Stock News Classification, Prediction, & Analysis

BY: Vincent Welsh
11-24-2021

Outline

- Business Problem & Understanding
- Data
- Modeling Methodology & Metrics
- Results
  - Title and Text sentiment analyses are equally important for machine learning
  - Optimal weights to use to deal with class imbalance
  - Best model for stock news classification
- Conclusions
- Further Work

Business Problem

- Predicting stock prices is a highly valuable asset to have. Assuming high precision in our predictions, we can:
    - Profit from short or long term stock purchases
    - Analyze company projected growth or decline
    - Determine public sentiment toward a company or industry

- The Assumption of this project is that stock news is related to stock price. This assumption is based on:
    - How stock prices move up or down
    - Public sentiment toward a stock and stock price

Business Understanding

- Understanding the value that comes from stock prediction, I went into this project with the following goals:
    - Build various machine learning models predicting stock price with precision and accuracy as the goal metrics.
        - Value: If we have a model that can confidently predict a gain or loss for a given stock then we can invest company funds with higher confidence
    - Derive Importance Results
        - Value: Is the headline of a news article have more valuable information than the text? Knowing this will allow for more advanced models in the future
    - Visualize Relationships
        - Value: By creating a visual representation of the relationship between stock news and a given stock we can detect patterns which can be used to enhance model performance

Data

- All of the data was retrieved from the FMP Cloud API (https://fmpcloud.io/)
- Preprocessing:
    - Obtain Stock News using api key, limit (Number of News articles to retrieve) stock symbol list
        - This returns the title and text of the news, date + time published
        - Custom Features:
            - Vader_sentiment_scores (For first two models)
            - Boolean is_weekday (True if on Friday, Saturday, Sunday)
            - Weekday number (0 = Monday)
            - Stock Prediction Day
    - Obtain stock price using the date range from the stock news dataframe above

NLP

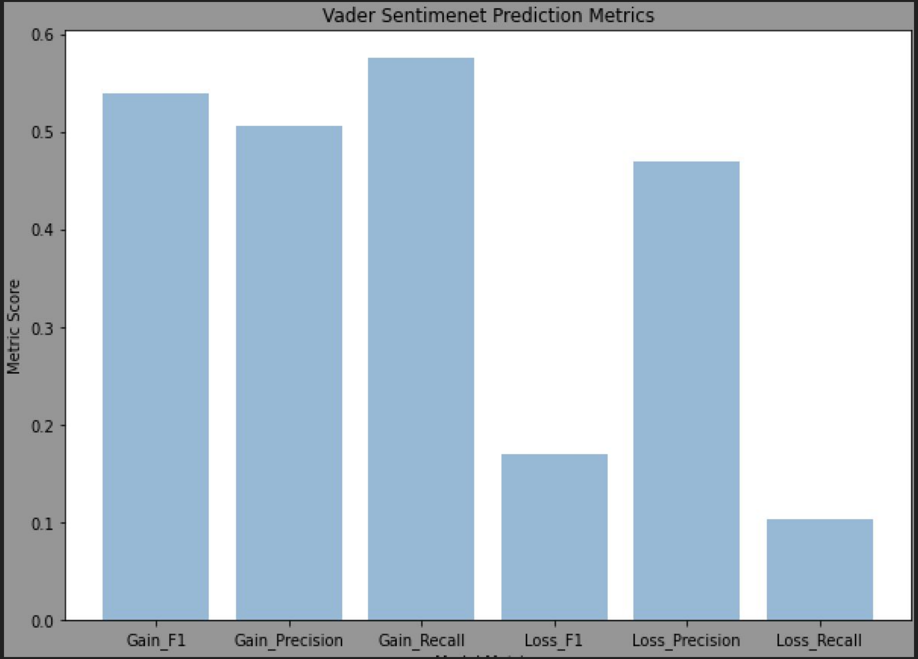
- For the recurrent networks and transformer model I took the following natural language process steps
  - Combine title and text of a news article
  - Count unique words
  - Determine max length of a given text
  - Tokenize words (removes punctuation)
  - Embed words using Glove word vector matrix
    - (https://nlp.stanford.edu/projects/glove/)

Methodology (Baseline Models)

