R for Explainable Stock Price Predictions

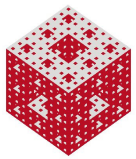
Sou-Cheng (Terrya) Choi

IIT Applied Mathematics and Allstate Insurance

Thanks: **Jackson Kwan, Adam Ginensky, Justin Shea**

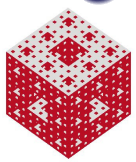
May 15, 2019

CRUG Pre-R/Finance



Bits and Pieces About Me

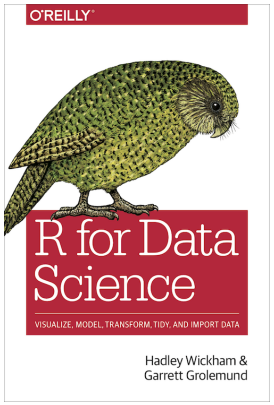
1. Research Associate Professor, Applied Math, IIT
2. Lead Researcher in Machine Learning, Allstate
3. Instructor of SCI-498 Machine Learning Algorithms on Heterogeneous Big Data @ IIT this Summer
4. Co-creator of MATLAB package Guaranteed Automatic Integration Library (GAIL) for one or high dimensional integration with guaranteed accuracies
5. PhD in Computational Mathematics & Engineering, MS in Statistics and Applied Probability, BS in Computational Sciences and Mathematics.
6. Originally from Hong Kong. Lived in Singapore and California.



My R Journey: Acquired Taste

Before R: Matlab, Java, JEE, C++, C, Fortran, Pascal.

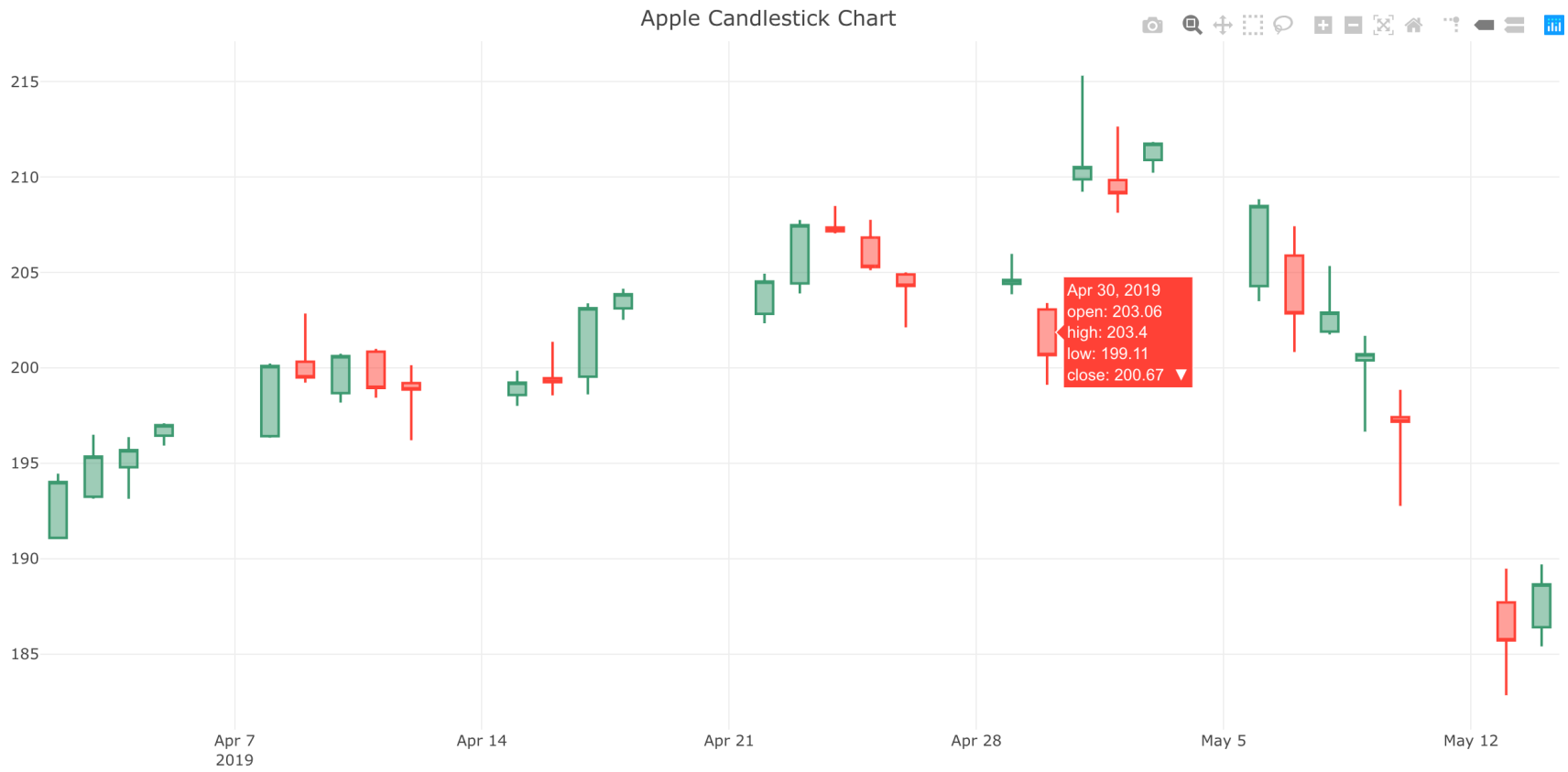
1. Looked at expert code by my collaborators and colleagues.
2. Read documentation of R packages.
3. Listened to talks
   1. by Hadley Wickham.

