INFORMS Data Mining Contest 2010 (2nd Place)

Improved Stock Price Predictions via Pre-Processing

Christopher Hefele

c.hefele@verizon.net

www.linkedin.com/in/christopherhefele

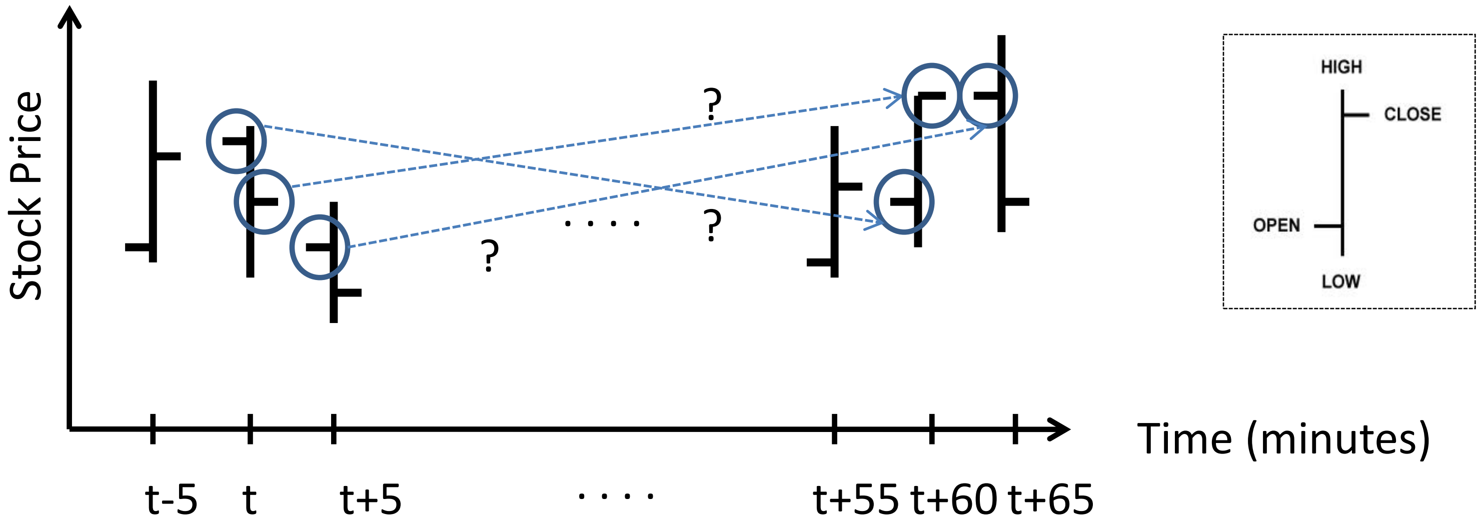
Contest Description

- Goal: Predict if an unnamed stock will go up or down in one hour
- Dataset Description
  - 609 variables provided
    - Other stock prices, sectoral data, economic data, experts' predictions, indices
  - Data given for each 5-minute period
  - 5922 periods in training set
  - 2539 periods in test set

Solution Overview

- Create returns variables from prices
- Time-of-Day normalization of returns
- Percentile transform of returns
- Forward stepwise variable selection
- Classifier
  - Logistic regression with L2 regularization
  - SVM w. RBF kernel (used only briefly)