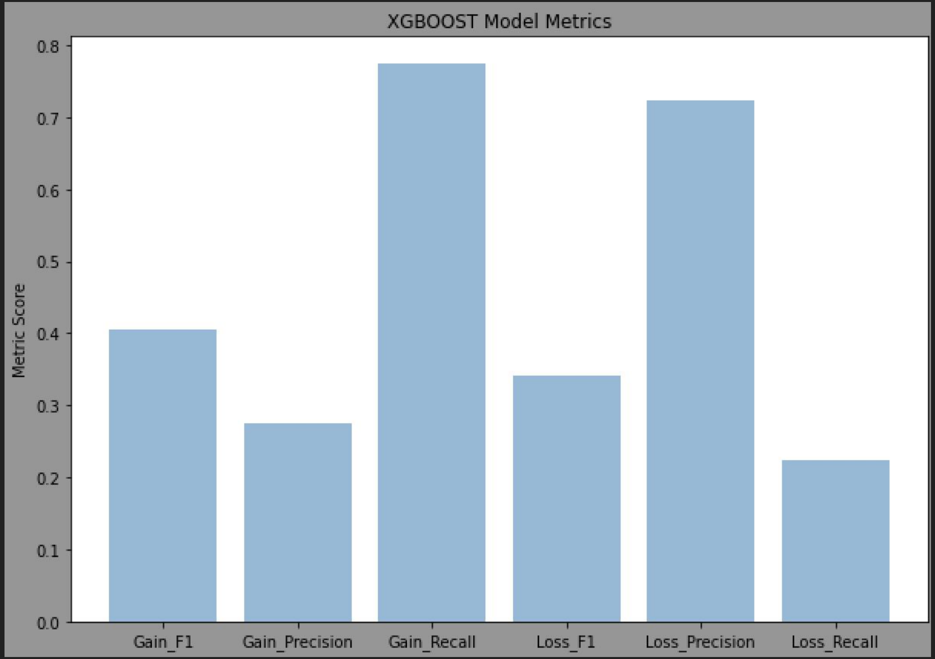
- I define gain/loss by change i.e. Stock Open Price - Stock Close Price
  - If change > 0 stock == gain
  - If change < 0 stock == loss
- Baseline Model (Vader Sentiment Scores)
  - Does not use machine learning
  - Predicts stock price using average score of title and text
- Baseline Model (XGBOOST with Vader Sentiment Scores)
  - Supervised binary classification machine learning
  - Predicts stock price using vader sentiment scores after being trained on actual gain/loss of a stock

Metrics (Baseline Models)