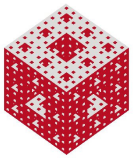


`https://r4ds.had.co.nz`

   2. Attend CRUG meetings and R/Finance days
4. Learn Shiny from RStudio Online Learning:
   `https://www.rstudio.com/online-learning/`

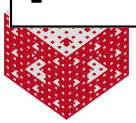
Example 1: Interactive Candle Stick Chart, APPLE





Demo 1: Interactive Candle Stick Chart, APPLE

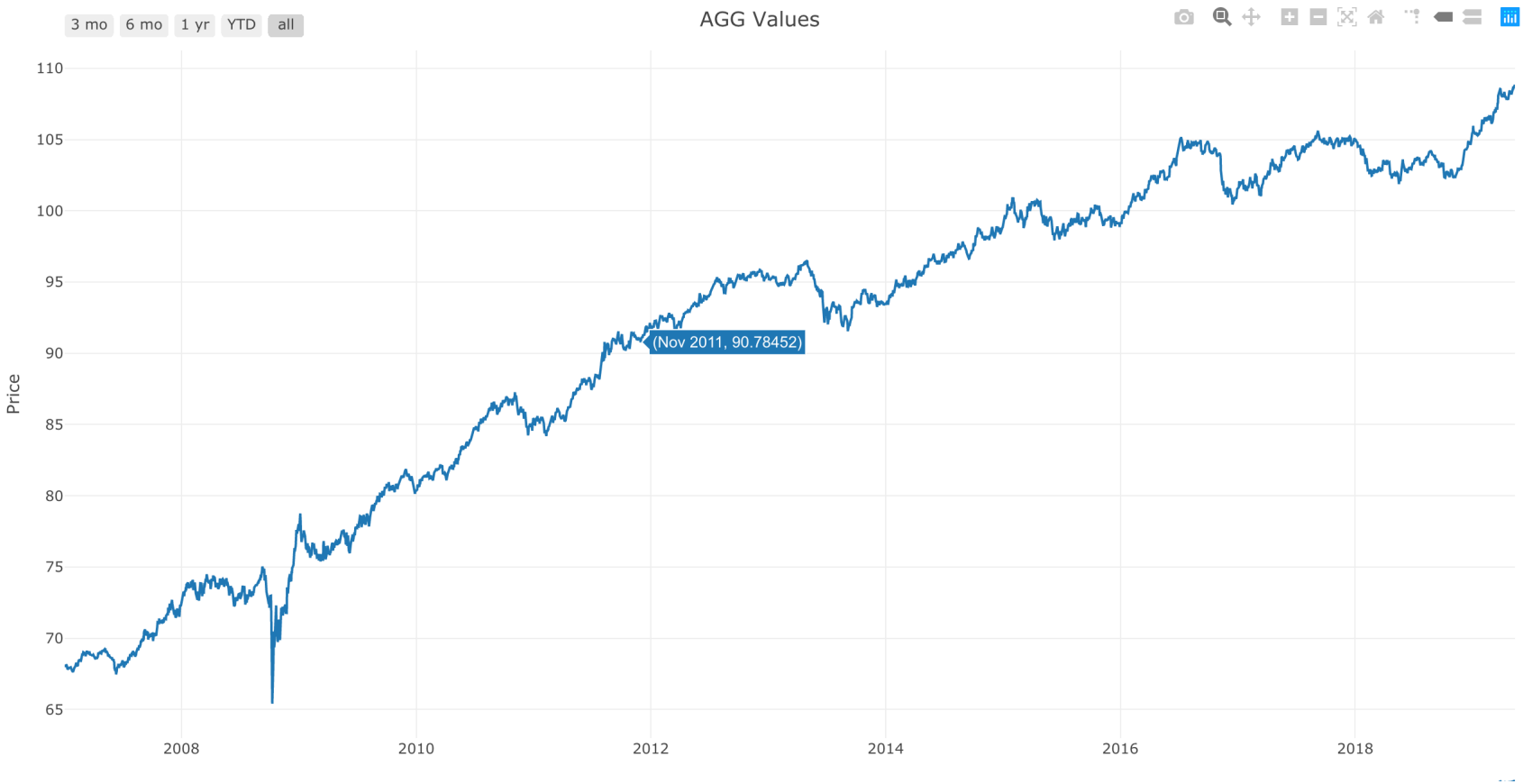
```
library(pacman)
p_load('plotly', 'quantmod')

### Set working directory to where this R script is
directory_of_this_script = dirname(rstudioapi::
    getActiveDocumentContext()$path)
setwd(directory_of_this_script)

### download data
getSymbols("AAPL", src='yahoo')
df <- data.frame(Date=index(AAPL), coredata(AAPL))
df <- tail(df, 30)

### chart of open, high, low, close prices
p <- df %>%
  plot_ly(x = ~Date, type="candlestick",
          open = ~AAPL.Open, close = ~AAPL.Close,
          high = ~AAPL.High, low = ~AAPL.Low) %>%
  layout(title = "Apple Candlestick Chart")

p
```



