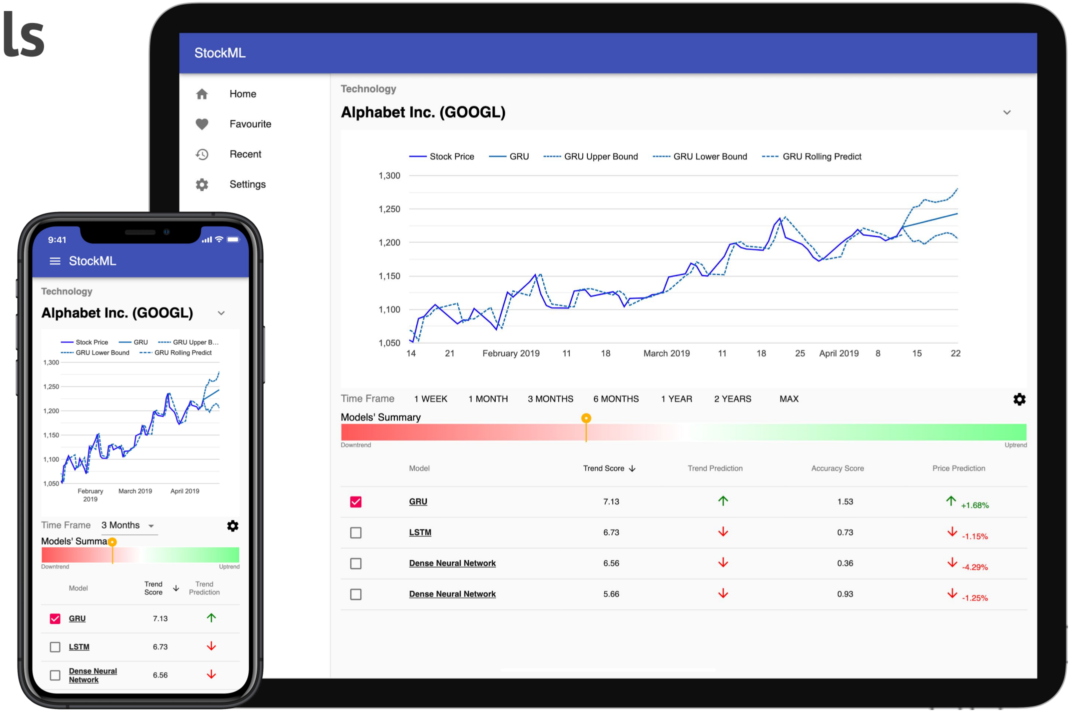
Stock Price Prediction App using Machine Learning Models Optimized by Evolution



Team RO4

CHAU Tsun Man,
SUEN Heung Ping,
TO Cheuk Lam,
WONG Cheuk Kin

Advised By

Prof. David ROSSITER

Motivation

Retail investors in Hong Kong

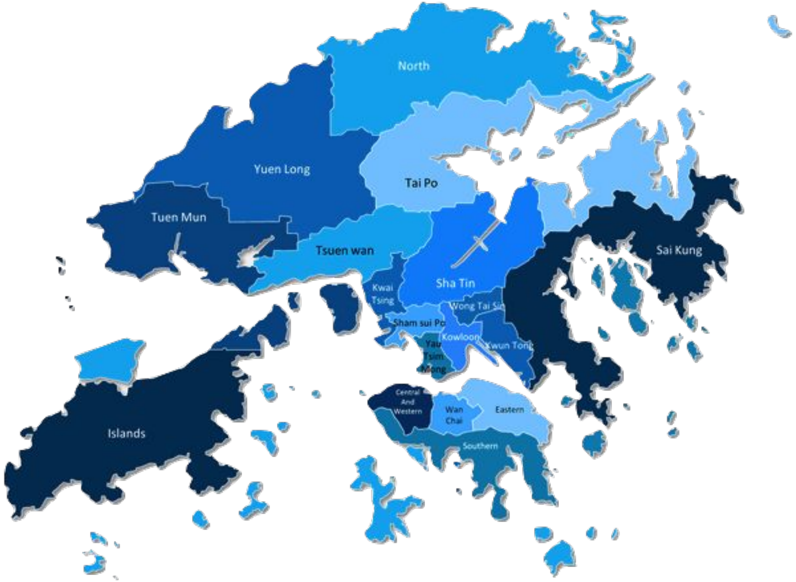
A lot of investment decisions that involve a large sum amount of money being made.

