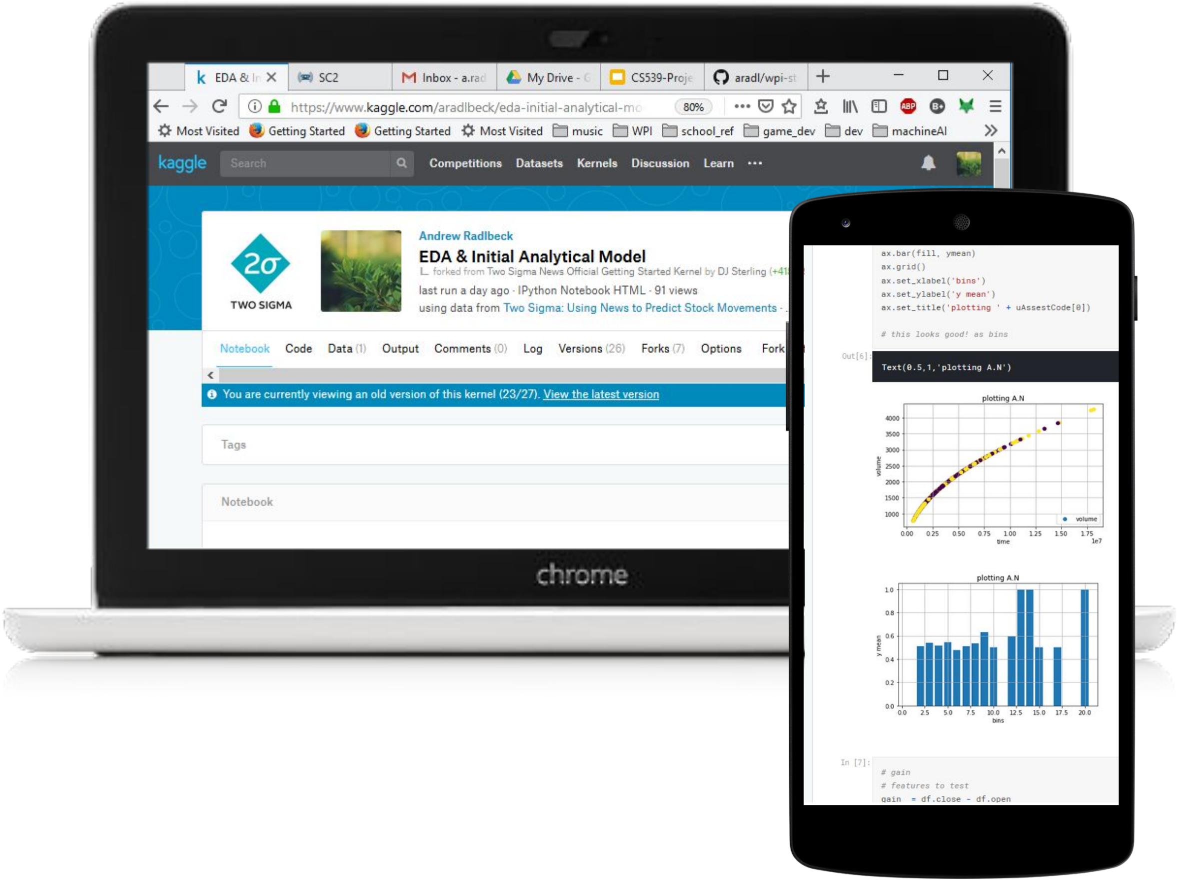
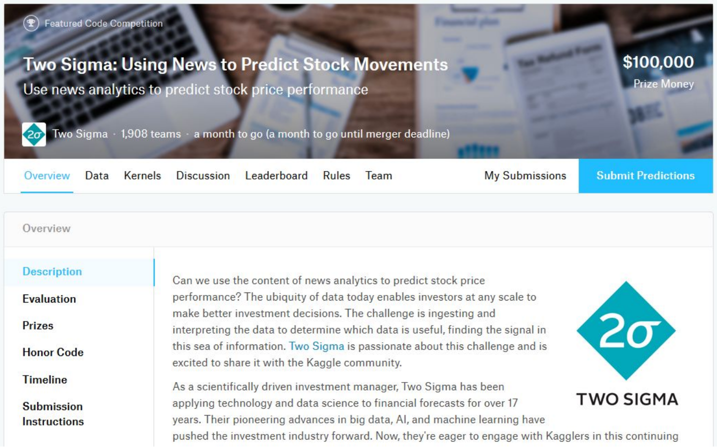
Making Money from the News

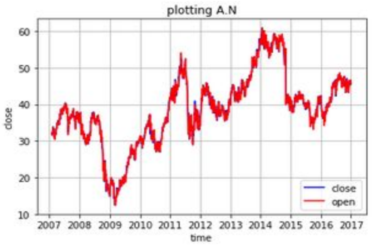
...and learning a lot about machine learning in the process