./code/ex1.R

Example 2: Interactive AGG Close Time Series

AGG: iShares Barclays Aggregate Bond Fund



Demo 2: Interactive AGG Close Time Series Code

```
library(pacman)
p_load('plotly', 'quantmod')

# Download data
getSymbols(Symbols = c("AGG"))
ds <- data.frame(Date = index(AAPL), AGG[, "Close"])

p <- plot_ly(ds, x = ~ Date) %>%
  add_lines(y = ~ AGG.Adjusted, name = "AGG") %>%
  layout(
    title = "AGG Values",
    xaxis = list(
      rangeselector = list(buttons = list(
        list(
          count = 3,
          label = "3 mo",
          step = "month",
          stepmode = "backward"
        ),
        list(
          count = 6,
          label = "6 mo",
          step = "month",
          stepmode = "backward"
```

Example 2: AGG Minute Data (from eSignal)

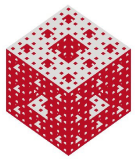
Used R packages `xtable` for generating LaTeX table from R data frame and `basicStats` for computing statistics.

First five rows

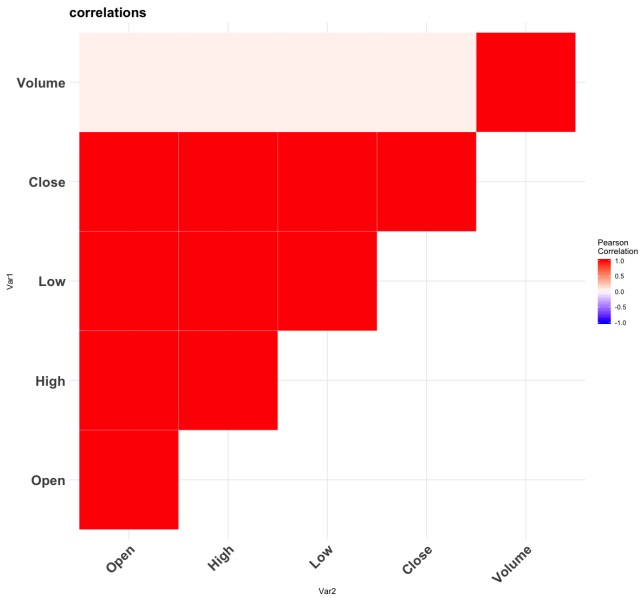
Date	Time	Open	High	Low	Close	Volume
01/03/2006	11:24:00	100.62	100.62	100.61	100.61	900
01/03/2006	11:25:00	100.60	100.60	100.60	100.60	100
01/03/2006	11:26:00	100.57	100.60	100.57	100.60	300
01/03/2006	11:27:00	100.57	100.57	100.57	100.57	300
01/03/2006	11:30:00	100.57	100.57	100.57	100.57	300

