# Machine Learning in Stock Price Movement Direction Forecasting

DUKE COMPUTER SCIENCE

Research
Tatiana Jiayi Tian
Supervisor: Prof. Rong Ge

## Objective

financial statement | daily stock prices | management data | related events →


The objective is to predict the moving direction of stock prices by analyzing financial and management data, daily stock prices, and news with machine learning algorithms.

## Data

155 numerical/categorical features in total

500 Stock tickers

| financial data 113 features | daily stock prices 30 features | management data 11 features | related news 1 feature |
|---|---|---|---|

**Ticker**: 500 largest market capitalization stocks listed on the NYSE and NASDAQ, components of S&P 500.
**Data time span**: 4/18/2014 - 1/25/2019
701402 records

Ex:

| Ticker | Date | 155 features |
|---|---|---|
| FB | 12/30/18 | ... |

## Methods

**collect data**

- financial statement data
- business related events
- daily stock prices
- management team data

→

**preprocess data**

- merge data
- one-hot encode categorical variables
- normalize num - erical variables

↓

**fit in algorithms**

- use Linear Regression, Decision Tree, Random Forest, SVM, and Artificial Neural Network (ANN) model, and compare model results

↓

**Optimize model performance and obtain results**

- select significant feature subset
- compare models to find outperforming models
- tune parameters of selected models to achieve the highest prediction accuracy rate

## Optimization

### Attempt 1: Improve performance with data normalization

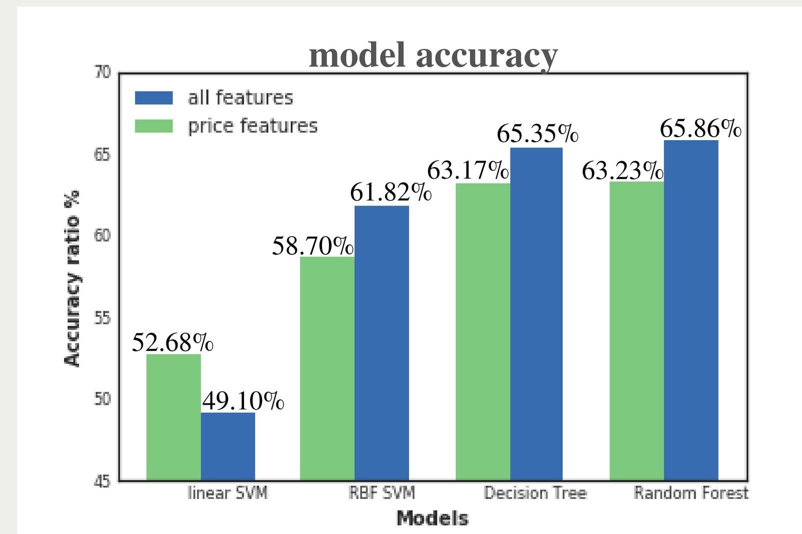
- respectively applied normalized data and non-normalized data to models such as Decision Tree and Random Forest
- compared accuracy ratios before and after data normalization

$$C = (C - \mathrm{mean}(C)) / \mathrm{std}(C)$$

**Result 1:** Normalized data show higher accuracy in DT (65.35%) and RF (65.86%) than non-normalized data (50.90%, 50.43%).

### Attempt 2: Improve performance with feature selection

- considered historically daily stock price features the most significant features due to high coefficients
- compared accuracy ratios between *all features* and *selected features*