Vader Sentiment Model



Vader Sentiment XGBOOST Model

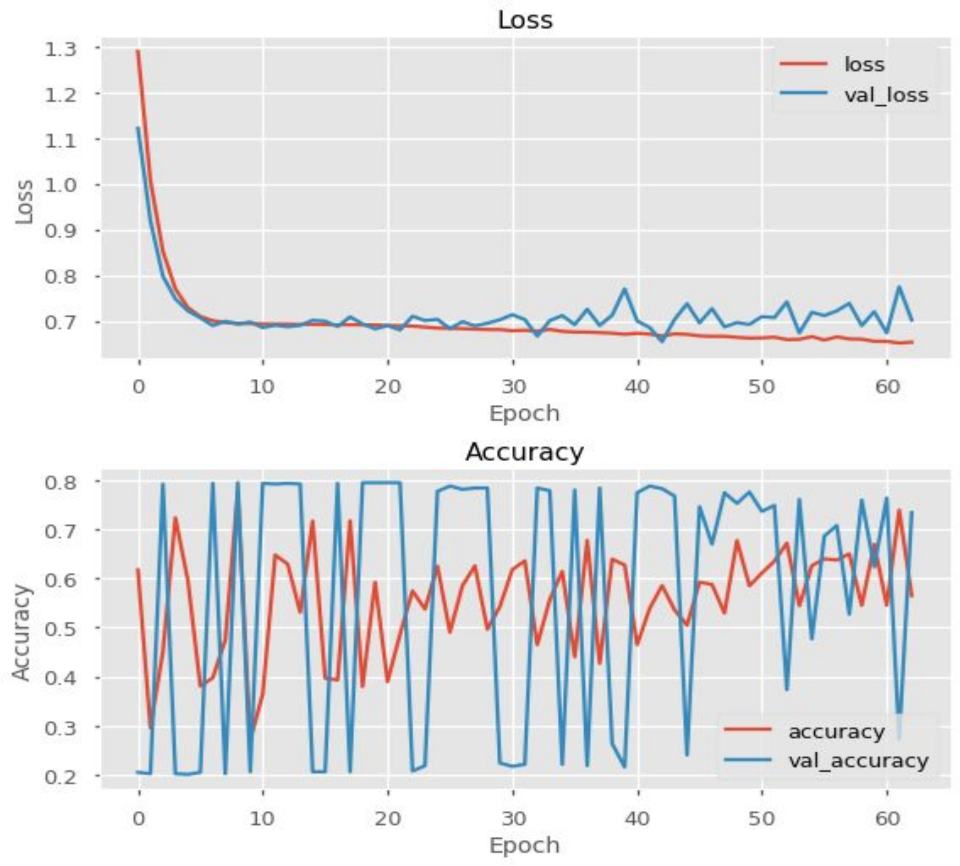


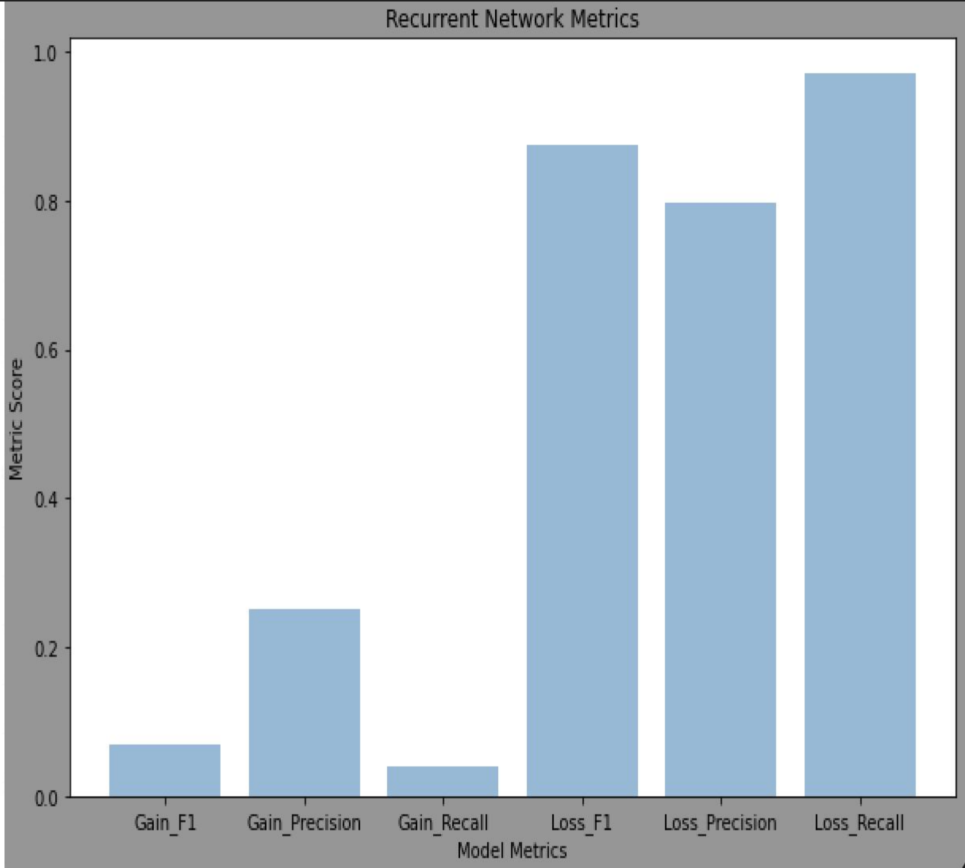
Methodology (Recurrent Neural Networks)

- GRU Model
  - Embedding layer with 50 vector size
  - GRU layer
    - 64 neurons
    - L2 regularizer
  - Dense Layer
    - 128 neurons
    - Tanh activation
  - Output Layer
    - Sigmoid activation
  - Loss: binary_crossentropy
  - Optimizer : Adam

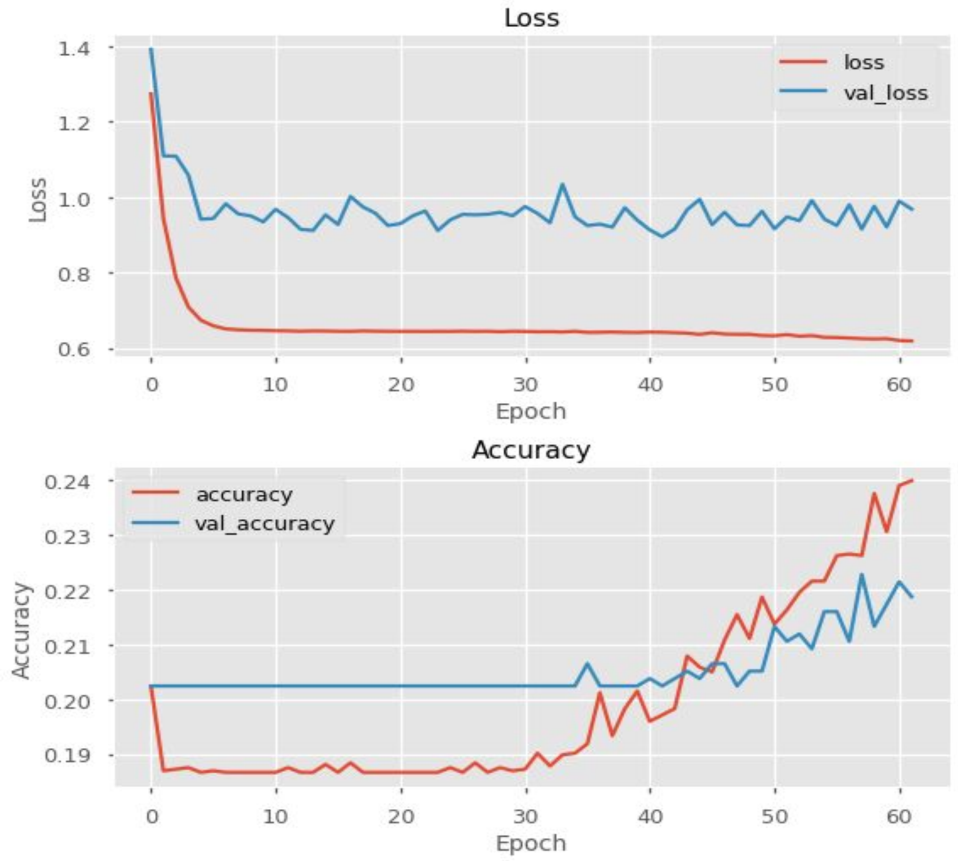
- LSTM Model
  - Embedding layer with 50 vector size
  - LSTM Layer
    - 64 neurons
    - L2 regularizer
  - Dropout Layer
    - Rate : .2
  - Dense Layer
    - 128 neurons
    - Tanh activation
  - Output Layer
    - Sigmoid Activation
  - Loss: binary_crossentropy
  - Optimizer : Adam

