4000	
weighted avg	0.84	0.84	0.84	4000	

Classifier	Feature Extr.	Train Acc.	Test Acc.	
SVC	TF-IDF	0.93881	0.82450	
Classification Report:				
	precision	recall	f1-score	support
-1.0	0.77	0.70	0.73	1380
1.0	0.85	0.89	0.87	2620
accuracy			0.82	4000
macro avg	0.81	0.80	0.80	4000
weighted avg	0.82	0.82	0.82	4000

w/ Hyperparams

C: 0.1

max\_iter: 500

-----				
Classifier	Train Acc.	Test Acc.		
SVC	0.88431	0.84175		
-----				
Classification Report:				
	precision	recall	f1-score	support
-1.0	0.82	0.69	0.75	1380
1.0	0.85	0.92	0.88	2620
accuracy			0.84	4000
macro avg	0.84	0.81	0.82	4000
weighted avg	0.84	0.84	0.84	4000

```

--- Transformer Test Results ---
Loss:      0.3661
Accuracy:  0.8167
Precision: 0.8847
Recall:    0.8312
F1 Score:  0.8571
Runtime:   127.23 seconds
Samples/sec:4.72
Steps/sec: 0.30
--- Test Evaluation Complete ---

```

## Tweets datasets

Without Hyperparameters:

```
Logistic Regression      Train Accuracy: 0.86675      Test Accuracy: 0.75300
Classification Report:
              precision    recall  f1-score   support

     0           0.77       0.74       0.75       2021
     1           0.74       0.77       0.76       1979

 accuracy                   0.75       4000
 macro avg           0.75       0.75       0.75       4000
 weighted avg        0.75       0.75       0.75       4000
```

```
Linear SVC      Train Accuracy: 0.97469      Test Accuracy: 0.74475
Classification Report:
              precision    recall  f1-score   support

     0           0.75       0.74       0.75       2021
     1           0.74       0.75       0.74       1979

 accuracy                   0.74       4000
 macro avg           0.74       0.74       0.74       4000
 weighted avg        0.74       0.74       0.74       4000
```

```
Support Vector Machine  Train Accuracy: 0.88581      Test Accuracy: 0.75250
Classification Report:
              precision    recall  f1-score   support

     0           0.76       0.74       0.75       2021
     1           0.74       0.77       0.75       1979

 accuracy                   0.75       4000
 macro avg           0.75       0.75       0.75       4000
 weighted avg        0.75       0.75       0.75       4000
```

Naive Bayes Classification Report:		Train Accuracy: 0.91394		Test Accuracy: 0.74100	
		precision	recall	f1-score	support
0		0.71	0.81	0.76	2021
1		0.78	0.67	0.72	1979
accuracy				0.74	4000
macro avg		0.75	0.74	0.74	4000
weighted avg		0.75	0.74	0.74	4000

With Hyperparameters:

Logistic Regression max_iter: 500 solver: 'liblinear'					
-----					
Logistic Regression Classification Report:		Train Accuracy: 0.86787		Test Accuracy: 0.75200	
		precision	recall	f1-score	support
0		0.76	0.74	0.75	2021
1		0.74	0.77	0.75	1979
accuracy				0.75	4000
macro avg		0.75	0.75	0.75	4000
weighted avg		0.75	0.75	0.75	4000

Linear SVC C: 0.1 max_iter: 500					
-----					
Linear SVC Classification Report:		Train Accuracy: 0.86556		Test Accuracy: 0.75350	
		precision	recall	f1-score	support
0		0.77	0.73	0.75	2021
1		0.74	0.78	0.76	1979
accuracy				0.75	4000
macro avg		0.75	0.75	0.75	4000
weighted avg		0.75	0.75	0.75	4000

```
Support Vector Machine
alpha: 0.001
loss: 'squared_error'
max_iter: 1000
```

```
-----
Support Vector Machine      Train Accuracy: 0.81112      Test Accuracy: 0.74775
Classification Report:
              precision    recall  f1-score   support

         0           0.77       0.72       0.74        2021
         1           0.73       0.78       0.75        1979

 accuracy                   0.75         4000
  macro avg                 0.75         4000
 weighted avg               0.75         4000
```

```
Naive Bayes
alpha: 1.0
fit_prior: True
force_alpha: True
```

```
-----
Naive Bayes      Train Accuracy: 0.91394      Test Accuracy: 0.74100
Classification Report:
              precision    recall  f1-score   support

         0           0.71       0.81       0.76        2021
         1           0.78       0.67       0.72        1979

 accuracy                   0.74         4000
  macro avg                 0.75         4000
 weighted avg               0.75         4000
```

```
--- Transformer Test Results ---
Loss:      0.3287
Accuracy:  0.8598
Precision: 0.8612
Recall:    0.8578
F1 Score:  0.8595
Runtime:   72.33 seconds
Samples/sec:1659.17
Steps/sec: 103.70
--- Test Evaluation Complete ---
```