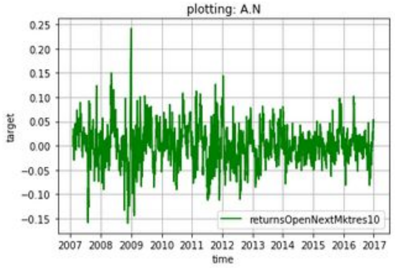


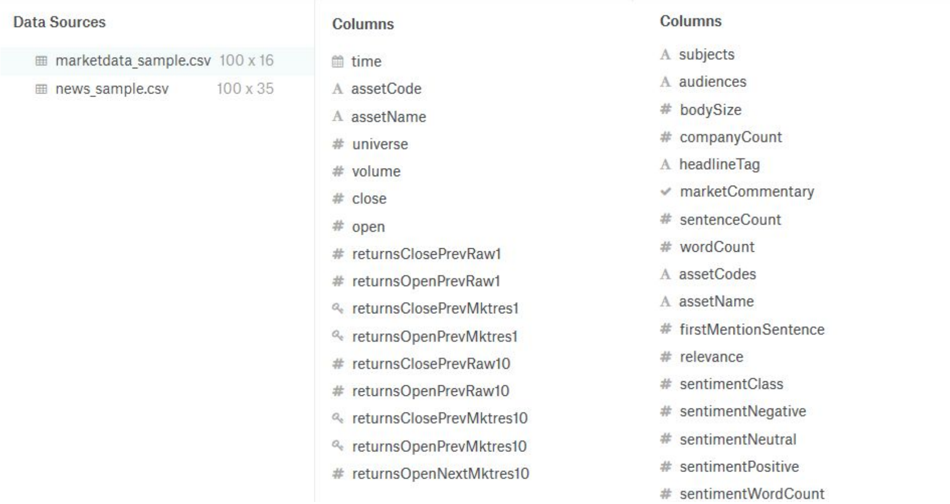
Code hosted on GitHub

https://aradl.github.io/wpi-stock-project/









Target Prediction

The goal of this project is to combine news information and stock market data to predict performance ten days later

Project Dataset

The data set is comprised of market and news information

Scope Reduction & Scoring

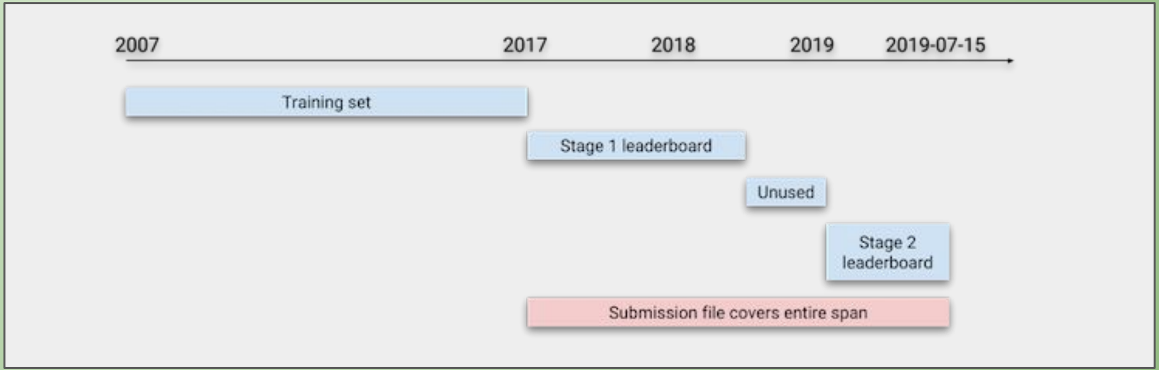
For this competition we chose to reduce the scope to allow us to explore a wider range of algorithms and ensemble methods. This also helped with the added overhead of learning to deal with large datasets.

- We focused on standard confusion metrics and include competition score as a bonus
- Looked at features one day at a time and assumed independence across days
- Prediction is a confidence value from -1 to 1, 1 being that the stock will do better in 10 days

$$x_t = \sum_i \hat{y}_{ti} r_{ti} u_{ti} \qquad \text{score} = \frac{\bar{x}_t}{\sigma(x_t)}$$

