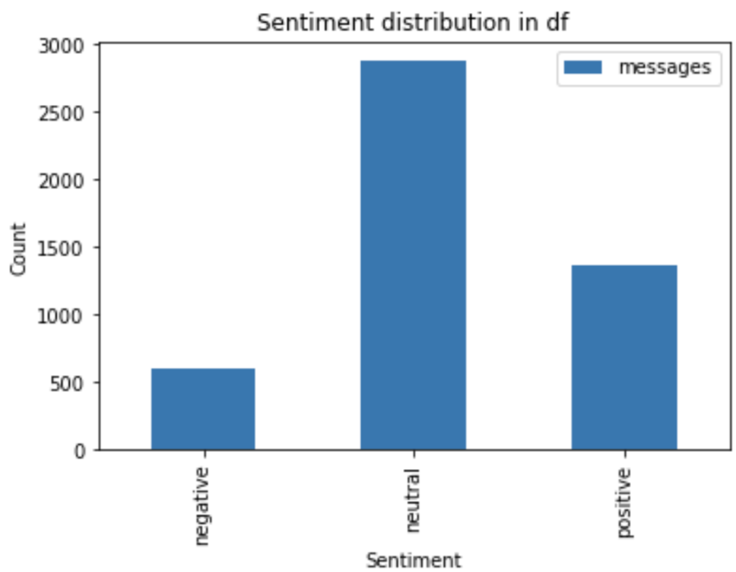
Financial News - Sentiment Analysis

1. Data

The following data is intended for advancing financial sentiment analysis research. It's two datasets (FiQA, Financial PhraseBank) combined into one easy-to-use CSV file. It provides financial sentences with sentiment labels.

https://www.kaggle.com/datasets/sbhatti/financial-sentiment-analysis



2. Method

There are three main types of text classification models

1. **BoW | Count | Tf-idf:** term frequency-inverse document frequency, a statistical measure that evaluates how relevant a word is to a document. If a word appears in every headline (aside from removed stopwords), then tf-idf reduces the weight of the term since it is not highly relevant for understanding the headline. On the other hand, if a term appears only a few times then the word has greater significance to the meaning of the headline.

2. **Word2Vec:** word embeddings in a word2vec model do not rely on the 'count' of the terms, as in bag-of-words, countvectorizer, and tfidf. Instead, word2vec extracts syntactic and semantic information by extracting the term's context in relation to other terms.

Thus, the classic example of subtracting the vector that corresponds to ‘man’ from the vector that corresponds to the term ‘king’ yields the vector ‘queen’. This is established simply by the usage of the terms in relation to each other, and not by their *inherent* meaning.

3. **BERT | Transformer Learning:** Transformers are a deep learning model in which every output element is connected to every input element, and the weightings between them are dynamically calculated based upon their connection. (In NLP, this process is called attention.) BERT, which stands for Bidirectional Encoder Representations from Transformers, is based on Transformers. Historically, language models could only read text input sequentially -- either left-to-right or right-to-left -- but couldn't do both at the same time. BERT is different because it is designed to read in both directions at once. This capability, enabled by the introduction of Transformers, is known as bidirectionality. Using this bidirectional capability, BERT is pre-trained on two different, but related, NLP tasks: Masked Language Modeling and Next Sentence Prediction.

Winner: Hugging face transformers

Hugging face transformers have made sentiment analysis a seamless process, with little effort in implementation. Previously, tf-idf and word2vec models could be used along with a neural network, but this required a large amount of training data and is expensive computationally. There are many reasons to use a pretrained model, but computational costs as well as developer time are the main reasons to utilize a pretrained model.

