Boom or Bust

Team 5:
Kedar Mohare
Alex Rwamashongye
Nick Nordby

Motivation & Summary

Our Goal:

- See if we could predict an equity price “boom” or “bust” (increase or decrease) from sentiment analysis
- To predict whether a stock price will go up or down, using one month sentiment analysis data, and utilize a ML model (classification) to get the job done

Questions Asked:

- Which companies and how many total for our analysis? (we ran into constraints here)
- Where is the sentiment analysis coming from and how robust?
- Do we want to focus on other forms of analysis (back testing, technical analysis, volatility)?
- What is feasible given our constraints (timing, data, expertise)?

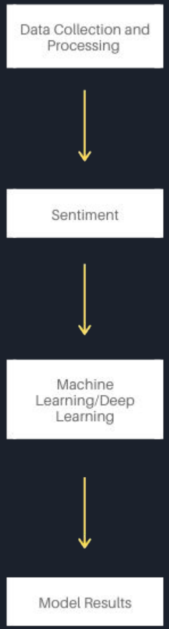
Our Starting Point:

- Get recent news articles and analyze the sentiment (vader sentiment analysis) and train models using recent price action as our “target” for whether a stock went up or down

Questions & Data

How do we achieve our goal?

- We used technologies learned so far in the class:
  - News API for the news articles on the company profiles
  - Natural Language Toolkit (nltk) for sentiment analysis
  - scikit-learn (sklearn) for classification and linear regression analysis
  - Tensorflow (tf) for the neural network model
- "Manual Data":
  - Fidelity for stock data, including indicators such as market cap, industry, short term sentiment, social sentiment.
- Issues?
  - Finding good data! (**had to abort our original project plan after two classes**)


Data Collection and Processing
Sentiment
Machine Learning/Deep Learning
Model Results

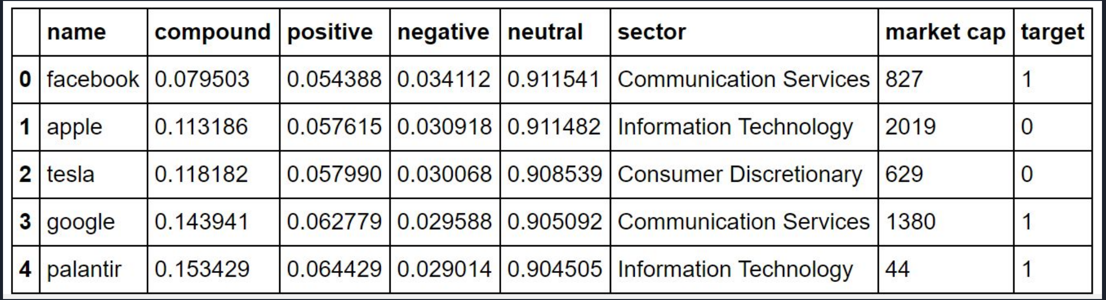
Model Training & Summary

Data Preparation - DataFrame:

- Gathered data into a dataframe, on which we applied several ML methods used in class
- For sentiment data, as mentioned, we used the Vader Sentiment Analysis
- For non-numerical data (sector) we utilized the the "get_dummies" function

Data Preparation - Model:

- Pre-Processing:
  - Our "X" values were sentiment, sector, market cap, etc.
  - "Y" value was our "target" column in the df; binary outcome "1" for increase, "0" for decrease
  - Train_Test_Split to create X_train, X_test, y_train, y_split
  - Standard scaler to scale the data
  - Run Model
  - Predict!



	name	compound	positive	negative	neutral	sector	market cap	target
0	facebook	0.079503	0.054388	0.034112	0.911541	Communication Services	827	1
1	apple	0.113186	0.057615	0.030918	0.911482	Information Technology	2019	0
2	tesla	0.118182	0.057990	0.030068	0.908539	Consumer Discretionary	629	0
3	google	0.143941	0.062779	0.029588	0.905092	Communication Services	1380	1
4	palantir	0.153429	0.064429	0.029014	0.904505	Information Technology	44	1

Models Utilized

Linear Regression Model:

- Produced the best results; "Market Forecasting" is a proper use of regression + classification
- **Accuracy: ~71%**
- Conclusion: Indicates that our model/thesis may have potential if improved with additional data

Easy Ensemble Classifier (Classification):

- Classification is not necessarily standard for a stock price prediction, but given examples in identity fraud ("default"/"not default") we thought this may be useful here ("stock up"/"stock down"); mitigate "loss" risk?
- Thought our data might make for a "weak learner" and so an easy ensemble would work well here
- **Accuracy: ~46%**
- Conclusion: Indicates our model/thesis is not strong