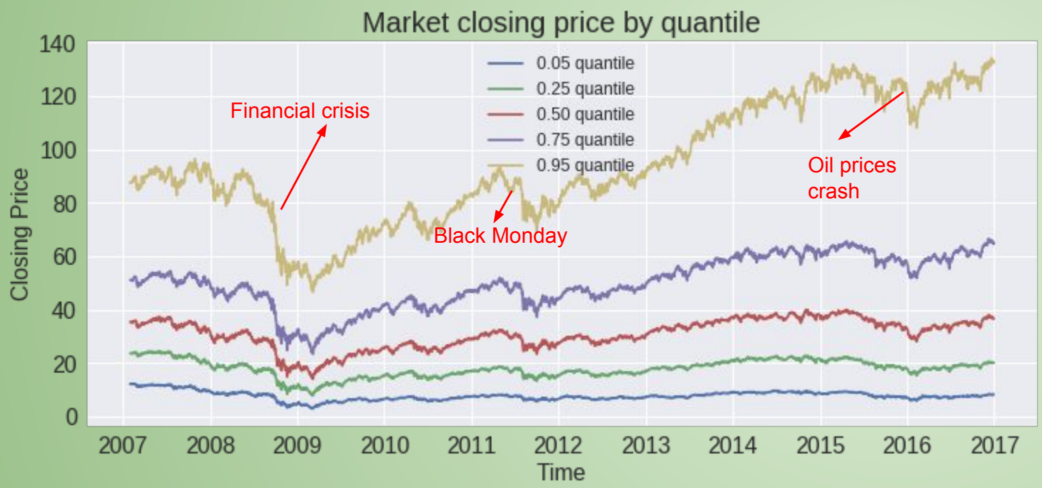
Dataset Description

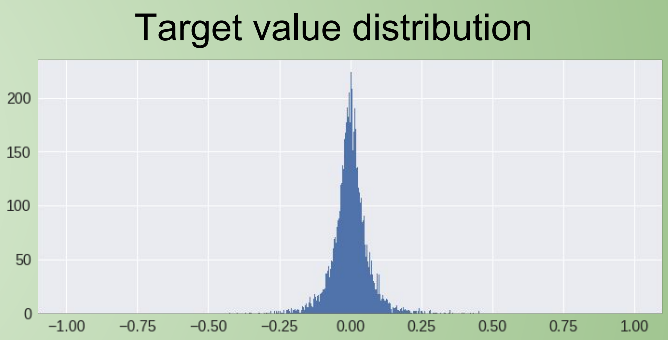
- Market data
  - 4 million samples with 16 features
  - Interesting features: close, open, volume, **returnsOpenNextMktres10**
- News data
  - 9 million news data with 35 features
  - Interesting features: sentiment, relevance …
- Shared features
  - assetCode
  - assetName
  - Date



EDA - Exploratory Data Analysis

Market

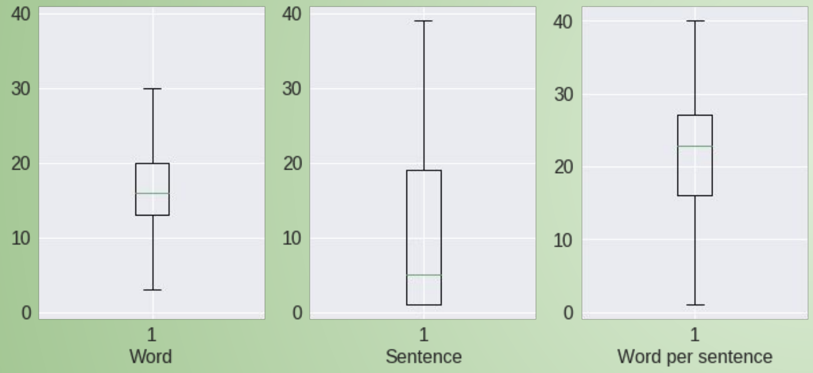


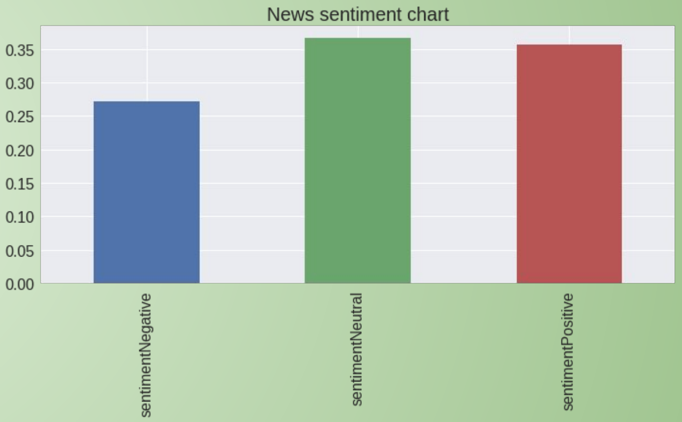


Interesting phenomenon can be seen analyzing the data relative to known historical events.
The ground truth target values has a normal distribution.

EDA - Exploratory Data Analysis

News





The news data appears to have a normal distribution of sentiment.

