# Predicting stock prices after quarterly reports

Joshua Hartmann - jchart@uvic.ca
Diego Aquino Chavez - diegoaquinochavez@uvic.ca
Rishabh Sandil - rishabhsandil@gmail.com

SENG474 Data Mining

# Table of contents

# Introduction

# Quarterly reports

- Stock prices often follow the butterfly effect. Are hard to model over long time scales
- Quarterly reports contain information that can have influence on stock prices
- Goals:
  - Produce a database with at least 10,000 samples
  - Predict whether the price will increase or decrease with 70% accuracy
  - Choose 3 classifiers to compare and contrast
- **Given quarterly report data along with other data attempt to predict short-term changes in stock price following their release**

# Summary

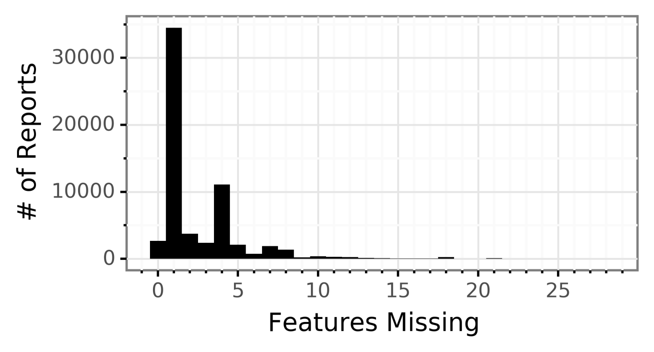
- Data is collected
- Data is scaled, transformed, and normalized
- Classifiers Selected
- Classifiers Fit
- Results assessed
- Solutions to problems explored!
- Conclusion

# Data Collection

# Data Collection - Quarterly Reports

- **Source:** StockPup [2]
- 65,000 quarterly reports
- Up to 46 pre-cleaned figures from each report
- Revenue, liabilities, net margin, asset turnover, debt...
- Lots of Data Missing

# Data Collection - Quarterly Reports

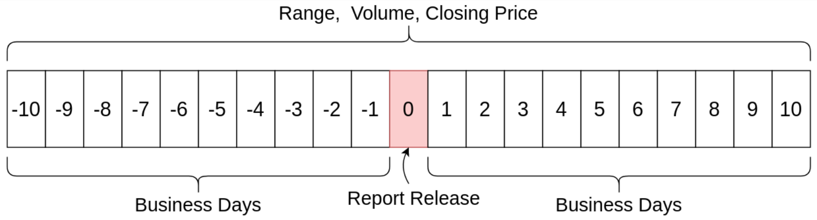


# Data Collection - Composite Indices

- **Source:** Yahoo Finance [3]
- DOW, NASDAQ, TSX, S&P 500
- All highly correlated
- Volume and closing price

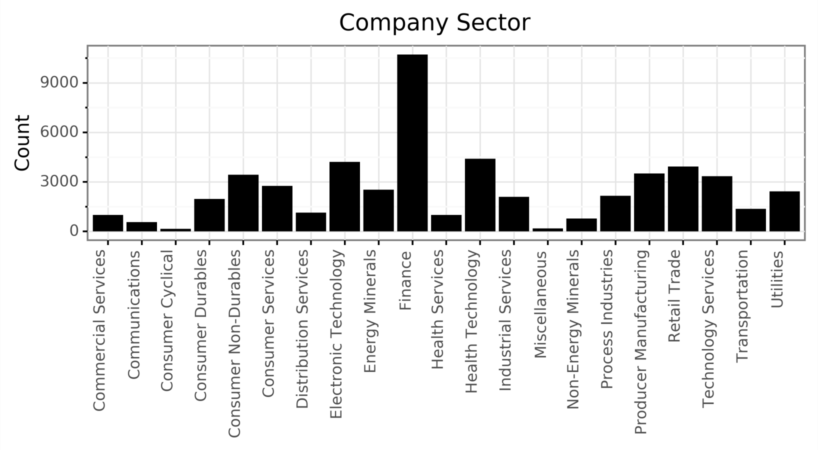
# Data Collection - Stock Price

- **Source:** IEX Finance [1]
- Stock closing price, range and volume data
- 10 business days prior and post quarterly report
- Some missing stocks
- Accounted for changes in stock symbols