>2.2M
Retail Investors

~$1.6T
Market Turnover

~15%
Trading Value

~$240B
Turnover 2016



Motivation

Hurdles for retail investors to track market trend

Costly Financial Advices

Most retail investors have to figure out the market themselves
and make informed decisions on their own as advises are costly.

Investment Decisions Driven by Emotions

Without quantitative, data-driven models, decisions get swayed by
cognitive biases or personal emotions, resulting in unnecessary losses.

Access to Simple Tools Only

Using moving averages and technical indicators may help,
but may be limited in its prediction power.

Objectives

Research

Model for 10-Days Prediction
Optimize Root Mean Squared Error (RMSE)

Parameters Search by Evolution Algorithm
Evolve and improve models to reduce RMSE

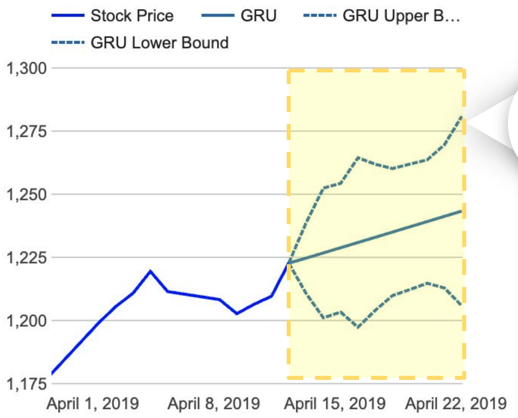
Application

Responsive Web Application
Easy to use user-friendly interface

Compare and Contrast Models Easily
Make machine learning easy for ordinary users

"Our project aims to **democratize machine learning technologies** for retail investors."

Features



01

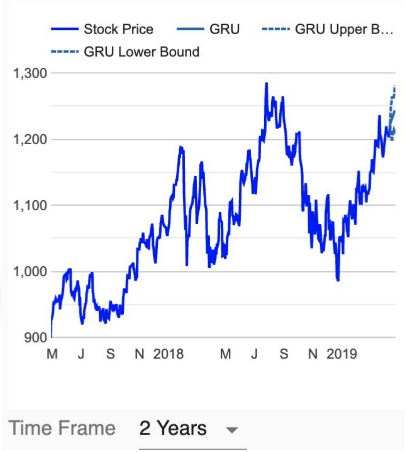
Price & Trend Prediction

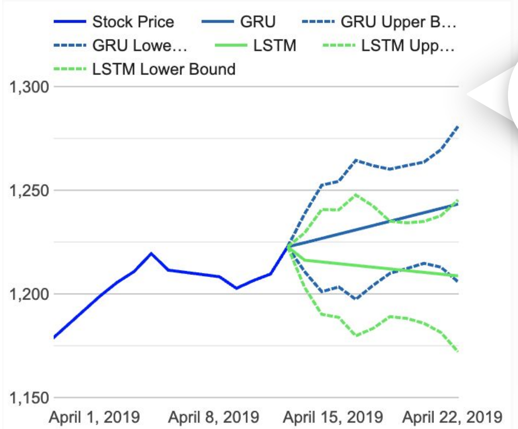
Upper/Lower Error Bounds and Buy/Sell Score for investors to quickly spot an opportunity.

Historical Stock Price

Up to 20 years of stock data enables investors to see how stocks move in the past.

02





03

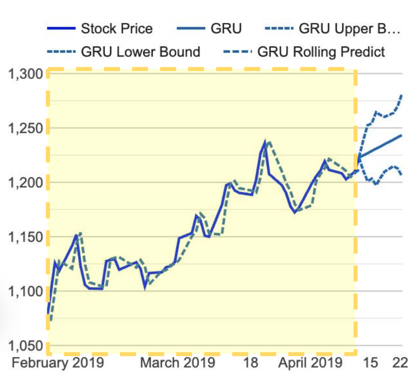
Different Models Compared Side-by-Side

Trend Scores and Predicted Direction for comparing how well individual models predict.

