Machine Learning Final Sentiment Classification

This project aimed to classify Data from FiQA and Financial PhraseBank into Negative, Neutral and Positive sentiments. The original data consisted of 5322 unique sentences divided as follows: 53.6% Neutral, 31.7% Positive and 14.7% Negative.

In order to complete this project we chose to focus on the following 4 machine learning algorithms:

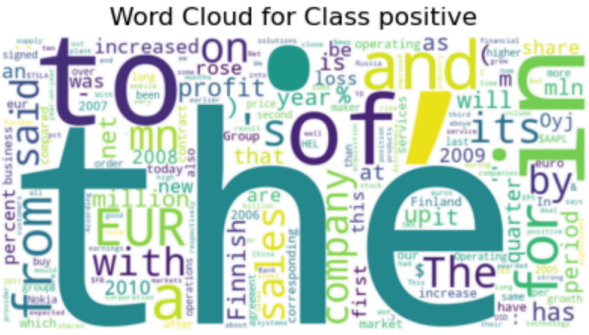
1. K-Nearest-Neighbors (KNN)
2. Decision Tree Classifier
3. Support Vector Machine (SVM)
4. Logistic Regression

We began by transforming our data from textual into numerical using TF-IDF and by that it should help us find the most important/ relevant word in the text. The next step was running our 4 models on the data before any other preprocessing so we can compare it to the results after preprocessing and post hyper parameter tuning.

Without any preprocessing we noticed that the best classification model is Logistic Regression with 72% accuracy followed by SVM with 69% accuracy, KNN with 63% and Decision Tree with 58%. The accuracy is fairly high considering the fact that no preprocessing was done yet and another important thing we noticed is the fact that the precision and recall for the Negative class was very low (between 10 and 49%) compared to Positive and Neutral classes. This shows us how important a balanced data is and also explains the high accuracy as even a dummy model would receive over 54% accuracy.

In order to address the unbalanced classes issue we explored two different methods which are Downsampling and Upsampling.
The first method explored is downsampling where we choose to do a cluster sampling as it better preserves the distribution, after downsampling the Neutral and Positive classes we ended with 1000 samples of each and 859 samples of the Negative class.
We then ran the algorithms on the downsampled data and noticed that the accuracy has greatly decreased to Logistic Regression with 62%, SVM with 64%, KNN with 53% and Decision Tree with 46%.

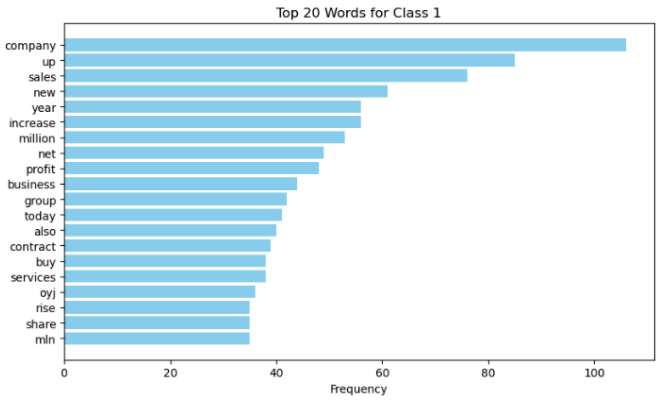
After plotting the word cloud where the words that popped out the most (meaning they are found the most in the data) are words that aren't too useful for the classification such as 'the', 'of', 'to', 'from', these words are most commonly known as Stopwords which we would like to remove so the important words receive a higher emphasize.

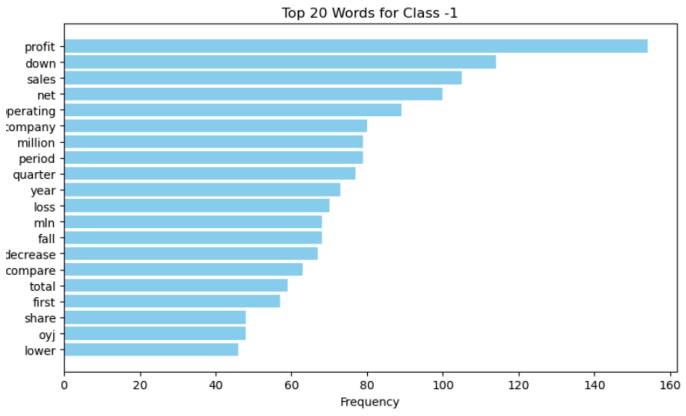


Another important adjustment that was made to the dataset is converting adjectives to present tense which could potentially help the model classify better as otherwise the words 'Increase' and 'Increased' are different when converting the words using TF-IDF.