Statistics

	Open	High	Low	Close	Volume
nobs	1051297.00	1051297.00	1051297.00	1051297.00	1051297.00
NAs	0.00	0.00	0.00	0.00	0.00
Minimum	87.22	87.26	86.80	87.21	1.00
Maximum	113.27	113.27	113.26	113.27	6398413.00
Mean	106.66	106.67	106.66	106.66	3660.05
Stdev	4.13	4.12	4.13	4.13	18893.03



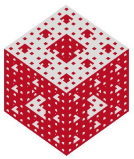
Example 2: AGG Close Prediction Model



Want to build a prediction model for predicting minute close

$$C_t = f(C_{t-1}, O_t, O_{t-1}, H_{t-1}, L_{t-1}, V_{t-1}, \text{engineered features})$$

Data	Summary
All data	2013-01-01–2018-10-03 (525195 records)
Training data	2013-01-01–2016-12-31 (356502 records)
Test data	2017-01-01–2018-10-03 (168693 records)

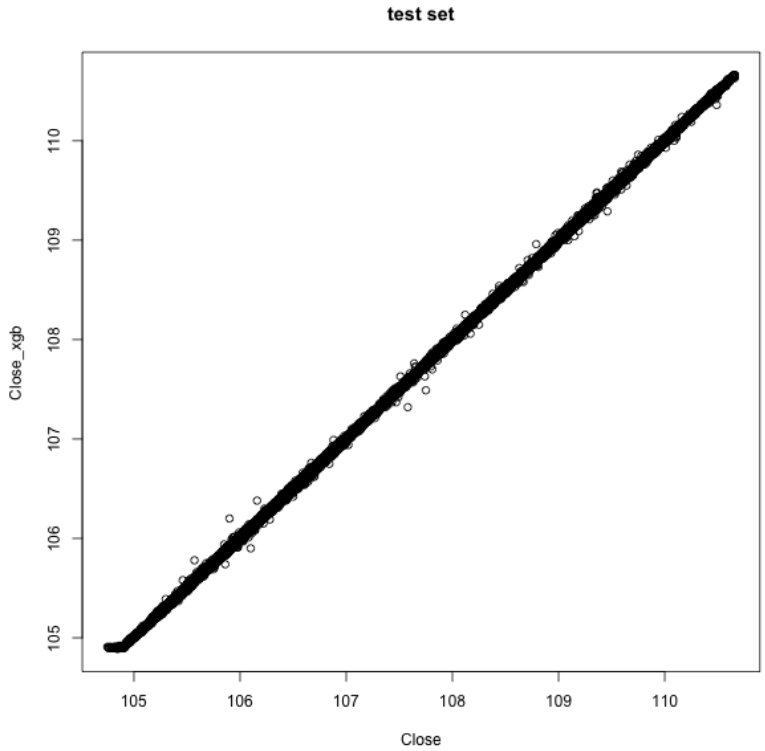


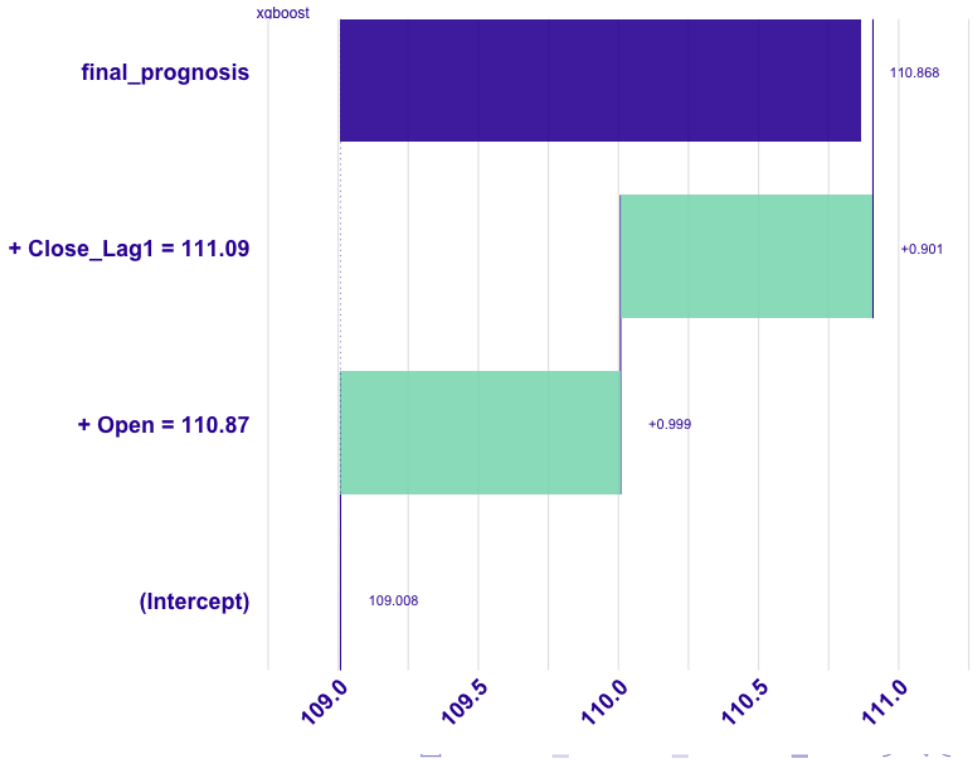
Predictions and Record-Level Factors

Use R package `xgboost` for building ensemble models and `DALEX` (Descriptive mAchine Learning EXplanations) for explaining model decisions for every record.

$$\text{Mean absolute error (MAE)} := \sum_{t=1}^{n} |C_t - \hat{C}_t|/n$$

model_name	train.mae	test.mae
Xgboost	0.0057	0.0059

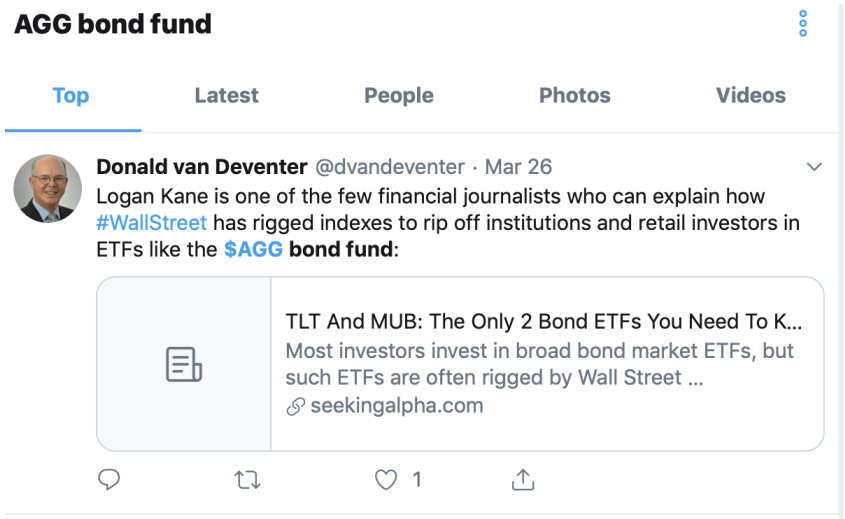




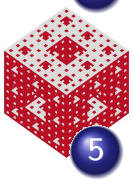
Conclusions and Future Work

Very preliminary work and results. Need to do

1. hyperparameter search
2. play with TensorFlow to predict OHLC's