Create Returns Variables from Prices

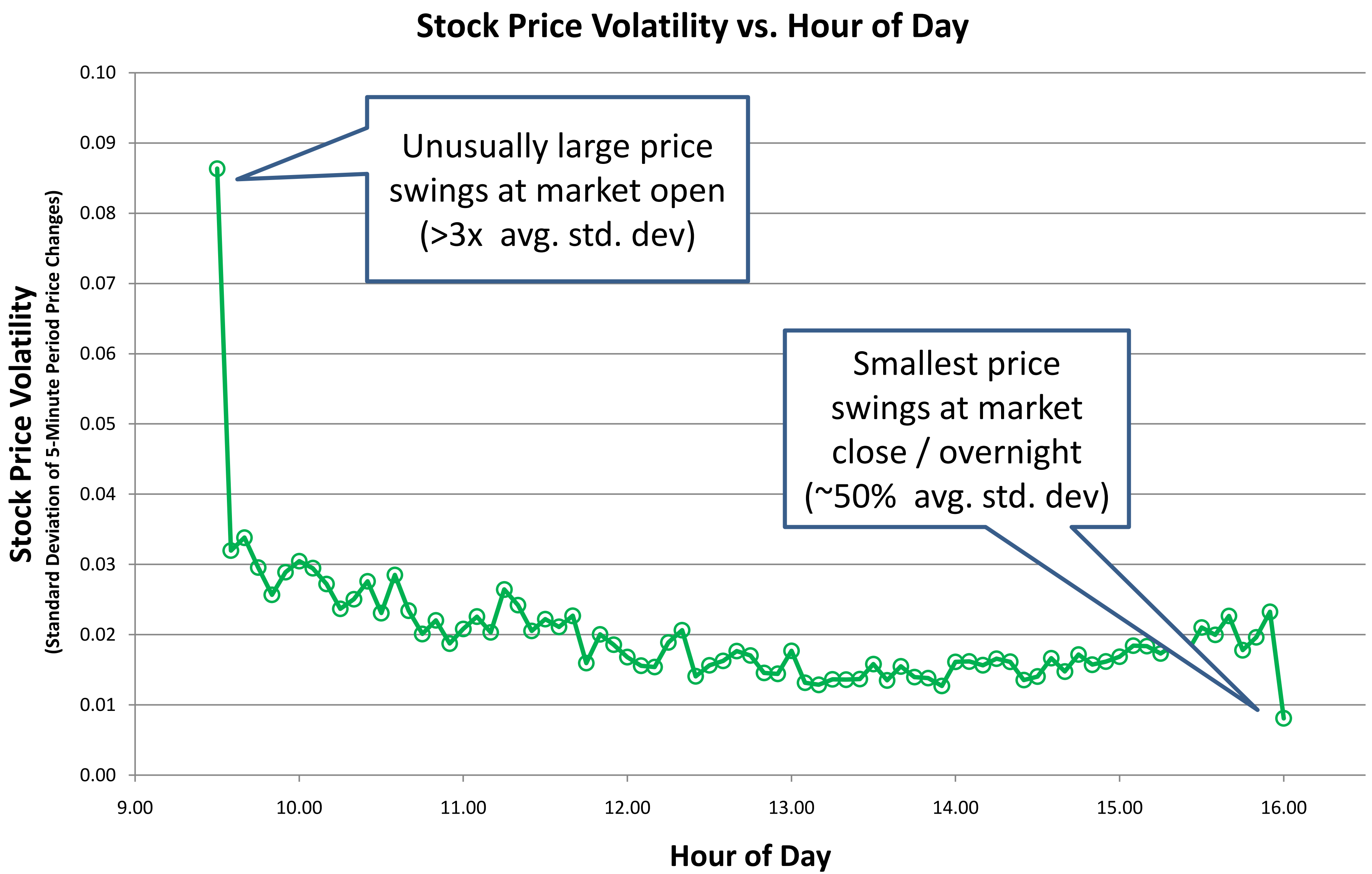


- Target Variable is 1 hour price change in an unknown stock…but…
  - Are those changes in OPEN or CLOSE prices after 1 hr? Something else?

- Created new returns variables from each stock's prices, for later variable selection
  - Return(t,L) = log Price1(t+Lag) – log Price2(t+60+Lag) , where:
    - Price1 & Price2 are OPEN, HIGH, LOW or LAST prices for a given stock
    - Lag = one of: -5, 0 or 5 minutes

Variable Selection

- Used forward stepwise logistic regression
  - Included top 3 selected variables in the final model (to minimize overfitting)
  - L2 regularization also used with an automatic parameter tuner & K-fold cross-validation
- Why not use L1 regularization to select variables?
  - Forward stepwise variable selection + L2 regularization seemed to outperform L1 regularization on this data