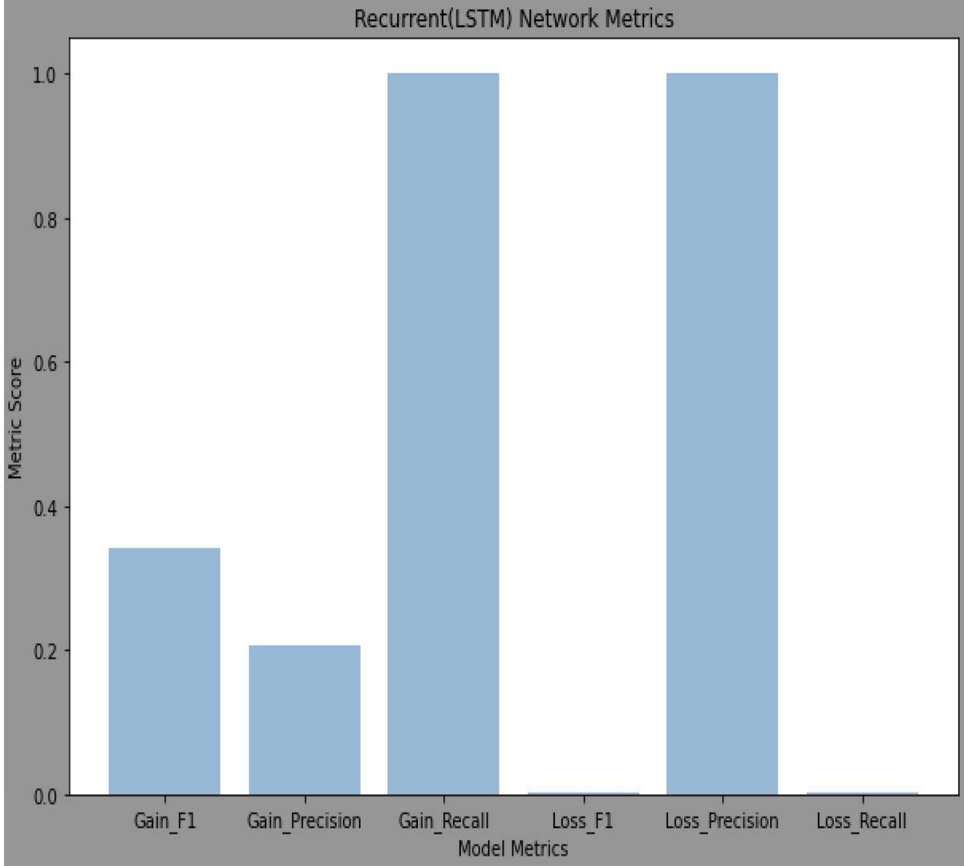
Metrics (GRU Recurrent Network)