3. Data Preparation:

Text normalization process:

- Expand contractions
- Lowercase
- Stop word removal
- Lemmatize

4. EDA

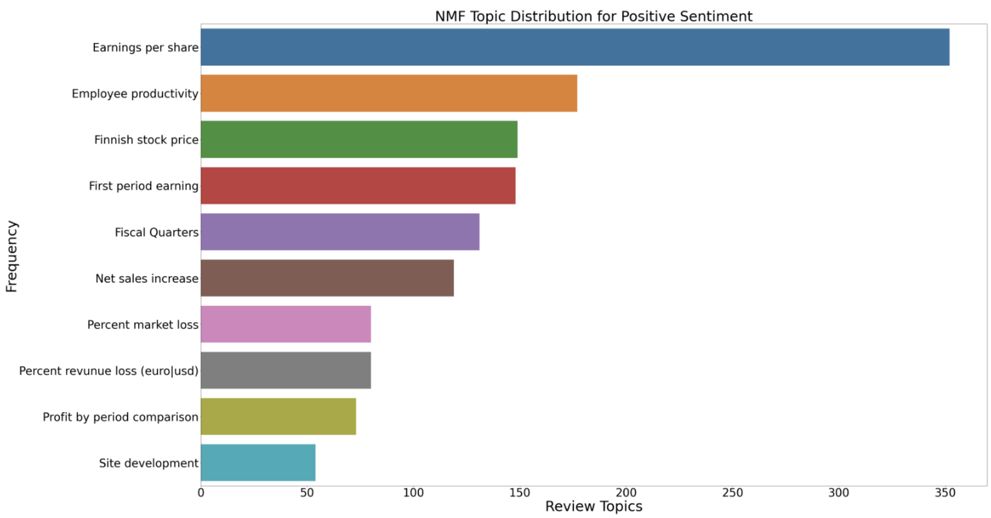
As the dataset is a series of headlines, typical EDA methods (e.g., length, a longer review when a customer is dissatisfied) are rendered generally useless. Latent Dirichlet Allocation (LDA) Topic Modeling, an unsupervised clustering method, rendered interesting results, however. The method returns a set of terms that represent topics across the body of the text.

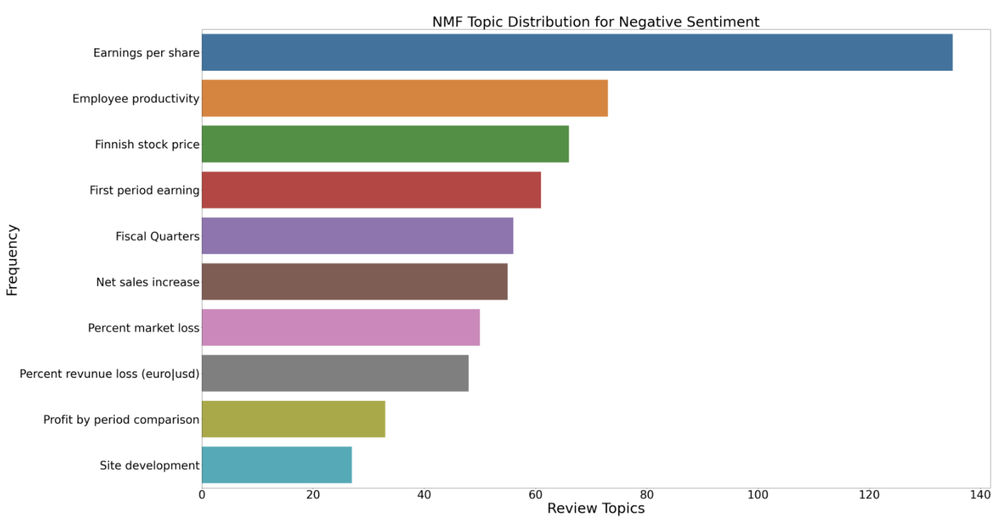
e.g., “share per capital earning eur right eps voting number option” was returned as topic 3, which in turn was flagged as ‘earnings per share’.

Identifying topics returned by LDA is best left to domain experts; however, this provides an interesting opportunity for data scientists and domain experts to collaborate on identifying overall topic trends.

<u>Method</u>:

- Identify topics with LDA, specifying ‘n_components=10` for the top ten subjects
- Name topic groupings and map to text
- Subquery by sentiment and take value counts of topics.