Stock Price Volatility vs. Hour of Day
Unusually large price swings at market open (>3x avg. std. dev)
Smallest price swings at market close / overnight (~50% avg. std. dev)
Stock Price Volatility
(Standard Deviation of 5-Minute Period Price Changes)
Hour of Day
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.10
9.00
10.00
11.00
12.00
13.00
14.00
15.00
16.00

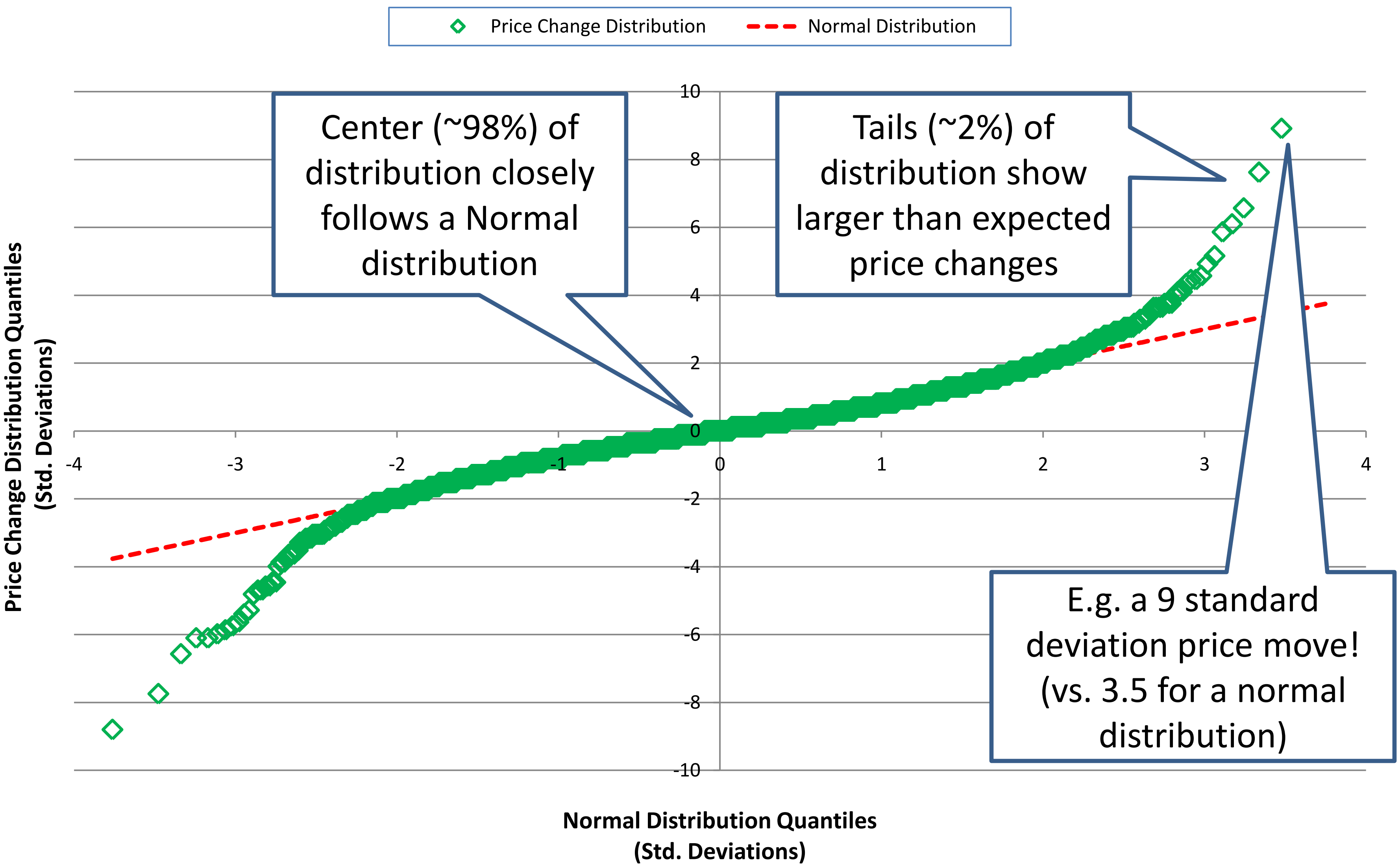
Time-of-Day Normalization

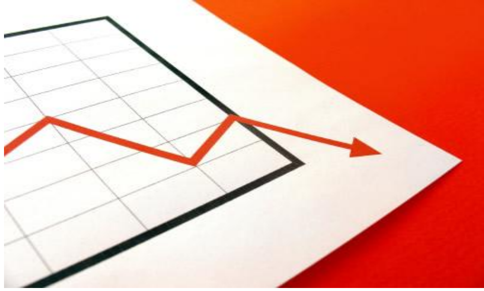
- *Problem*: Volatility variations degrade classifiers' accuracy
  - Total error (& fit) may be dominated by largest swings (aka 'outliers')
  - Smallest swings may be partially 'ignored' if use L1 or L2 regularization (or any other penalty for larger regression weights)

- *Solution*: Normalize each 5-min time period <u>separately</u>
  - Bin each variables values by 5-min time period
  - Divide each bin's values by that bin's standard deviation