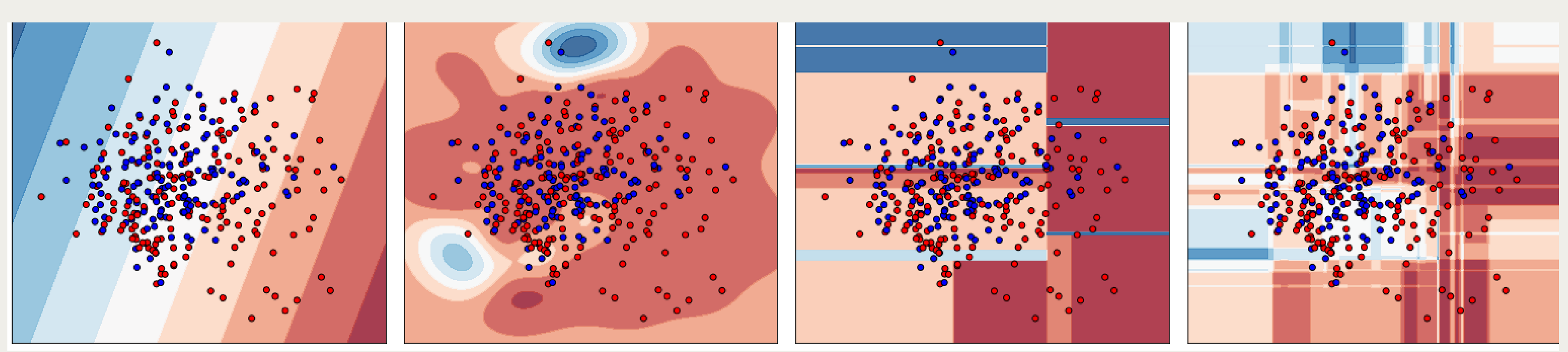

**Result 2:** Overall, *all features* show higher accuracy ratios than *price features* in 3 out of the 4 tested models.

### Attempt 3: Improve performance with algorithm selection

- applied *all features* data to Linear SVM, RBF SVM, DT, RF and ANN (artificial neural network)
- compared accuracy ratios among all models

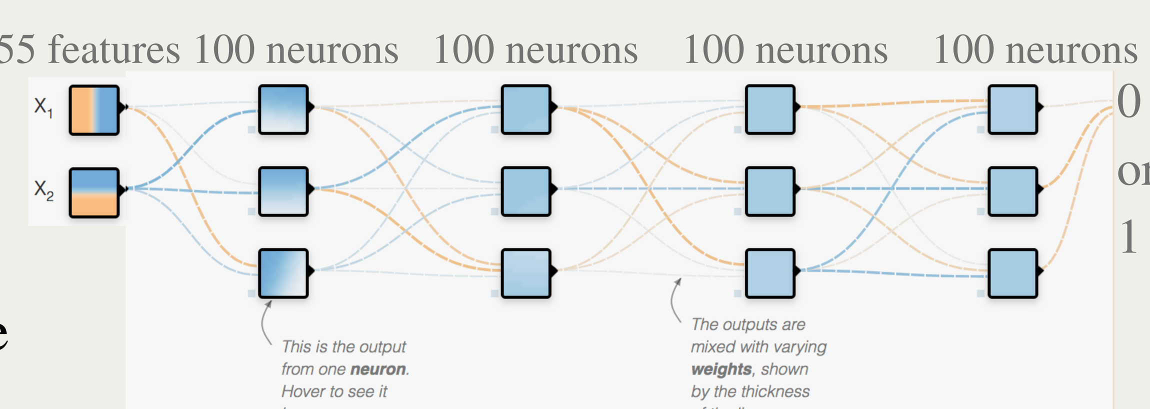
| Algorithm | Accuracy Ratios |
|---|---|
| Linear SVM | 49.10% |
| RBF SVM | 61.82% |
| Decision Tree | **65.35%** |
| Random Forest | **65.86%** |
| Neural Network* | **62.44%** |

