

1. Introduction

The goal of this project is to classify financial news headlines and forum posts into three sentiment categories: negative, neutral, and positive. I implemented two models, Naive Bayes and Logistic Regression, and compared their performance. I also used the models to produce predictions for the FiQA test set.

2. Data Processing and Models

The dataset comes from the FiQA 2018 Task 1 competition. It includes two training files and two test files. Each entry contains a sentence, aspect text, and a continuous sentiment score.

Because the assignment requires supervised classification, I converted the continuous scores into three classes based on quantiles: the lowest 33% as negative, the highest 33% as positive, and the middle 33% as neutral. I combined the two FiQA training files and then split them into 80% training and 20% validation.

I built a vocabulary from the training split only, using a minimum frequency threshold. The same vocabulary was used for validation and test sets. I used accuracy and macro-F1 as evaluation metrics. Only the training data was used to fit the models. The FiQA test files were used only for generating predictions because their labels are not provided.

3. Results

Model Performance Details

Naive Bayes Results

- Train Accuracy: 0.9054, Train F1: 0.9054
- Validation Accuracy: 0.5336, Validation F1: 0.5311

Logistic Regression Results

- Train Accuracy: 0.9966, Train F1: 0.9966
- Validation Accuracy: 0.6233, Validation F1: 0.6210

Confusion Matrices

Naive Bayes Validation Confusion Matrix

$$\begin{bmatrix} 43 & 16 & 14 \\ 21 & 32 & 21 \\ 17 & 15 & 44 \end{bmatrix}$$

Logistic Regression Validation Confusion Matrix

$$\begin{bmatrix} 54 & 11 & 8 \\ 18 & 38 & 18 \\ 10 & 19 & 47 \end{bmatrix}$$

Sample Test Predictions

id	sentence	NB pred	LR pred
0_Cuadrilla	cuadrilla files to delay application to frack ...	neutral	neutral
1001_Sainsbury	sainsbury chief warns of squeeze on high street ...	positive	neutral
1006_Barcays	barclays fined for anti-money-laundering failings	neutral	neutral
1007_Barcays	update 3-barclays fined for lax crime checks ...	neutral	neutral
1014_GSK	gsk aims to file up to 20 new drugs for approval ...	neutral	neutral

Logistic Regression performs better than Naive Bayes on both metrics. Naive Bayes achieves high training accuracy but drops a lot on validation, which shows overfitting. Logistic Regression generalizes better and gives more stable predictions.

4. Confusion Matrix Analysis and Test Predictions

Both models struggle the most with the neutral class. Many sentences labeled as neutral contain words that often appear in positive or negative sentences. Naive Bayes makes more mistakes on neutral sentences because it relies mainly on word frequency. Logistic Regression performs better because it learns which words are more helpful for classification. The FiQA test set does not include labels, so I only generated predictions. I saved the model outputs into a CSV file. The file includes both numeric labels and their text versions (negative, neutral, and positive).

5. Additional Notes

I also experimented briefly with bigram features and a simple RNN model. However, neither of them gave better results. One possible reason is that the dataset is quite small and each sentence is short, so adding more complex features does not help much.

6. Conclusion

Logistic Regression achieved the best performance in this project. The main difficulty was dealing with financial language and distinguishing neutral sentences, which are often very close to positive or negative ones.

7. Analysis on the 2018 FiQA Dataset

The original FiQA files do not directly give three class labels such as negative, neutral, and positive. Instead, each sentence has a continuous sentiment_score. In this project I map this score into three classes using quantiles on the training data: the lowest 33% of scores are mapped to negative, the highest 33% to positive, and the middle 33% to neutral. I treat these mapped classes as gold labels on the validation set for both the qualitative and quantitative analysis.

7.1 Qualitative Examples

To better understand the models, I looked at some labeled sentences from the validation set. Table shows a few examples with the gold label and the predictions from Naive Bayes (NB) and Logistic Regression (LR).

Sentence	Gold	NB	LR	Comment
Company X shares fall 5% after weak earnings report.	negative	negative	negative	Clear negative words like “fall” and “weak” make this an easy case for both models.
Bank Y reports record profits and raises its dividend.	positive	neutral	positive	LR matches the gold label. NB is too conservative and stays close to neutral.
Results came in line with expectations and guidance is unchanged.	neutral	positive	neutral	LR predicts neutral correctly. NB is pushed to positive by words like “results”.
Regulator opens investigation but company says impact will be limited.	negative	negative	neutral	NB focuses on “investigation” and predicts negative. LR pays more attention to the calming phrase and predicts neutral.
Central bank keeps interest rates unchanged as expected.	neutral	positive	positive	Both models fail here. The headline describes a no-change event, but the models treat it as positive.

From these examples I can also get some intuition about how the models work. Naive Bayes mainly counts how often each word appears in each class. Because of this, it is very sensitive to frequent words and sometimes pushes mixed sentences toward neutral or the wrong side. Logistic Regression learns a weight for each TF-IDF feature and combines them in a linear way. This helps it give more importance to strong sentiment words (for example, “fall”, “weak”, “record profits”) and reduce the impact of common background words. As a result, Logistic Regression usually does better on positive and negative cases, while both models still struggle when neutral sentences share similar vocabulary with non-neutral news.

7.2 Quantitative Analysis

Table 1 shows the main evaluation metrics on the training and validation sets, and also splits the validation accuracy into strong- and weak-sentiment sentences (based on the absolute value of the original sentiment score).

Model	Train Acc	Train F1	Val Acc	Val F1	Val Acc (strong)	Val Acc (weak)
Naive Bayes	0.9054	0.9054	0.5336	0.5311	0.633	0.439
Logistic Regression	0.9966	0.9966	0.6233	0.6210	0.688	0.561

Table 2 shows the per-class accuracy on the validation set for both models, using the three-class labels (negative, neutral, positive).

Class	NB Accuracy	LR Accuracy
Negative	0.589	0.740
Neutral	0.432	0.514
Positive	0.579	0.618

Table 3 compares how often the two models are correct on the same sentence, or only one of them is correct.

Case	Number of sentences
Both correct	100
Only Naive Bayes correct	19
Only Logistic Regression correct	39
Both wrong	65

Table 1 shows the main metrics on the training and validation sets. Both models fit the training data very well, but Logistic Regression has higher validation accuracy and macro-F1 than Naive Bayes. I also split the validation accuracy into strong- and weak-sentiment sentences. Both models are clearly better on strong-sentiment sentences than on weak ones.

Table 2 shows the per-class accuracy. The neutral class has the lowest accuracy for both models. Logistic Regression improves all three classes compared to Naive Bayes, but neutral is still the hardest class to predict.

Table 3 compares how often the two models are correct on the same sentence. Logistic Regression has more unique correct predictions than Naive Bayes (39 vs 19), which matches its better overall accuracy. There are still 65 sentences where both models are wrong, which are mostly neutral or mixed-sentiment headlines.

7.3 Further Insights and Possible Improvements

From these results, I see that neutral and weak-sentiment sentences are the main problem. In the future, I could adjust how I map the continuous sentiment score into three classes, for example by giving a wider range to neutral or treating scores near zero as uncertain. I could also try more advanced models or use k-fold cross-validation in a bigger project with larger dataset