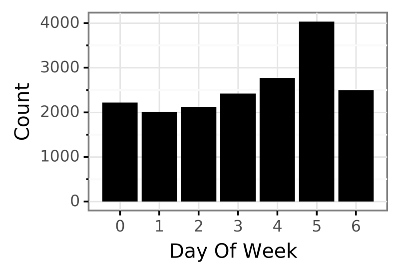
# Data Collection - Company Information

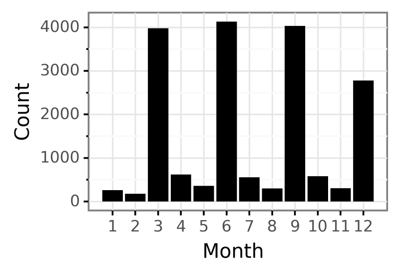
- **Source:** IEX Finance [1]
- Country, exchange, sector



# Data Collection - Additional Data

- Day of week (0 = Monday)
- Month



# Data Cleaning and Normalization

# Data Cleaning and Normalization - Quarterly Reports

- Change relative to previous report
- Trims the first report from every company
- Puts all companies on the same scale
- A few crazy outliers manually removed

$$report_{i_{cleaned}} = \frac{report_i - report_{i-1}}{report_{i-1}}$$

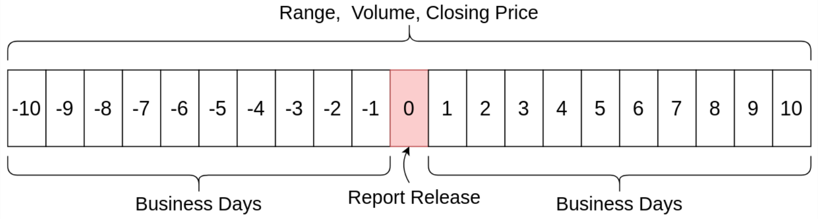
# Data Cleaning and Normalization - Composite Indices

- Change relative to the price/volume at the time of the previous report

# Data Cleaning and Normalization - Stock Price

- Change relative to stock price 1 day before report

$$day_{i_{cleaned}} = \frac{day_i - day_{-1}}{day_{-1}}$$



# Data Cleaning and Normalization - Company Info

- One-hot encoded
- Adds many dimensions to input

$$Sector \in \{Communications, Finance, Utilities, ...\}$$
$$\downarrow$$
$$Sector = [0, 1, 0, 0, 0, 0, 0, 0, ...]$$

# Model Selection

# Model Selection - Random Forest

- Simplicity and Flexability
- Impossible to overfit
- Resilient to outliers
- Less dependant on hyperparameters

| Parameter | Range |
|---|---|
| Number of Trees | 1000 |
| Criterion | gini, entropy |
| Min Samples Split | 0.01%, 0.1%, 1%, 10% |

**Table 1:** Range of parameters to test for Random Forest model fit

# Model Selection - MLP

- Theoretically able to model arbitrary complexity
- Likely complex relationships in data

| Parameter | Range |
| --- | --- |
| Hidden Layers | 1, 2, 3 |
| Activation Function | Relu, logistic |
| Solver | lbfgs, adam |
| Alpha | $1 \times 10^{\{-1,-2,-3,-4,-5\}}$ |

**Table 2:** Range of parameters to test for MLP model fit

# Model Selection - Gaussian Naive Bayes

- Easy to interpret
- Most features are likely gaussian
- Contrast the other two models

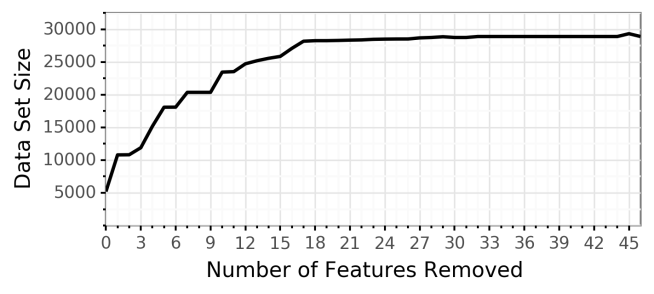
| Parameter | Range |
|---|---|
| Smoothing Parameter | $1 \times 10^{-1,2,3,4,5,6,7,8,9}$ |