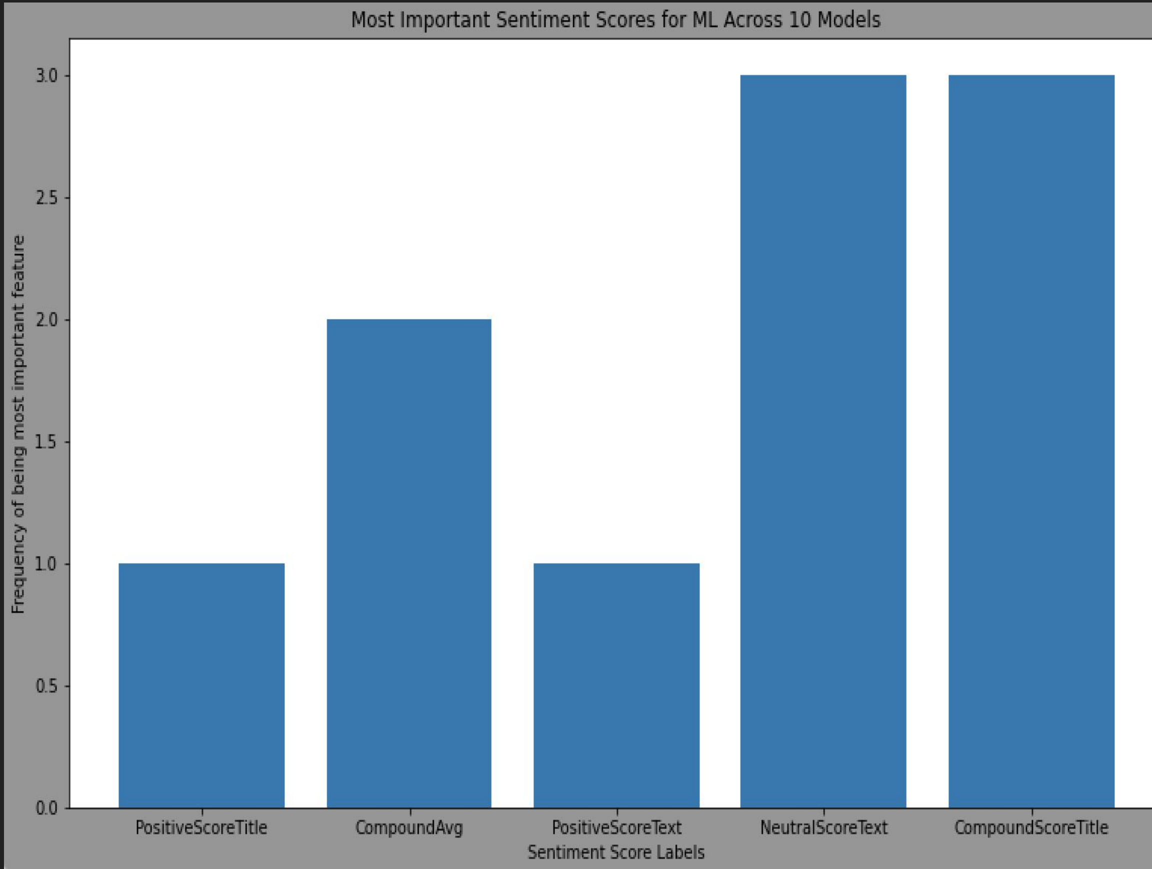
Metrics (LSTM Recurrent Network)