*Volatility = standard deviation of the set of price changes or returns

Price Change Distribution vs. Normal Distribution

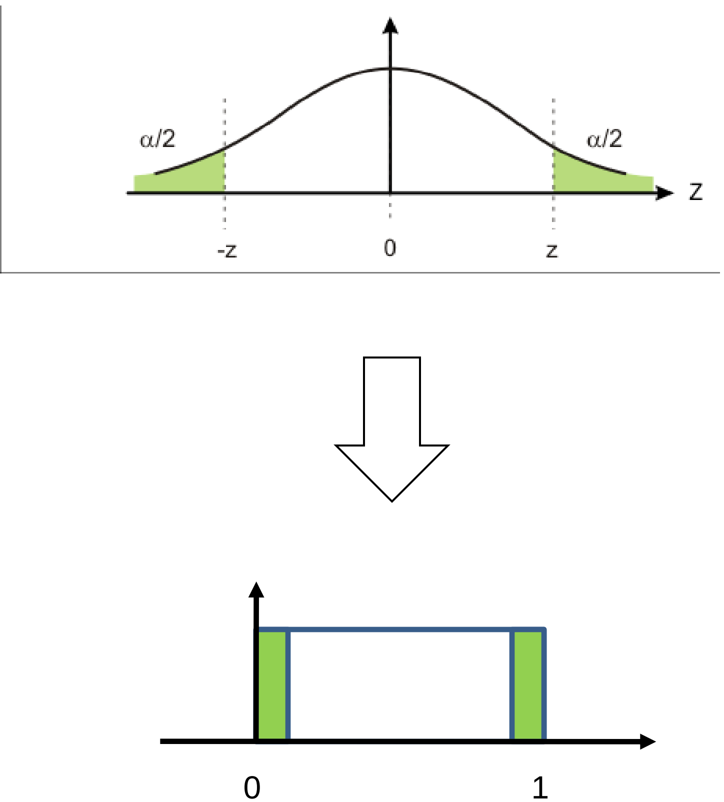




Price Jumps

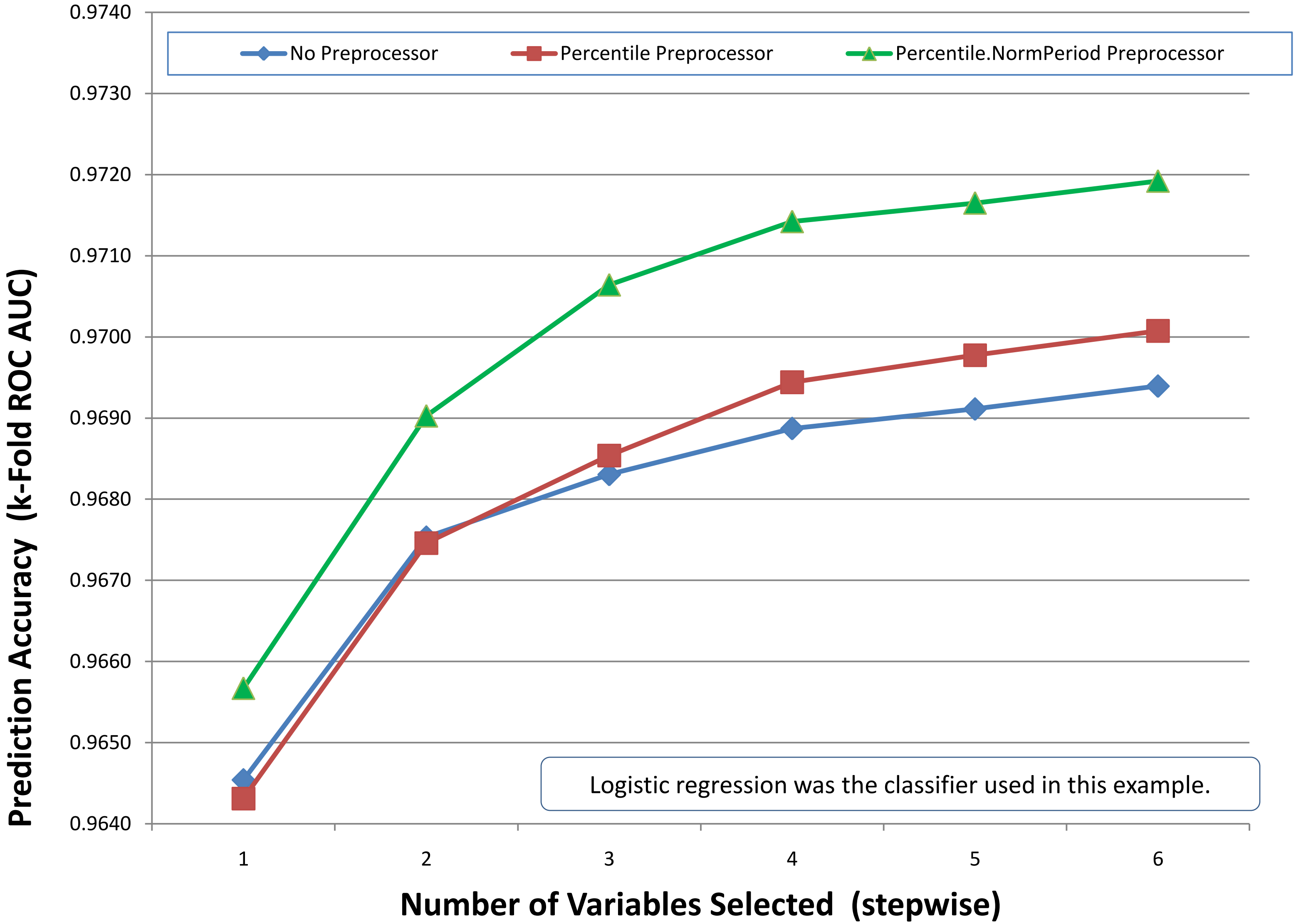
- Price-change distribution *mostly* normal, but…there are infrequent, large price jumps
- "Long-tail" / leptokurtic distributions of returns often reported in the financial literature
  - Power-law distribution of returns often seen in tails
  - Typical causes of unusually large price swings include earnings announcements, press releases, crashes, etc.