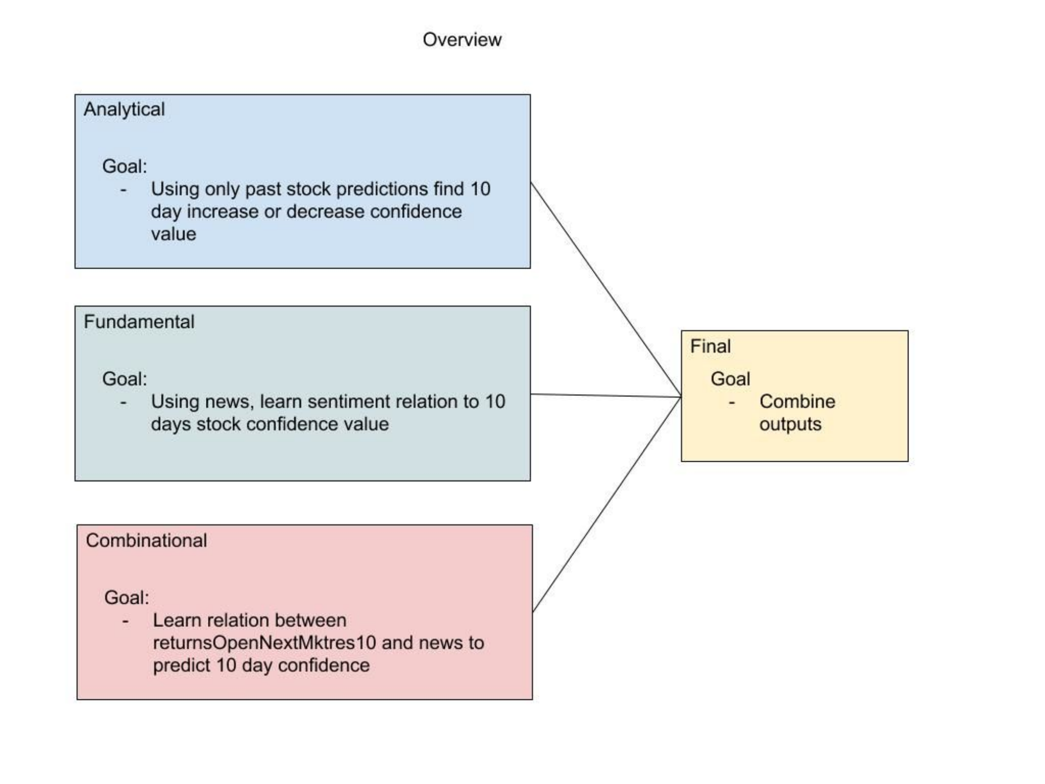
Project Architecture

The project consists of three parts:

1. Analytical model - looks only at market data
2. News model - looks only at news data
3. Combinational model - looks at the relation between news and market

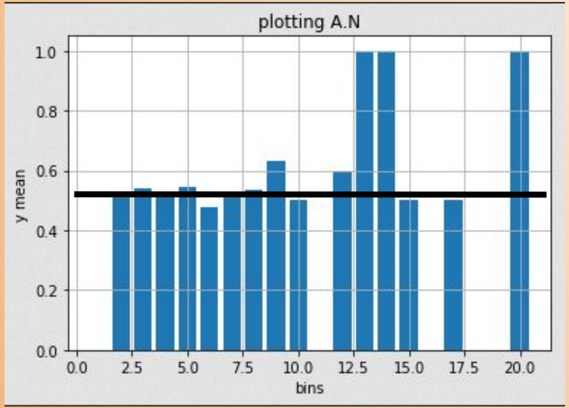
These three blocks are then combined using an ensemble method to determine who is the best at predicting the target value given the features

Project Architecture

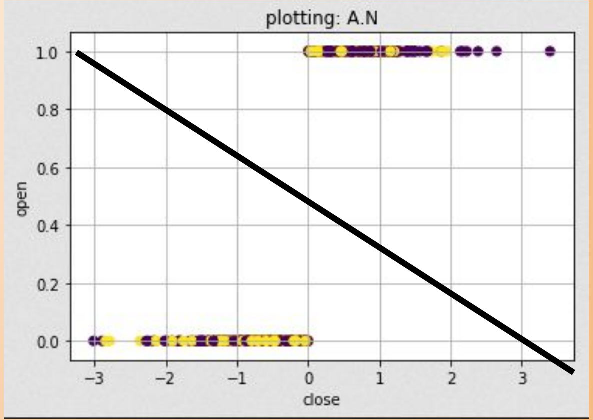


Analytical Block Features

- The analytical block looks only at the market data.
- Features chosen to create a simple **Linear** relation between the current day and the target value
  - gainb - a bin separated daily gain value
  - gain - the daily gain
  - volumeb - a bin separated stock volume



volumeb



gain vs gainb

Analytical Block Algorithm

- The algorithm chosen is a linear SVM classifier provided by scikit learn [LinearSVC]
- We have also tried out other algorithms, but this was much faster with roughly the same performance. Below are a few examples:

LinearSVC					
	A (-1)	A (1)		Accuracy	0.49
P(-1)	37	205		Percision	0.15
P(1)	47	210		Recall	0.44

SVR(C=0.7, kernel='rbf')					
	A (-1)	A (1)		Accuracy	0.56
P(-1)	85	140		Percision	0.38
P(1)	82	193		Recall	0.51