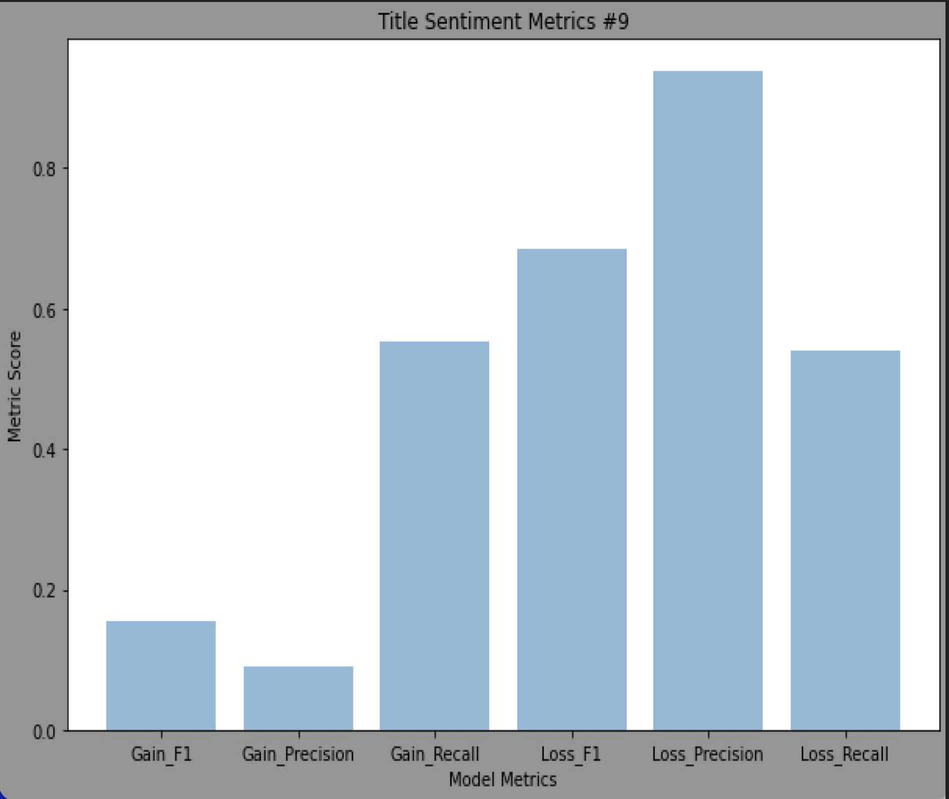


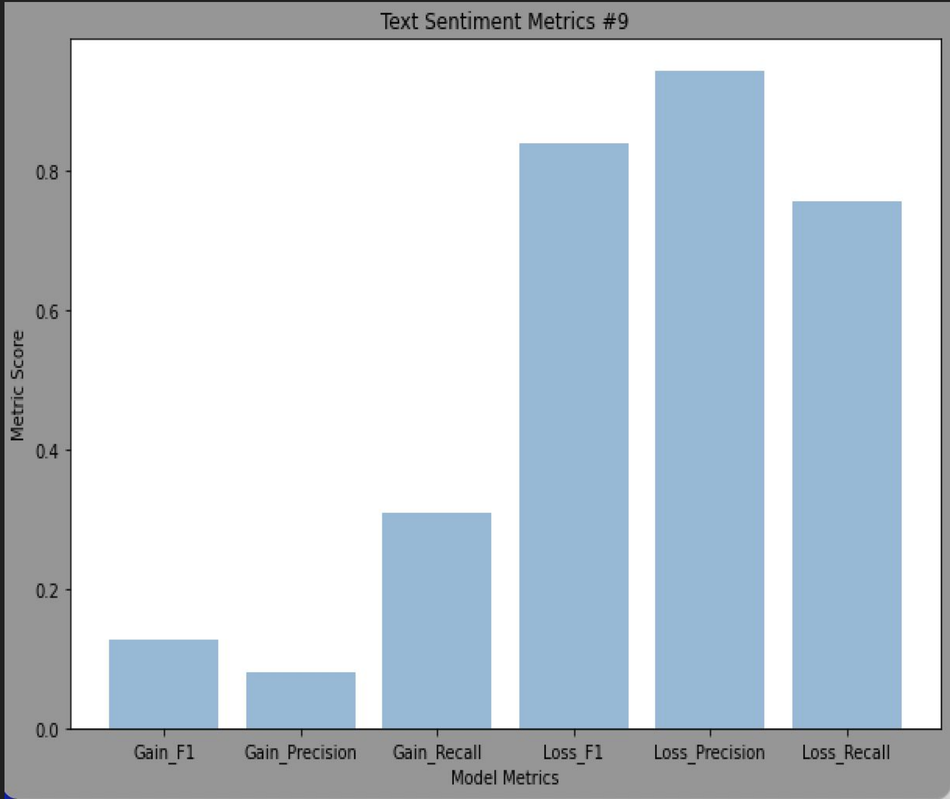
Results #1

- Text & Title are equally important when classifying stock news data



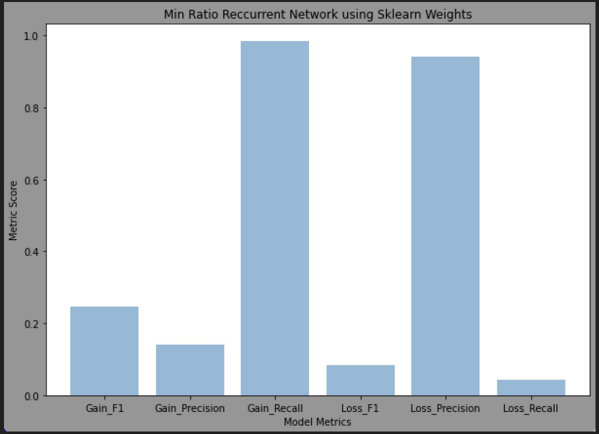
Results #1 Additional Evidence



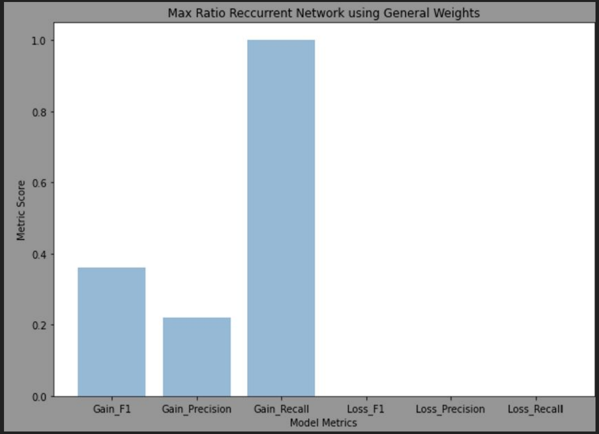


Results #2

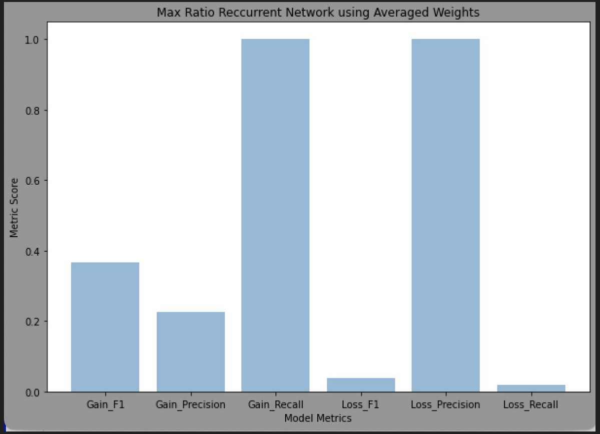
The best weighting system for larger ratios, loss/gain, is the averaged weights