**Table 3:** Range of parameters to test for Gaussian Naive Bayes model fit

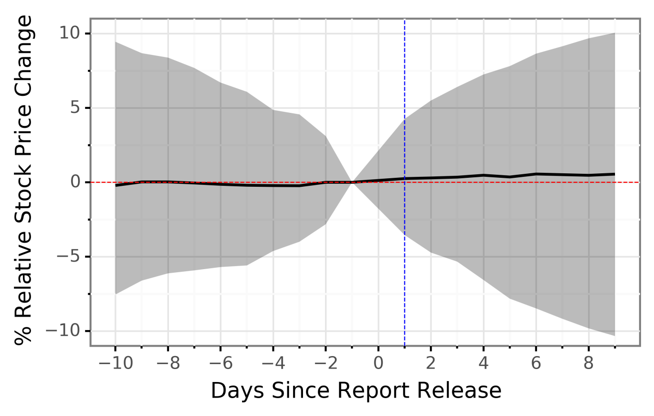
# Training

# Training - Logistics

- Predict whether the closing price 1 day after the report release is higher or lower than the price 1 day before.
- Binary classification
- 20-80 test-train ratio
- 5-fold cross validation
- Grid search through parameter space
- Database with 1, 5, 10, 15, 20, 25, 30, 35, 40 and 45 features removed (except from time series)



# Training - Prior



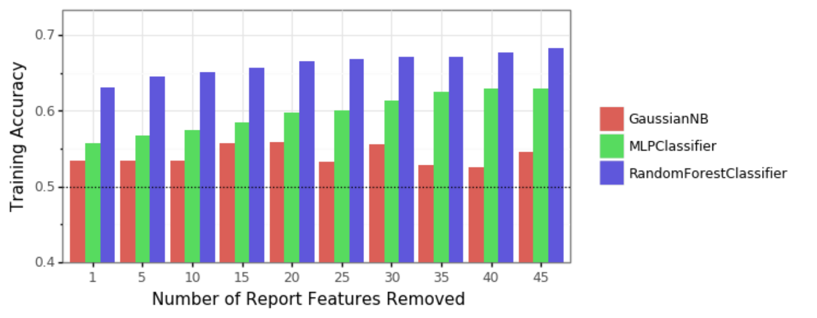

Bands represent 5th and 95th percentile, solid line is the median

# Results

# Results


- Relative to a ≈ 0.556 prior

# Results




- Relative to a $\approx 0.556$ prior
- 5-fold mean cross validation accuracy

# Results - Summary

- Better with less and larger dataset
- Just predicting stock price?
- Gaussian Naive Bayes does worse than trivial!

# Results - Model Performance

- Gaussian Naive Bayes
    - Worse than trivial performance
    - Gaussian assumption does not hold!
    - Correlated features
    - High dimensionality
- Neural Network
    - Good results
    - Best with 2 hidden layers, high L2 regularization, logistic activation, and LBFGS.
- Random Forest
    - Best Results
    - More trees $\rightarrow$ more better (no overfitting)
    - Better with less features (fewer misleading trees)

# Further Research

# Further Research

- Why is the quarterly report data not important?
- Better normalization?
- Use a better subset of features
  - Which features are most significant?
  - Search for features that add accuracy?
- Predict magnitude of change
- More domain specific knowledge needed
- Is pre/post-quarterly report any different than just regular stock price?
- Better classifiers? Ensemble Methods?

# Further Research - Trying to hit the 70% accuracy goal!

Super Models:

- **Attempt 1:** A Bigger Random Forest
  - 2000 trees
  - Only sector, stock price/volume/range, month, day, and SP 500 information
  - **Results:** 69.2% Accuracy, marginally better
- **Attempt 2:** Neural Network Boosting
  - 30 networks
  - 1 hidden layer, 30 different hidden layer sizes
  - Soft voting
  - **Results:** 63.7% Accuracy, a little better

# Further Research - Trying to hit the 70% accuracy goal!

Smarter Feature Selection: Leave One Out Selection

1. Iterate through features
2. Fit random forest with 2000 trees leaving the one feature out
3. If the classifier performs better with the feature missing, don't include it in the final dataset
4. Create final dataset with selected features
5. Run clasification

- **Results:** Features: 41, Samples: 26491, Accuracy: 67.7%
- Features selected due to random model fitting noise?

# Conclusion

# Conclusion

- Worked... kind-of
- Random Forest worked best
- Breaking assumptions of Gaussian Naive Bayes ruined it
- Neural net did alright, needed high regularization to not overfit
- Automated feature selection and creating super models failed.
- Need better method for selecting features

# References


[1] Iexcloud. *https://iexcloud.io/docs/api/*. Accessed: 2020-03-07.

[2] Stockpup. *http://www.stockpup.com/data*. Accessed: 2020-02-27.

[3] Yahoo finance. *https://ca.finance.yahoo.com/*. Accessed: 2020-03-07.

# Presentation Link

*https://drive.google.com/open?id=1AdjXkXIrAqUClFH56REBIC7YoW9oN5jw*