<u>Results</u>:

Notably, there is more discussion of revenue loss and overall market loss in negative sentiment. Again, the accuracy of these topics depends heavily on domain knowledge, but it is a sign that the topic modeling method is accurately parsing topics.

5. Modeling

FinancialBERT is a BERT model pre-trained on a large corpora of financial texts. The model was fine-tuned for Sentiment Analysis task on Financial PhraseBank dataset. Experiments show that this model outperforms the general BERT and other financial domain-specific models.

> More details on FinancialBERT's pre-training process can be found at: https://www.researchgate.net/publication/358284785_FinancialBERT_-_A_Pretrained_Language_Model_for_Financial_Text_Mining
>
> Training data FinancialBERT model was fine-tuned on Financial PhraseBank, a dataset consisting of 4840 Financial News categorised by sentiment (negative, neutral, positive).

Fine-tuning hyper-parameters:

learning_rate = 2e-5 batch_size = 32 max_seq_length = 512 num_train_epochs = 5 Evaluation metrics The evaluation metrics used are: Precision, Recall and F1-score. The following is the classification report on the test set.

NLP pipeline:

```
from transformers import BertTokenizer, BertForSequenceClassification
from transformers import pipeline

model = BertForSequenceClassification.from_pretrained("ahmedrachid/FinancialBERT-Sentiment-Analysis",num_labels=3)
tokenizer = BertTokenizer.from_pretrained("ahmedrachid/FinancialBERT-Sentiment-Analysis")

nlp = pipeline("sentiment-analysis", model=model, tokenizer=tokenizer)
```

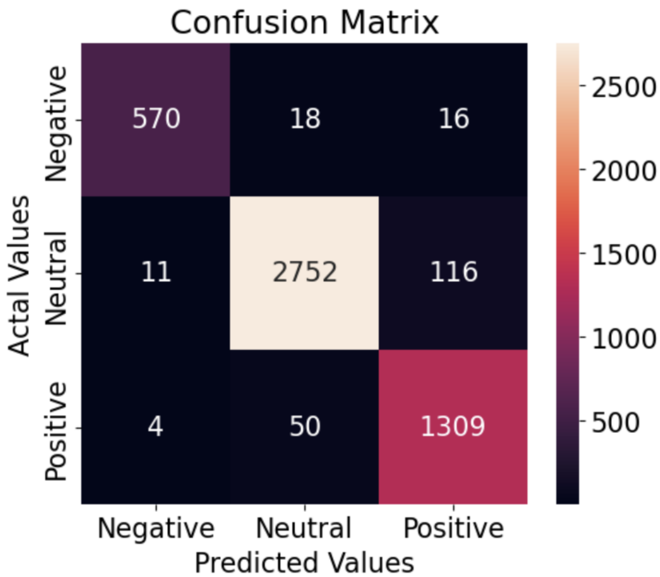
Classification report:

```
print(classification_report(y_true_finbert, y_pred_finbert, zero_division=0))
```

```
              precision    recall  f1-score   support

    negative       0.97      0.94      0.96       604
     neutral       0.98      0.96      0.97      2879
    positive       0.91      0.96      0.93      1363

    accuracy                           0.96      4846
   macro avg       0.95      0.95      0.95      4846
weighted avg       0.96      0.96      0.96      4846
```



FinBert resulted in:

570 true negative, 34 false negative

2752 true negative, 127 false neutral

1309 true positive, 54 false positive

Results:

Default Distilbert performs better on negative sentiments, which suggests it's a better model generally speaking; even despite lacking a 'neutral' classifier. Overall performance is clearly better due to the FinBERT model having a neutral classifier (Default Distilbert is only 28% accurate on the text data, while FinBERT is 95% accurate). The FinBERT model performed worst on a negative sentiment (with 5.5 % inaccurate suggestions) compared to the 2.3% inaccurate suggestions of Default DistilBERT.