Min Ratio Reccurrent Network using Sklearn Weights



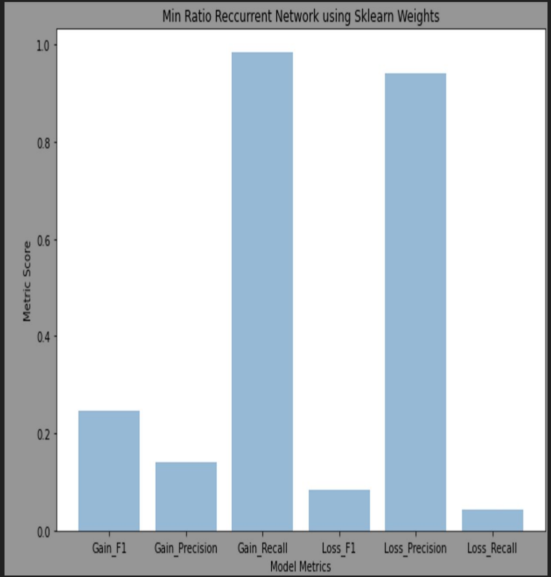
Max Ratio Reccurrent Network using General Weights

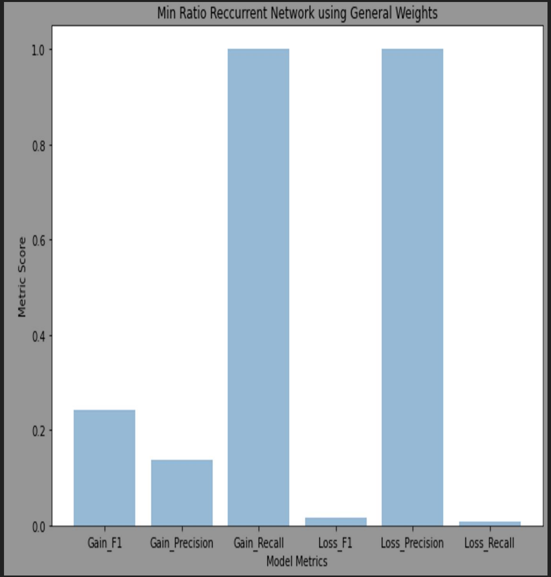


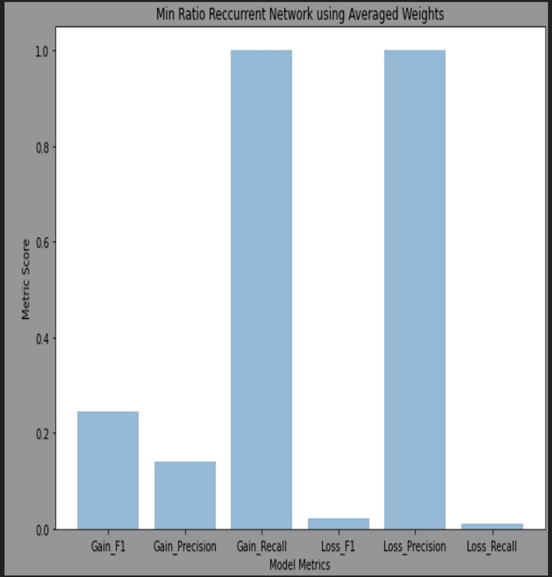
Max Ratio Reccurrent Network using Averaged Weights