## **Analysis and Conclusions:**

### **Amazon Dataset**

Multinomial NB shows a decent overall accuracy at 0.8862. However, its performance is heavily skewed towards classifying as positive. It has a very high recall for class 1 at 0.99 meaning it correctly identifies most positive reviews but struggles with class 0, exhibiting very low recall at 0.31. This suggests the model is prone to classifying negative reviews as positive.

Logistic Regression shows improved overall accuracy at 0.92 compared to NB. It also shows better performance on class 0 with higher recall at 0.43 and F1-score at 0.58. However, it still exhibits a significant imbalance in performance between the two classes, heavily favoring Class 1 (positive reviews) in terms of recall at 0.99. Similar to NB, it's better at identifying positive reviews than negative ones.

Linear SVC has a similar overall accuracy to Logistic Regression at 0.9175. Notably, it shows a considerable improvement in identifying Class 0 (negative) instances, with a much higher recall (0.63 vs 0.97) and F1-score (0.71 vs 0.95) between the two classes is smaller, making it more balanced model than NB or Logistic Regression for this dataset.

With an accuracy of 0.90650, the Support Vector Machine model performs slightly below Logistic Regression (0.92) and Linear SVC (0.91750) in terms of overall accuracy. Its class-specific metrics remain as previously analyzed: strong performance on Class 1 (high recall and F1) but weaker performance on Class 0 (lower recall and F1) compared to its performance on Class 1 and also compared to the Class 0 performance of Linear SVC.

The Transformer (BERT) model significantly outperforms all the other models across the board. It boasts the highest accuracy (0.9725) and extremely high precision (0.9832), recall (0.9842), and F1-scores (0.9837). The metrics are macro-averaged. This is expected from large pre-trained transformer models on many NLP tasks.

In conclusion, while traditional models like Linear SVC can provide reasonable performance, especially when considering balanced performance across classes, the Transformer model demonstrates the significant leap in capability offered by more modern, pre-trained architectures for this type of NLP task.

## Steam Dataset

Pre-hyperparameter tuning, Logistic Regression having the highest test accuracy at 75.8%, despite it having the lowest training accuracy across the board with an 88.6%.

Model	Train Accuracy	Test Accuracy
Logistic Regression	0.88650	0.75800
Linear SVC	0.95550	0.73800
SGD Classifier	0.94575	0.72400
Naive Bayes	0.88700	0.74000

Post-hyperparameter tuning had Logistic Regression's position and performance being unchained, which indicates that the default parameters were the optimal parameters. The other models showed reduced training accuracy with improved test accuracy, with SGDClassifier notably improving its test accuracy by 3%, with the rest of the models having test accuracies around 75%.



Model	Train Accuracy	Test Accuracy
Logistic Regression	0.88650	0.75800
Linear SVC	0.88325	0.75500
SGD Classifier	0.89500	0.75400
Naive Bayes	0.89000	0.74300

For the F1-score, Logistic Regression, LinearSVC, and SGDClassifier performed similarly, with them having around .75 for negative and .76 for positive reviews. Naive Bayes slightly underperformed comparatively in positive reviews with a .73.

Precision for negative reviews had the highest for Logistic Regression, LinearSVC, and SGDClassifier, each having .77, while Naive Bayes lagged behind at .73. Surprisingly though, Naive Bayes had a slight lead for positive reviews, having a .76.

Surprisingly, Naive Bayes had the highest recall for negative reviews, having a .78, but had its positive review recall value lagging behind the rest with a .73, compared to the others being more balanced, having around ~ .75.

Overall, Logistic Regression was the best-performing model, with the highest test accuracy and smallest train/test accuracy gap of 13%, which suggests minimal overfitting. Naive Bayes came second, and benefited from its strong recall and an accuracy gap of 14.7%. In contrast, SGDClassifier and LinearSVC showed larger gaps in its accuracy, both being above 20%, which indicates potential overfitting.