SMOTE Oversampling (Classification)

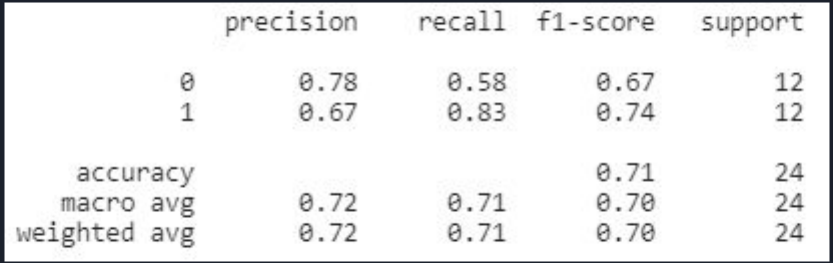
- We thought SMOTE may be appropriate as we didn't have as many "decreases" as "increases"
- **Accuracy: ~54%**
- Conclusion: Indicates our model/thesis is not strong

Neural Network:

- Less familiar with this model and likely not appropriate, but we wanted to try
- Inputs: Activation = Sigmoid; Loss = Binary Cross Entropy; Optimizer = Adam; Two Layers; 100 epochs
- **Accuracy ~54%; Loss ~76%** (similar to SMOTE)

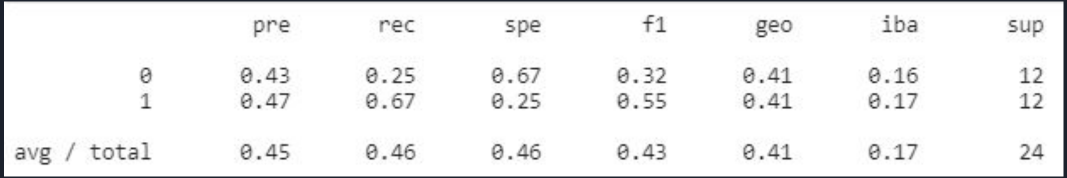
Model Outcomes

Linear Regression:



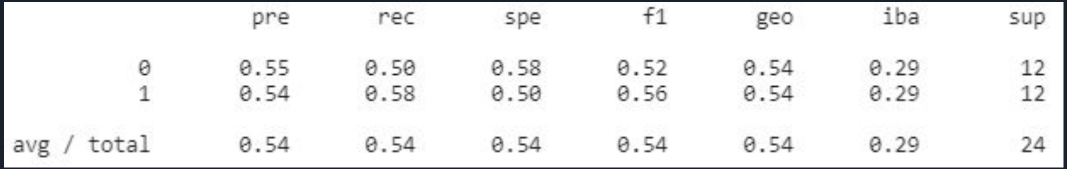
```
              precision    recall  f1-score   support

           0       0.78      0.58      0.67        12
           1       0.67      0.83      0.74        12

    accuracy                           0.71        24
   macro avg       0.72      0.71      0.70        24
weighted avg       0.72      0.71      0.70        24
```

Easy Ensemble:



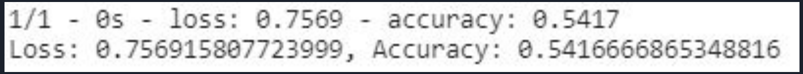
```
                   pre       rec       spe        f1       geo       iba       sup

          0       0.43      0.25      0.67      0.32      0.41      0.16        12
          1       0.47      0.67      0.25      0.55      0.41      0.17        12

avg / total       0.45      0.46      0.46      0.43      0.41      0.17        24
```

SMOTE Oversampling:

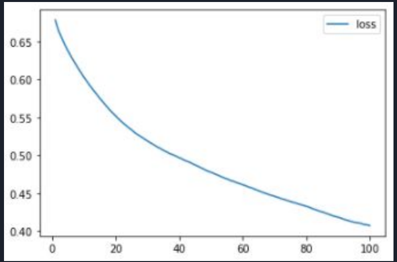


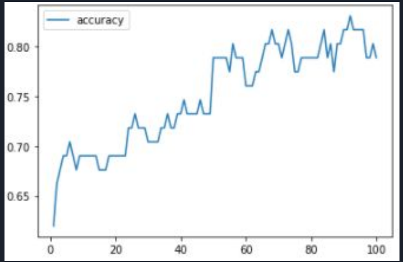
```
                   pre       rec       spe        f1       geo       iba       sup

          0       0.55      0.50      0.58      0.52      0.54      0.29        12
          1       0.54      0.58      0.50      0.56      0.54      0.29        12

avg / total       0.54      0.54      0.54      0.54      0.54      0.29        24
```

Neural Network:



```
1/1 - 0s - loss: 0.7569 - accuracy: 0.5417
Loss: 0.756915807723999, Accuracy: 0.5416666865348816
```



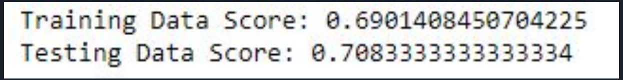


Linear Regression(Classification) Results

We did 3 different variations by adding features to the data.