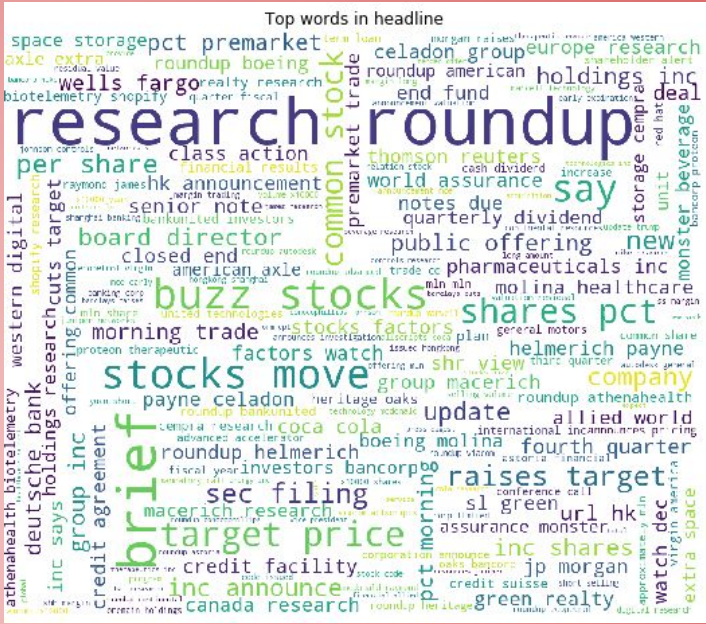
svm.SVC(C=0.7)					
	A (-1)	A (1)		Accuracy	0.55
P(-1)	55	185		Percision	0.23
P(1)	40	220		Recall	0.58

- Somewhat expected from stand alone daily data as stock market movement is often modeled as a random process
    - For better analytical results temporal features would work better
    - Attempt to model stock ‘patterns’

Random: 0.0708

This: 0.13416

News Block Preparation



- **Data Usage**

  Usage of only news data

- **Motivation**

  To understand the predictability explicitly w.r.t News

- **Feature reduction and engineering**

  Unstacked assetCode from each news.

  Cleaned entries with “no news”, i.e, no headline or wordCount, sentenceCount and zero bodySize.

  Added Position of first mention, Coverage of sentiment words

Meantime News

```
Top mentioned companies in the news are:


Barclays PLC             64350
Citigroup Inc            63689
Apple Inc                62783
JPMorgan Chase & Co      60635
Bank of America Corp     57560
Name: assetName, dtype: int64
```

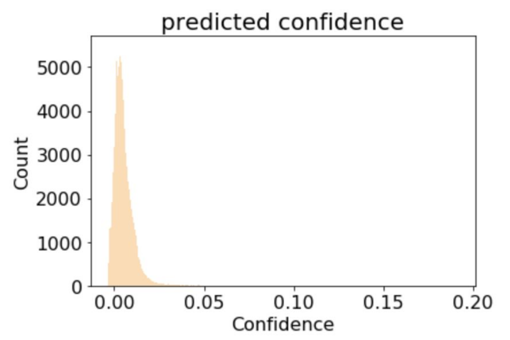
```
Top mentioned companies for positive sentiment are:
Barclays PLC                22855
Apple Inc                   22770
General Electric Co         20055
Royal Dutch Shell PLC       18206
Citigroup Inc               18025
Name: assetName, dtype: int64
```

```
Top mentioned companies for negative sentiment are:
Citigroup Inc                  30823
JPMorgan Chase & Co            29129
Bank of America Corp           28197
Apple Inc                      26702
Goldman Sachs Group Inc        25044
Name: assetName, dtype: int64
```

News Block Algorithm

- **LogisticRegression** is a linear model that is quick to train and quick to predict
  - We again sticked with something simple to map the news inputs to outputs looking to discover a relation between some of the features and the output
  - We use solver='Stochastic Average Gradient' method for optimization because it's fast for large dataset.
- Results for this stand alone model look like:
  - Might be slightly better than random…
  - Skewed distribution of confidence value.

```
accuracy : 0.511158
recall_score : 0.511459
precision_score : 0.978723
f1_score : 0.671833
```



Combinational Block Algorithm

lightGBM is a gradient boosting framework based on decision tree algorithms.