*4 hidden layers, 100 neurons/hidden layer



Linear SVM | RBF SVM | Decision Tree | Random Forest

**Result 3:** DT, RF, and ANN show superior accuracy ratios; ANN has potential to achieve higher accuracy.



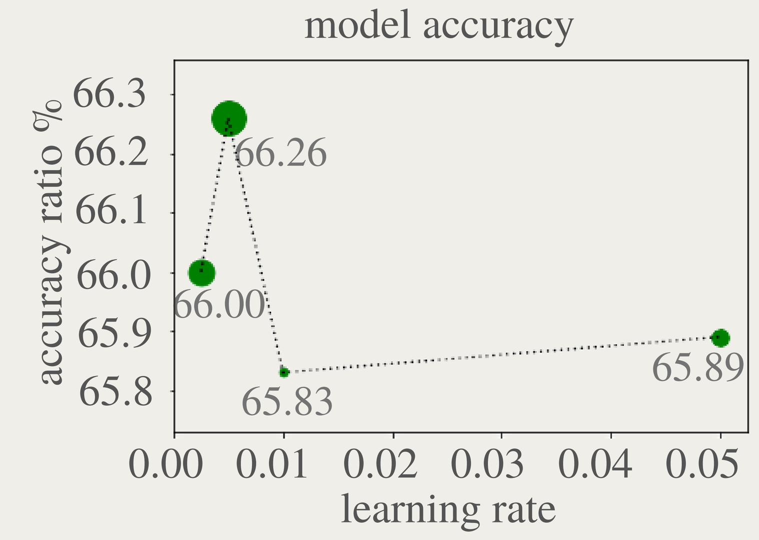

### Attempt 4: Improve performance with algorithm tuning (neural network)

- fit all features data in neural network based on attempt 2 & 3
- tuned # of hidden layers from 1 layer to 4 layers
- tuned neuron sizes from 500 to 1,000 neurons/hidden layer, tested 4 layers
- tuned fixed learning rates from 0.1 to 0.0025, adaptive learning rate (*0.5 every 5 epochs for 50 epochs in total)

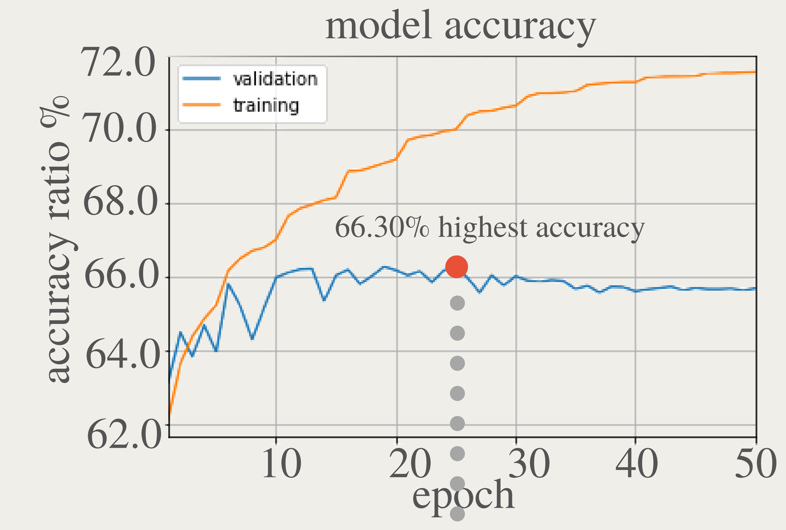
| layer 1 neuron | layer 1 accuracy | layer 2 neuron | layer 2 accuracy | layer 3 neuron | layer 3 accuracy | layer 4 neuron | layer 4 accuracy |
|---|---|---|---|---|---|---|---|
| **500** | **62.17%** | 500 | 62.66% | **500** | **63.20%** | **500** | **63.33%** |
| 750 | 62.03% | 750 | 62.55% | 750 | 63.18% | 750 | 63.19% |
| 1000 | 62.10% | **1000** | **62.60%** | 1000 | 62.17% | 1000 | 63.28% |