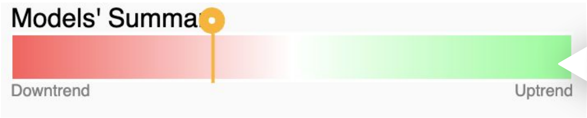
Test Set Performance for Advanced Users

Historical predictions with model configurations for advanced users to inspect.

04



Features



05

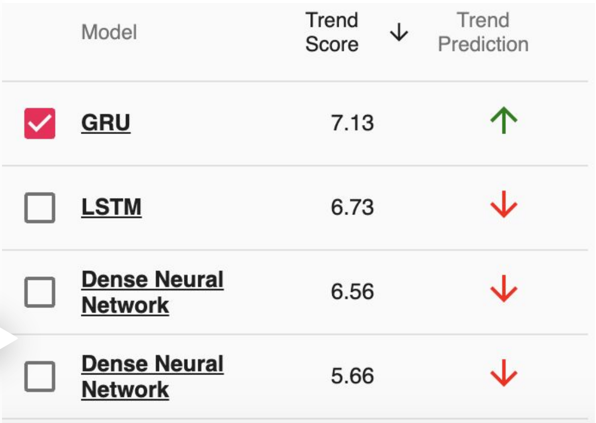
Models' Summary

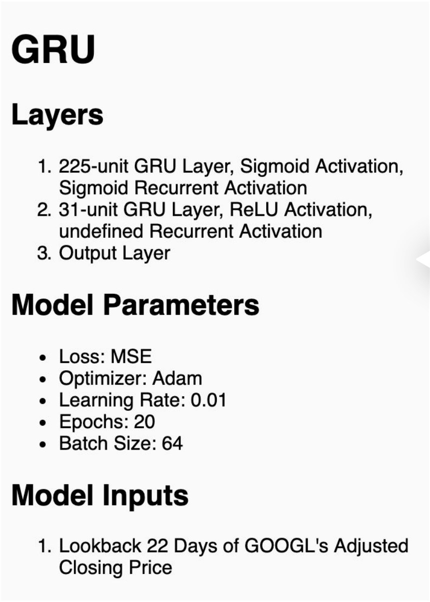
Let retail investors to quickly visualise the trend predictions of all models