After having adjusted our data, we trained various models such as KNN, Logistic Regression, Support Vector Machine and Decision Tree with GridSearch for each model, to find the best hyper parameters which will best fit our processed data.
We found out that in the downsampled processed data, the model that performed the best was SVM with ~64% accuracy. followed by Logistic Regression. Moreover, in the upsampled processed data, both Logistic Regression and SVM gave about ~69% accuracy.
Overall it is safe to say that for this problem, SVM is the best classifier for this data which makes sense as SVM is highly recommended to these kinds of problems as it performs well in high-dimensional spaces because it seeks the optimal hyperplane that maximizes the margin between different classes.
The Decision Tree classifier could very easily overfit and give too much attention to noise when training which would negatively affect the performance of the model. KNN too suffers from noise as it tries to find the nearest neighbors which could be misleading.

A significant aspect of this project involved making decisions throughout the entire process. One of the most crucial challenges arose during data preprocessing. The dataset consisted of sentences ranging from 2 to over 300 words, and the key difficulty was identifying which words to remove to improve model predictions without discarding potentially relevant terms. Although we removed some stopwords and irrelevant terms, such as country names, we still encountered words like "company" and "group," which are neutral and not inherently positive or negative. Another challenging word was "profit," which appeared in both the Positive and Negative classes. Surprisingly, "profit" was more frequently found in the Negative class than in the Positive class, which could be misleading had we removed the words ‘down’ or ‘loss’ for example.





Deciding on how we should deal with the unbalanced was another decision we had to make, removing some of the data does help balancing it, however it means that a lot of the data is being missed. We had to decide what ‘precision’ and ‘recall’ for the Negative class is too low and therefore the model isn’t actually learning the data and still manages to score a decent accuracy.

The following table summarizes the results of the 5 different trials conducted.
Note that

- Precision = Actual Negative / All sentences that were predicted Negative
- Recall = Actual Negative / All sentences that are actually Negative

	Original Data							
Model	KNN	DT	SVM	LR				
Accuracy	0.63	0.58	0.69	0.72				
Recall	0.23, 0.81, 0.53	0.17, 0.68 ,0.59	0.09, 0.90, 0.63	0.14, 0.9, 0.68				
Precision	0.27, 0.71, 0.64	0.15, 0.66, 0.66	0.28, 0.69, 0.78	0.14, 0.71, 0.77				
	Down sampled Data							
	Raw Data				Preprocessed Data			
Model	KNN	DT	SVM	LR	KNN	DT	SVM	LR
Accuracy	0.53	0.47	0.64	0.62	0.59	0.59	0.64	0.63
Recall	0.69, 0.51, 0.40	0.48, 0.46, 0.45	0.57, 0.67, 0.67	0.53, 0.68, 0.63	0.71, 0.56, 0.52	0.61, 0.66, 0.47	0.62, 0.61, 0.67	0.59, 0.65, 0.64
Precision	0.46, 0.59, 0.56	0.44, 0.47, 0.48	0.65, 0.65, 0.63	0.64, 0.61, 0.62	0.54, 0.65, 0.58	0.63, 0.54. 0.61	0.63, 0.65, 0.65	0.66, 0.62, 0.62
	Upsampled Data							
	Raw Data				Preprocessed Data			
Model	KNN	DT	SVM	LR	KNN	DT	SVM	LR
Accuracy	0.6	0.56	0.7	0.69	0.63	0.63	0.69	0.69
Recall	0.38, 0.75, 0.55	0.34, 0.63, 0.6	0.37, 0.85, 0.72	0.36, 0.84, 0.7	0.32, 0.9, 0.51	0.41, 0.75, 0.61	0.36, 0.85, 0.69	0.37, 0.85, 0.7
Precision	0.38, 0.68, 0.62	0.31, 0.65, 0.62	0.57, 0.71, 0.75	0.57, 0.7, 0.72	0.55, 0.65, 0.64	0.46, 0.68, 0.66	0.54, 0.69, 0.76	0.59, 0.69, 0.74

Looking at the confusion matrices created by the algorithms in the first part, we can see that barely any of the data was predicted Negative (as suspected). Balancing the data helped improve both recall and precision suggesting the model was actually improved even though the accuracy has clearly decreased.

Another topic that would be interesting to explore is the robustness of the models. Note that the SVM's accuracy has stayed constant throughout the 5 different trials, cleaning the data didn't increase the models accuracy as much as the other models. As mentioned before the SVM model handles outliers better than the rest of the models and this shows us just how much.
Downsampling has decreased the accuracy of all the models, some more and some less and preprocessing has increased the accuracy of all models but SVM.

Last, but not least, another question we decided to explore is how well can deep learning classify these sentences. A basic 3 dense layers with a ReLU activation function was used and managed to classify the data at 62% accuracy which is just around the average.