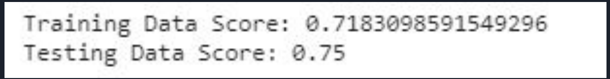
The data set was enhanced as follows:

1. The base data set included sentiment analysis scores, Industry sector and Market Cap.



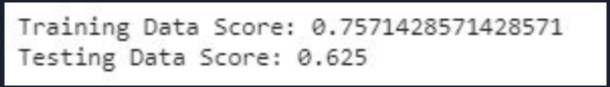
```
Training Data Score: 0.6901408450704225
Testing Data Score: 0.7083333333333334
```

2. In the second iteration, we added the short term sentiment(STS) data from Fidelity.



```
Training Data Score: 0.7183098591549296
Testing Data Score: 0.75
```

3. For the final iteration we added the ‘social sentiment’ indicator obtained from Fidelity



```
Training Data Score: 0.7571428571428571
Testing Data Score: 0.625
```

New Model?!

New Model Attempted:

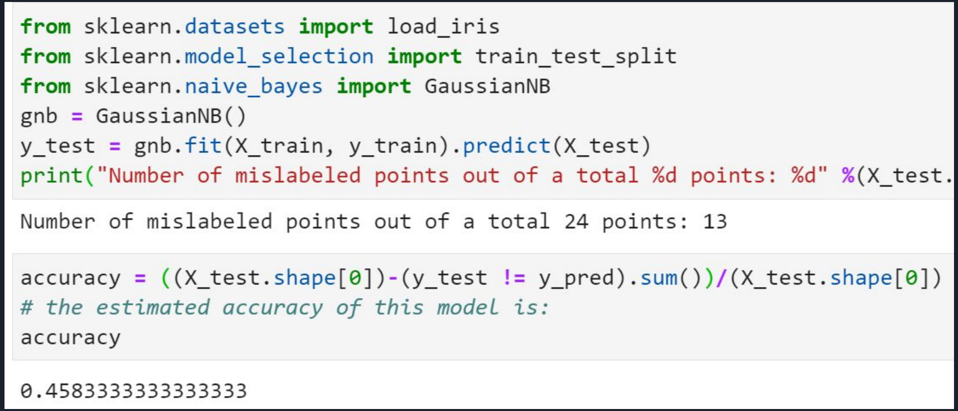
- Naive Bayes (Gaussian)

Why?

- Naive Bayes are supervised learning models that calculate the possibility of whether a data point belongs within a certain category or not (in this case “1” or “0”)
- Fit within the classification family of models we thought to be most effective
- Famously used in spam filtering
- Need limited data

Outcome?

- Not so great at **45.8% accuracy**



```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
y_test = gnb.fit(X_train, y_train).predict(X_test)
print("Number of mislabeled points out of a total %d points: %d" %(X_test.
```

```
Number of mislabeled points out of a total 24 points: 13
```

```
accuracy = ((X_test.shape[0])-(y_test != y_pred).sum())/(X_test.shape[0])
# the estimated accuracy of this model is:
accuracy
```

```
0.4583333333333333
```

Data Analysis & Discussing Results

Our Findings?

- Data is either hard to come by or expensive, even with democratization in tech, there are still no free lunches
- Different ML models yield different results, requiring a long deep analysis of each model and its parameters - linear regression worked the best
- **Existing data set & model is not yet ready for production, but it is an interesting consideration! Could be added as an appendage to our Project 1**

While successful…

- Our data and models should be tested over a longer period to build a more complete model; we need to build out a more robust dataset
- Deeper dive into news sources; the number of articles utilized for each analysis; different kinds of sentimentment analyses; include more unique indicators

Postmortem

Boom or Bust?

- Good data is expensive to get: we used News API to pull news articles but were limited by the amount of requests on the free tier. Solution: save the query into a .csv file to limit API calls & pull new data over several days and weeks.
- Scope: started with an ambitious goal of predicting company bankruptcies. Changed project scope to be workable.
- Model: predict company stock price move using sentiment (classification ML models)
- Next Time: use other methods of data collection (e.g. web scraping, reading SEC filings), larger predictive modeling (e.g. time series analysis based on rolling sentiment)

Major Issues & how we dealt?

- The data we wanted (historical sentiment)
- Having to change our project entirely
- Found a project that fit within our constraints

Q&A:

Thank You!

Kedar, Alex, & Nick