Percentile Transform

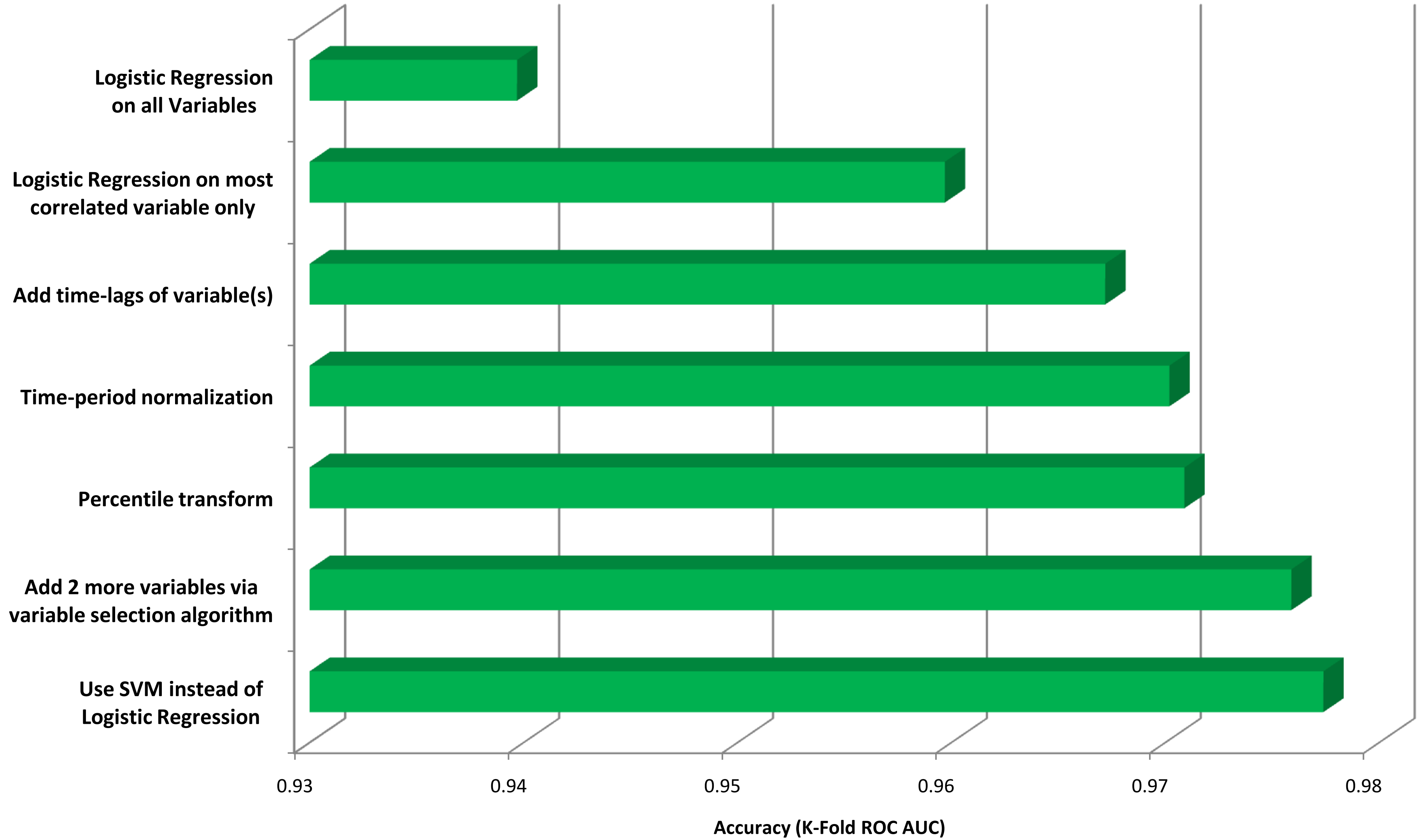


- Large price jumps can degrade classifier accuracy
  - Total error (& fit) driven by large swings
- Percentile Transform clamps jumps
  - Values in a distribution replaced by their percentile in that same distribution
    - E.g. (-10, 4, 3, 11,80) → (0, .5, .25, .75, 1)
  - Clamps large price swings to [0,1]
  - Provided just a *small* increase in classifier accuracy when combined with var. selection + logistic regression

Prediction Accuracy vs. Preprocessors Used with Variable Selection



Summary of Improvements



Implementation Details

- Coded in python utilizing:
  - Scikits.Learn (machine learning library)
  - SciPy & NumPy (C-extension math libraries)
- OS:
  - Ubuntu Linux (Release 10.04, 64-bit)
- Hardware:
  - Intel Core2 Quad (2.33 GHz)

  - 8GB memory (only ~400M used in the competition)

Questions?