3. engineer more features or join in more data, e.g., tweets, sentiment analysis

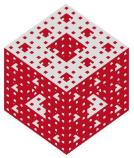


4. more careful evaluation of approaches and model performance via large scale testing against various stocks and funds
5. construct on top a portfolio optimization framework



Q&A

Question: Is MAE = 0.05 good enough? What should we target the accuracy to be?



References

1. Biecek, P., 2018. *DALEX: explainers for complex predictive models in R*. The Journal of Machine Learning Research, 19(1), pp.3245-3249.
2. Chen, T. and Guestrin, C., 2016, August. *Xgboost: A scalable tree boosting system*. In Proceedings of the 22nd ACM SIGKDD (pp. 785-794). ACM.
3. Choi, S.C. et al., 2019, *Real-time Prediction of Traffic Speed During Traffic Incidents*, SIAM Conference on Computational Science and Engineering, `http://tinyurl.com/y5ndj88t`
4. Dixon, M.F., Polson, N.G. and Sokolov, V.O., 2018. *Deep learning for spatio-temporal modeling: Dynamic traffic flows and high frequency trading*. Applied Stochastic Models in Business and Industry.
5. Gunning, D., 2017. *Explainable artificial intelligence*. Defense Advanced Research Projects Agency (DARPA), `http://tinyurl.com/yayu9wtx`
6. Ribeiro, M.T., Singh, S. and Guestrin, C., 2016, August. *Why should i trust you?: Explaining the predictions of any classifier*. In Proceedings of the 22nd ACM SIGKDD (pp. 1135-1144). ACM.