Results #2

The best weighting system for smaller ratios, loss/gain, is the sklearn weights

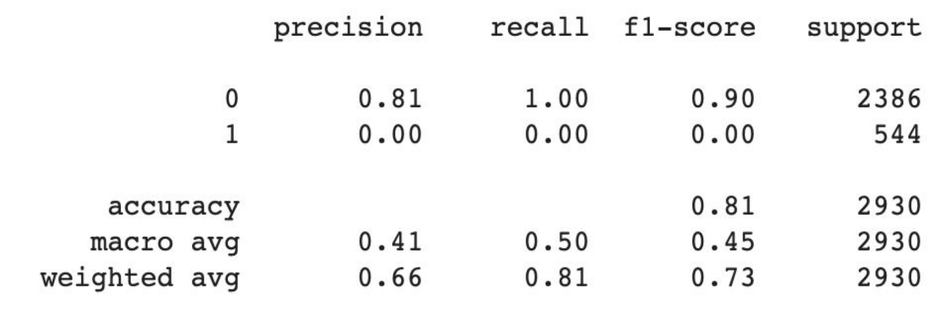






Results#3 Best Model

Recurrent Network GRU

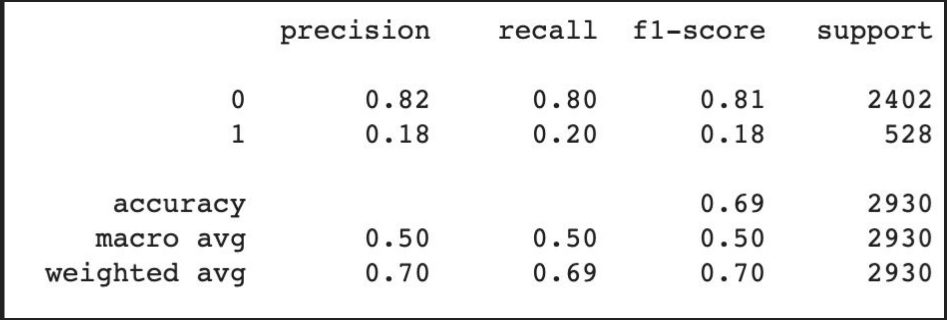


```
              precision    recall  f1-score   support

           0       0.81      1.00      0.90      2386
           1       0.00      0.00      0.00       544

    accuracy                           0.81      2930
   macro avg       0.41      0.50      0.45      2930
weighted avg       0.66      0.81      0.73      2930
```

Baseline
(XGBOOST)

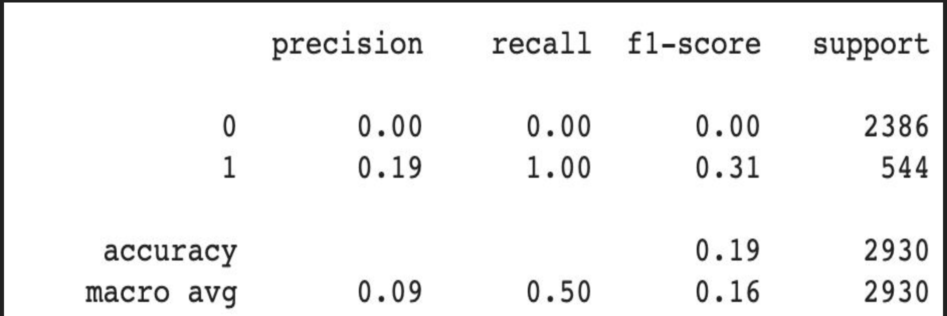


```
              precision    recall  f1-score   support

           0       0.82      0.80      0.81      2402
           1       0.18      0.20      0.18       528

    accuracy                           0.69      2930
   macro avg       0.50      0.50      0.50      2930
weighted avg       0.70      0.69      0.70      2930
```

Recurrent Network LSTM

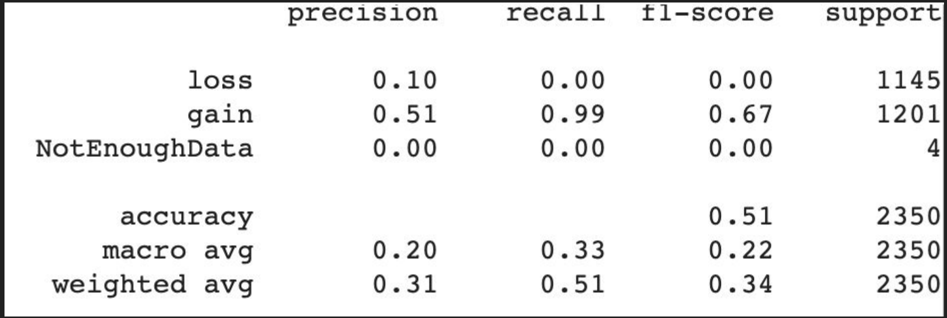


```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00      2386
           1       0.19      1.00      0.31       544

    accuracy                           0.19      2930
   macro avg       0.09      0.50      0.16      2930
```

Baseline (Sentiment | NON ML)



```
               precision    recall  f1-score   support

         loss       0.10      0.00      0.00      1145
         gain       0.51      0.99      0.67      1201
NotEnoughData       0.00      0.00      0.00         4

     accuracy                           0.51      2350
    macro avg       0.20      0.33      0.22      2350
 weighted avg       0.31      0.51      0.34      2350
```

Conclusions

- Develop & patent custom Natural Language Processing for stock news titles AND their corresponding text.
  - This will lead to higher return on investments,
  - business insights,
  - and allow your company to understand the sentiment of investors.
- To best handle the class imbalance where ratio = loss/gain:
  - Smaller Ratios (<8) : Use Sklearn generated weights
  - Larger Ratios (>8) : Use the average weight between sklearn and the general formula class/class
- Develop a sentiment analysis tool specifically for stock data. This will include labeling data for training, this should be done by human experts.

Further Work

Provided with more time, I would....

- Further develop transformer model
  - Parameter Optimization
  - Add weights
- Develop business product capable of analyzing stock sentiment to enhance company & customer return on investment
- Develop custom filtering methods capable of determining:
  - Optimal weights for class imbalance
  - Trending stocks (positive or negative)
- Develop process that collects equal gain & loss metrics so that models have higher performance

Questions?

Where to reach me:

Email : jvincentwelsh99@hotmail.com

LinkedIn:
https://www.linkedin.com/in/vincent-404/

GitHub: https://github.com/Eucalyptusss