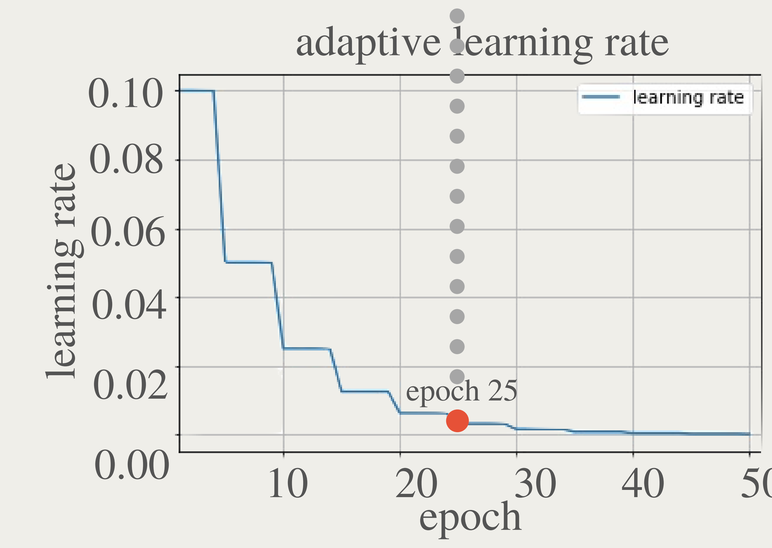
→ **optimized parameters**: **500, 1000, 500, 500** (4 hidden layers, neuron sizes)

**fixed learning rate = 0.05, epochs = 30**




**adaptive learning rate, epochs = 50**

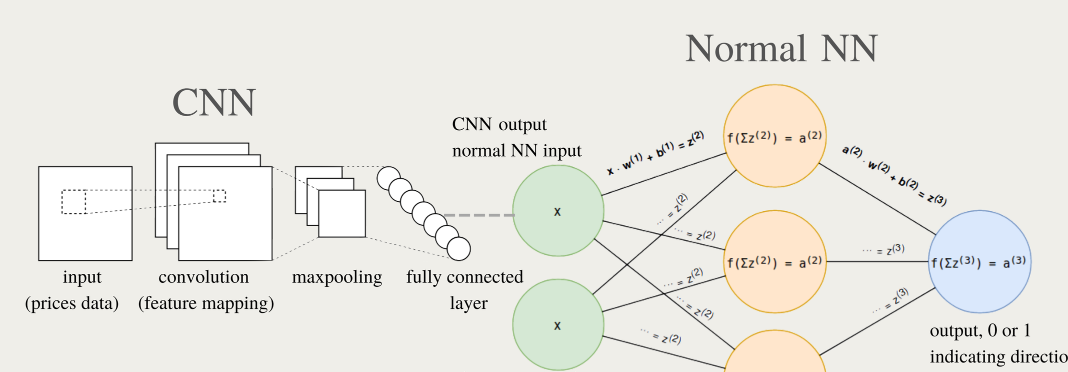



**Result 4:** The best performance, **66.30%**, was found in the ANN model with 4 layers, network size 500*1000*500*500, adaptive learning rate (*0.5 / 5 epochs) at epoch 25.

### Attempt 5: Improve performance with algorithm ensembles (CNN + normal NN)

- trained time sequential prices data with Convolutional Neural Network (CNN) and outputted probabilities of each direction
- used output of CNN, combined with other fundamental data, to fit in normal neural network




**Result 5:** With 2 convolutional layer in CNN and 4 hidden layers with adaptive learning rate in normal NN, algorithm ensembles produce **65.78%** accuracy rate.

## Conclusion

- Data normalization, feature selection, model selection, algorithm tuning and ensembles were used to optimize forecasting performance.
- The benchmark is 63%, the accuracy ratio achieved in studies of M.T. Leung et al [1].
- 66.30%, accuracy ratio achieved in this research, outperformed the benchmark.

## References

1. Leung MT, Daouk H, Chen A. Forecasting stock indices: a comparison of classification and level estimation models. Int J Forecast. 2000; 16(2):173–190.
2. Qiu M, Song Y (2016) Predicting the Direction of Stock Market Index Movement Using an Optimized Artificial Neural Network Model. PLoS ONE 11(5): e0155133. doi:10.1371/journal. pone.0155133