- Faster training speed and higher efficiency

- Lower memory usage

- Better accuracy

- Capable of handling large-scale data

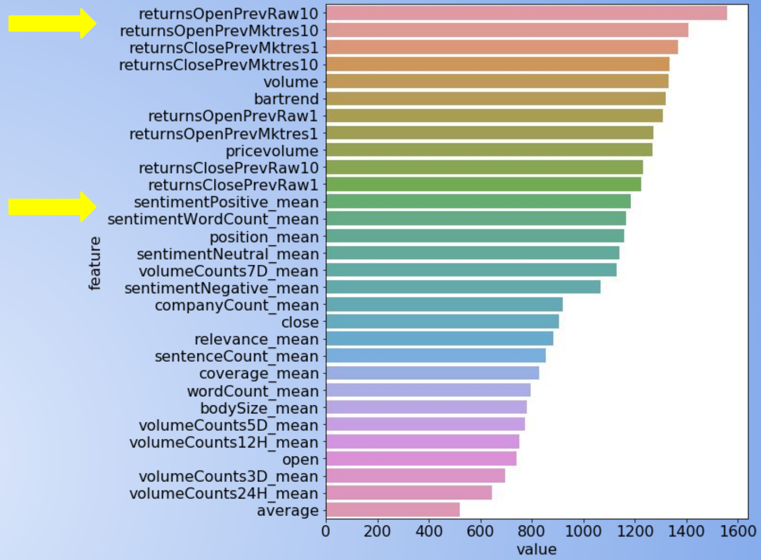
Combinational Block Features

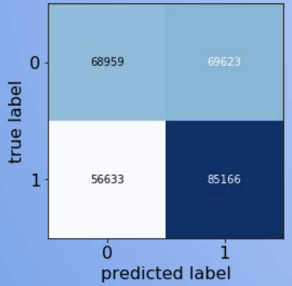
- Market data preparation.
  - Bartrend (close/open); average; pricevolume (volume*close)
- News data preparation.
  - Position (firstMentionSentence/sentenceCount); coverage (sentimentWordCount/wordCount)
  - Group by time and assetCode
- Merge on time and assetCode.
- Drop bottom features based on feature-importance ranking.

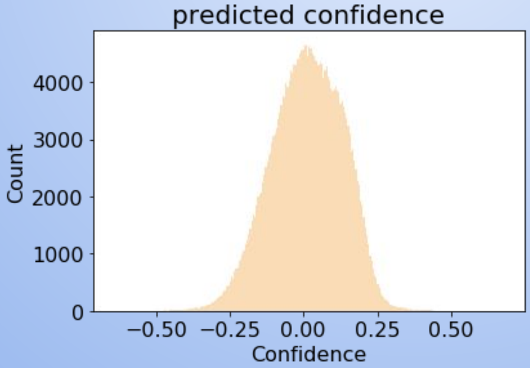
LightGBM

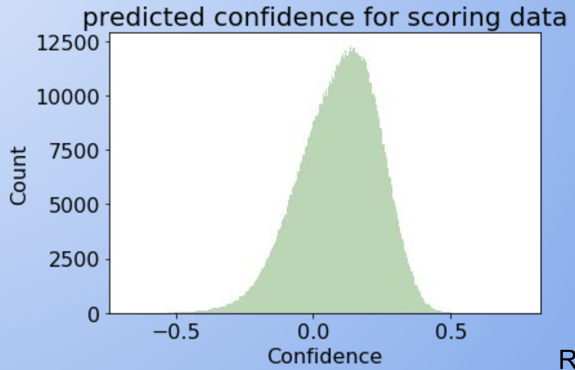
- 30 features
- Dataset: Training, validation, testing.
- Parameter tuning based on log loss.
- Error metrics on test dataset:

```
lgb accuracy : 0.549698
lgb AUC : 0.569611
```









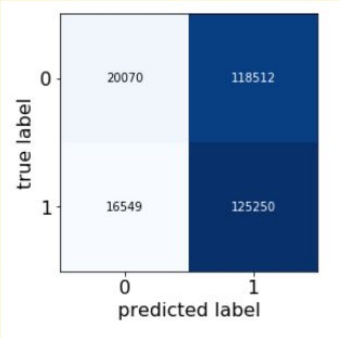
Random: 0.0708

This: 0.64084

Ensemble Method - I: Simple Voting

- To start out we first used a simple majority rules voting method calculated using numpy array math

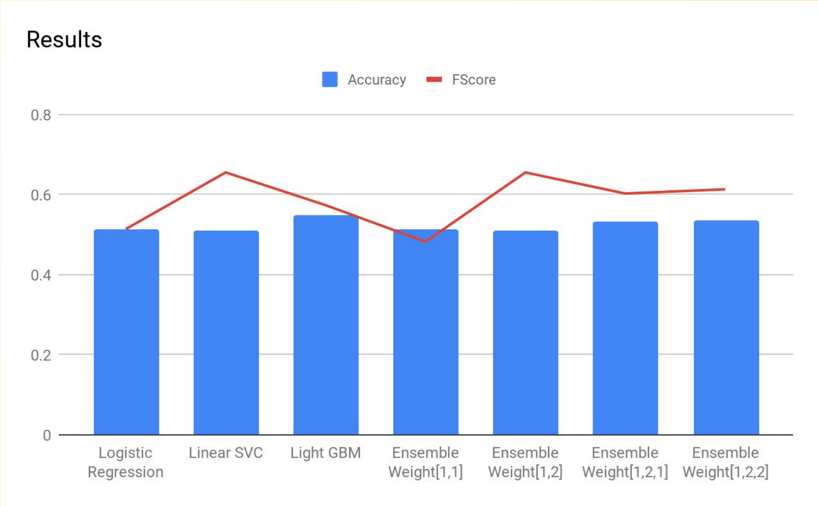
```
total accuracy : 0.518295
```



- Vote weighting is News, Market = 2 / 7 and Combinational = 3 / 7
- It worked ok, I think we will see the best gains with a more sophisticated method that takes feature importance into account

Ensemble Method - II: Ensembles of Classifiers that Operate on Different Feature Subsets