Now comparing the Deep Learning Model with Logistic Regression, the deep learning model has an accuracy of 75.06%, being .74% worse than the Logistic Regression model. This is where the information is a bit muddled, but its Recall, Precision, and F1-score was much more tightly averaged than Logistic Regression, being around the .75 margin while Logistic Regression is from .74-.78. However, in terms of efficiency, training the model took around 2 hours and 30 minutes, and testing the model took 3 minutes, while the classifier models training/testing time

were near instant. Generally, the Transformer model won, but in terms of speed and scalability, Logistic Regression is king here.

```
--- Transformer Test Results ---  
Loss:      0.5446  
Accuracy:  0.7512  
Precision:  0.7506  
Recall:     0.7525  
F1 Score:   0.7516  
Runtime:    185.42 seconds  
Samples/sec: 4.32  
Steps/sec:  0.27  
--- Test Evaluation Complete ---
```

Logistic Regression		Test Accuracy: 0.75800			
Classification Report:					
	precision	recall	f1-score	support	
0	0.77	0.74	0.75	500	
1	0.75	0.78	0.76	500	
accuracy			0.76	1000	
macro avg	0.76	0.76	0.76	1000	
weighted avg	0.76	0.76	0.76	1000	

## RateMyProfessor Dataset

For the RateMyProfessor dataset, the accuracy consistently stays around 0.84 for testing. The training accuracy varies much more heavily than the testing accuracy, with the lowest training accuracy after tuning being 0.849 and the highest being 0.888. The tuning of hyperparameters (changing alpha to 0.5 and setting force\_alpha to true) greatly improved both the Naive Bayes classifier, causing its accuracy to rise from 0.765 to 0.835. Notably, the test accuracy for Logistic Regression actually decreased from 0.84225 to 0.84200, showing a very small, but curious, slip. This could potentially be caused by setting the solver to “liblinear”, which may have performed better on the small subset selected to serve as training data. The best overall performance in terms of test accuracy was with Linear SVC after hyperparameter tuning, with C set to 0.1 and maximum iterations at 500. It achieved a test accuracy of 0.84475, giving it a small edge over the other tested classifiers. Each classifier had around an 0.83 weighted average F1 score, with Logistic Regression and Linear SVC taking a very small lead of 0.1 over Support Vector Machine and Naive Bayes. The same is true for recall, as the weighted average of Linear SVC and Logistic Regression were ahead by 0.1. The best weighted precision was with Linear SVC, achieving 0.85. Overall, it would appear that Linear SVC is the best performing of the used classifiers for this dataset, with Logistic Regression having a close second place.

## Tweets Dataset

For the tweets dataset, there is a general trend of each of the classifiers having a test accuracy around 0.75, with the highest test accuracy going to Logistic Regression with a score of 0.75350, and the lowest test accuracy going to the Linear SVC with an accuracy of 0.74475. Interestingly enough, Linear SVC had the highest Training Accuracy of the models used at 0.97469,

indicating that it overfitted the model to the training data. The various models seemed to have a similar display of f1 score throughout, with them being within the 0.74-0.76 range for 0 or 1 sentiment, with the only deviation here being in the naive Bayes classifier with an F1 score of 0.72. The recall and precision scores on the Naive Bayes classifier also varied wildly in comparison to the other classifiers, as while they recall and precision in the 0.74-0.77 range, the Naive Bayes had a precision(0) of 0.71, a recall(0) of 0.81, and a recall(1) of 0.67, meaning the model was overfitted to the examples given, and with it having a slightly higher preference to negative(0) sentiments(2021 vs 1979), it seemed to track into disproportionately valuing tweets as a negative sentiment. The highest test accuracy goes to the Linear SVC with a score of 0.75350, and the lowest Test accuracy goes to Naive Bayes with it being at 0.74100. Naive Bayes in this instance also has the highest training accuracy(0.91394) and the highest recall(0) and lowest recall(1)(0.81 and 0.67, respectively), indicating that the `fit_prior` led to it overfitting the model to the dataset and subsequently prioritizing negative [readings.In](#) comparing before the various models before implementing their hyperparameters in comparison to after, the training accuracy of the Logistic Regression Model went down when placing a limit on the maximum number of iterations(500). The training accuracy of the Linear SVC also went down, as well as its recall(though the precision(0) increased and the recall(1) dramatically increased), the Support Vector Machine testing accuracy went down, as did the precision(1) score, though the recall(1) increased from 0.77 to 0.78. Naive Bayes stayed the same in all metrics after implementing the `alpha` as 1.0, `fit_prior=true`, and `max_iter=1000`. Overall, the best model for the sentiment analysis of these tweets was most likely Logistical Regression, as not only was it the best performing models in terms of training accuracy prior to the implementation of hyperparameters,

it was still the 2nd best after the implementation, and had little shift in its performance metrics.