- Classifiers used- Logistic Regression, Linear SVC, LightGBM
- Classifiers prefitted on the subset of data as they were trained previously
- Merged all the features into a dataframe
- Ensembled with the soft voting



Results

Summary of individual models:

Analytical:

LinearSVC					
	A (-1)	A (1)		Accuracy	0.49
P(-1)	37	205		Percision	0.15
P(1)	47	210		Recall	0.44

0.13416

News:

```
LR clf accuracy : 0.507153
LR clf AUC : 0.502644
```

Combinational:

```
lgb accuracy : 0.549698
lgb AUC : 0.569611
```

0.64084

As you can see both factors are needed to do a good job at predicting future prices (despite similar looking accuracy and AUC)

Random: 0.0708

Results

Ensemble Method - I:

```
total accuracy : 0.518295
```

Ensemble Method - II:

```
accuracy Logistic Regression: 0.5121909899467243
f score Logistic Regression:0.5141512366235957

accuracy SVC Linear: 0.5095472682285281
f score SVC Linear:0.6558187409856991

accuracy LGBM: 0.548240119480172
f score LGBM:0.5710015833636741

LR SVC Ensemble Weight[1,1]
accuracy ECLF: 0.5128062236686655
f score ECLF:0.4826296503205159

LR SVC Ensemble Weight[1,2]
accuracy ECLF: 0.5095472682285281
f score ECLF:0.6558187409856991

LR SVC LGBM Weight[1,2,1]
accuracy ECLF: 0.5335681326764896
f score ECLF:0.6030406623134683

LR SVC LGBM Weight[1,2,2]
accuracy ECLF: 0.5359666525489847
f score ECLF:0.6137059511217503
```

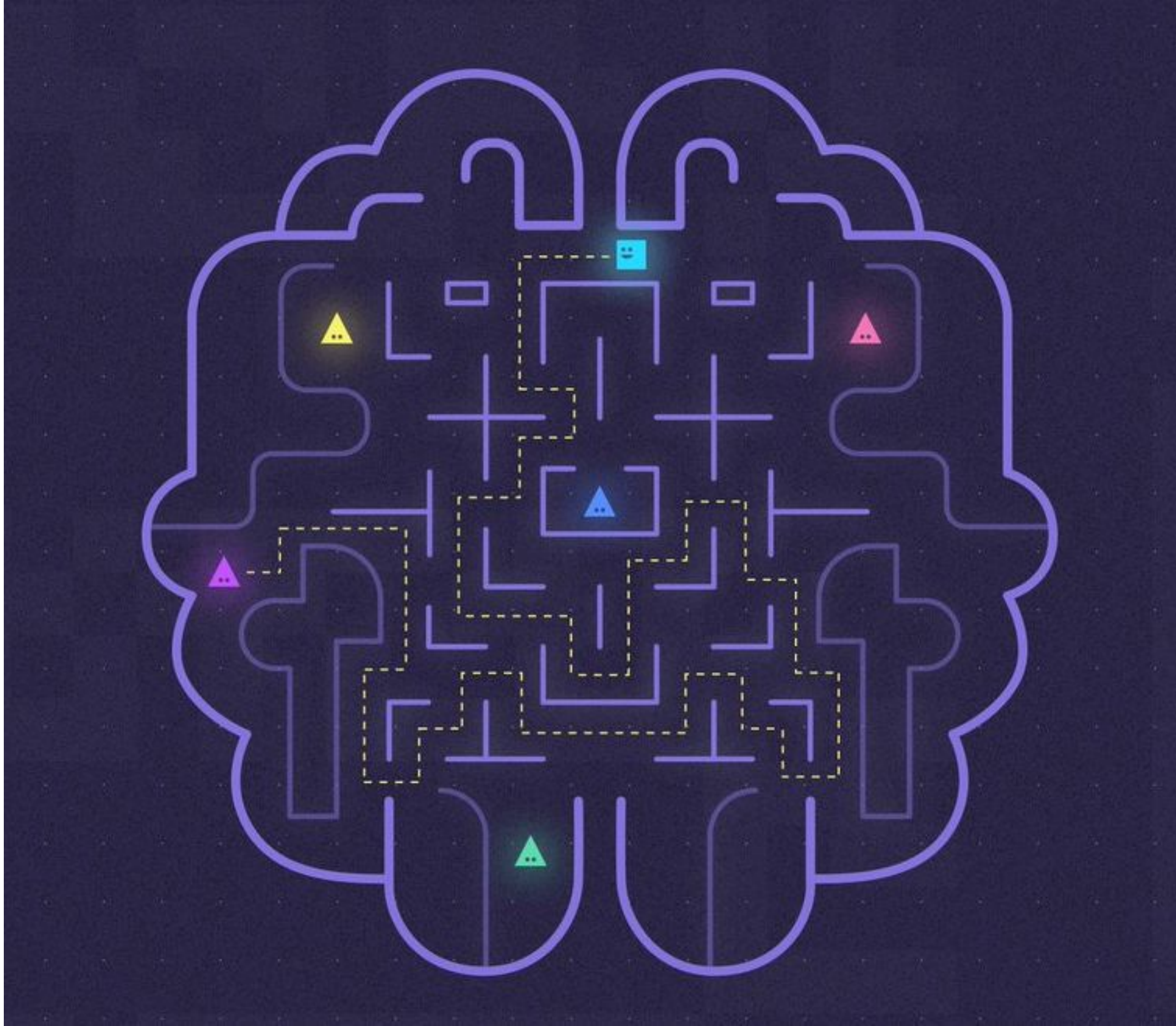
It looks like nothing extra can be learned from combining models that don’t take all the features into account

Lessons Learned

- PREDICTING THE MARKET IS HARD...
- Large dataset (easy to overfit and average out to random guessing)
- Kaggle kernel limitation (CPU, RAM, time restrictions)
- For future prediction more focus should be put on temporal features that help understand the past and we should no assume independence across days
- Unable to try some fancier algorithms

Team

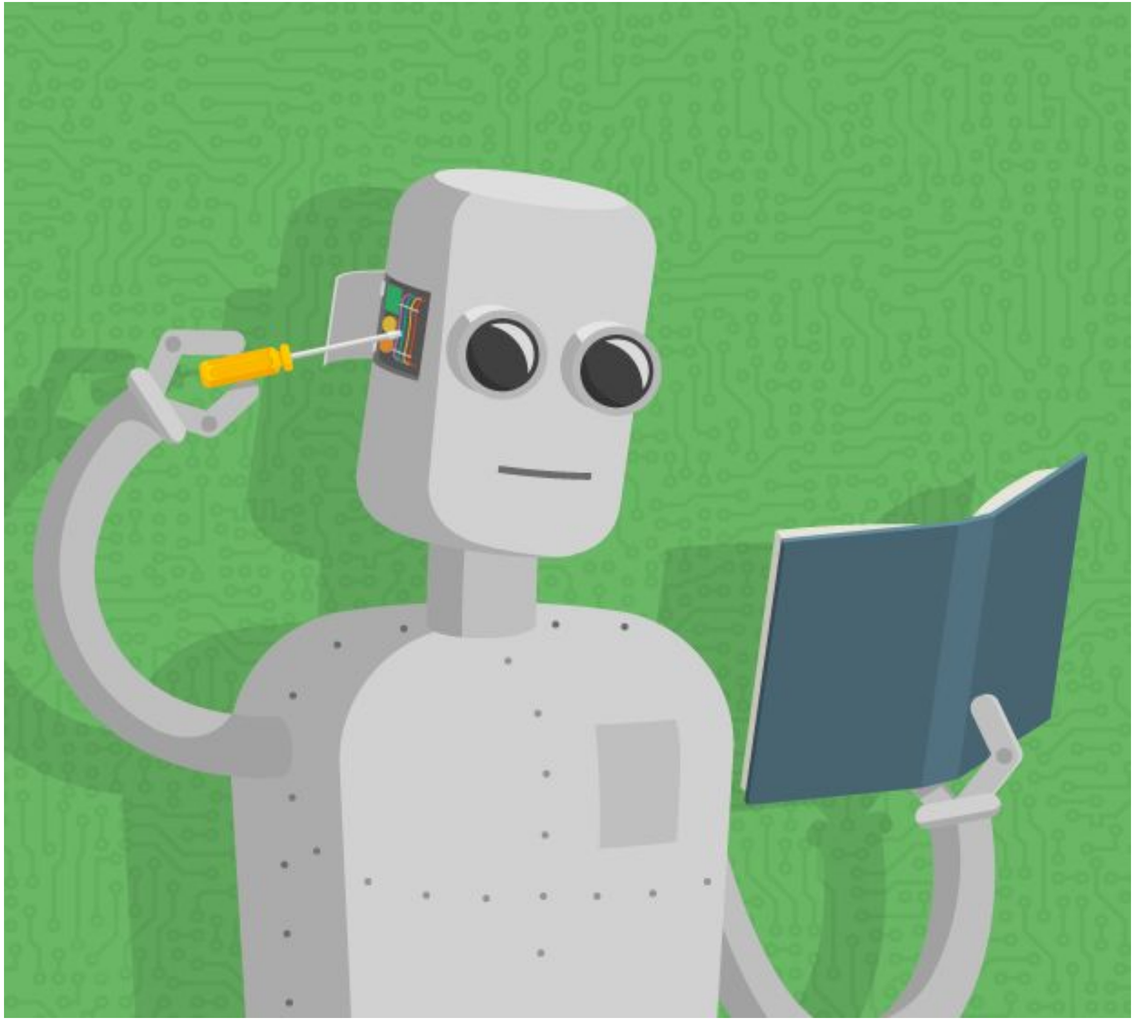
Andrew Radlbeck
Yang Fu
Ankit Gupta
Meghana Kasal



Time For a Demo?

https://github.com/aradl/wpi-stock-project/blob/master/src/eda.ipynb

https://www.kaggle.com/aradlbeck/vote-lrnews-lgball-anasvc



Questions?