Trend Stats for Each Model

Let users compare different models with its trend predictions

06





07

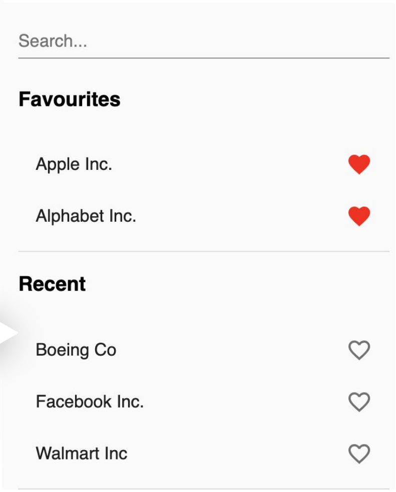
Detailed Model Configurations

Let advanced users to inspect what contributes to the predictions on the graph

Favourite and Recent List

Let users to view and choose the most frequently accessed stocks faster

08



Problem Framing

1. Input

Raw Stock Price

Stock Price Change

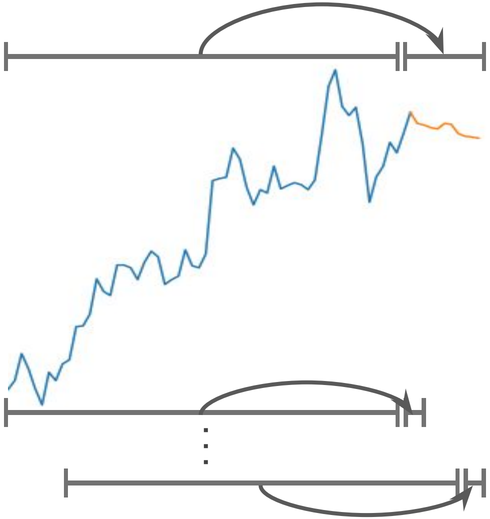
Moving Average

Lookback Time Frame

Min/Max Scaling

2. Process

A. 10-Day Direct Prediction



B. 1-Day Rolling Prediction

3. Output

Next

10

Day

Prices

by

Minimizing

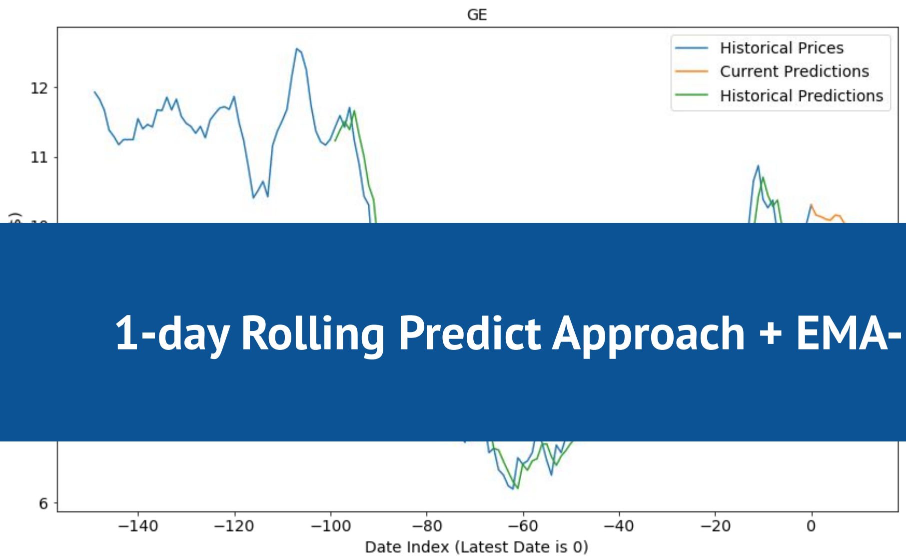
RMSE

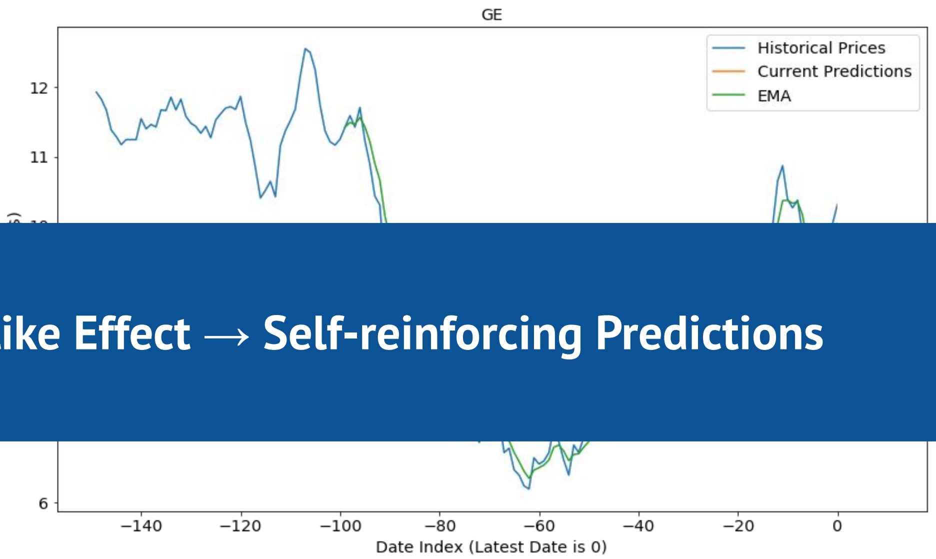
Demo

How Insightful are the Predictions?

Prediction Findings 1

Next Day Prediction Looks Like Exponential Moving Average, but with Predictive Power





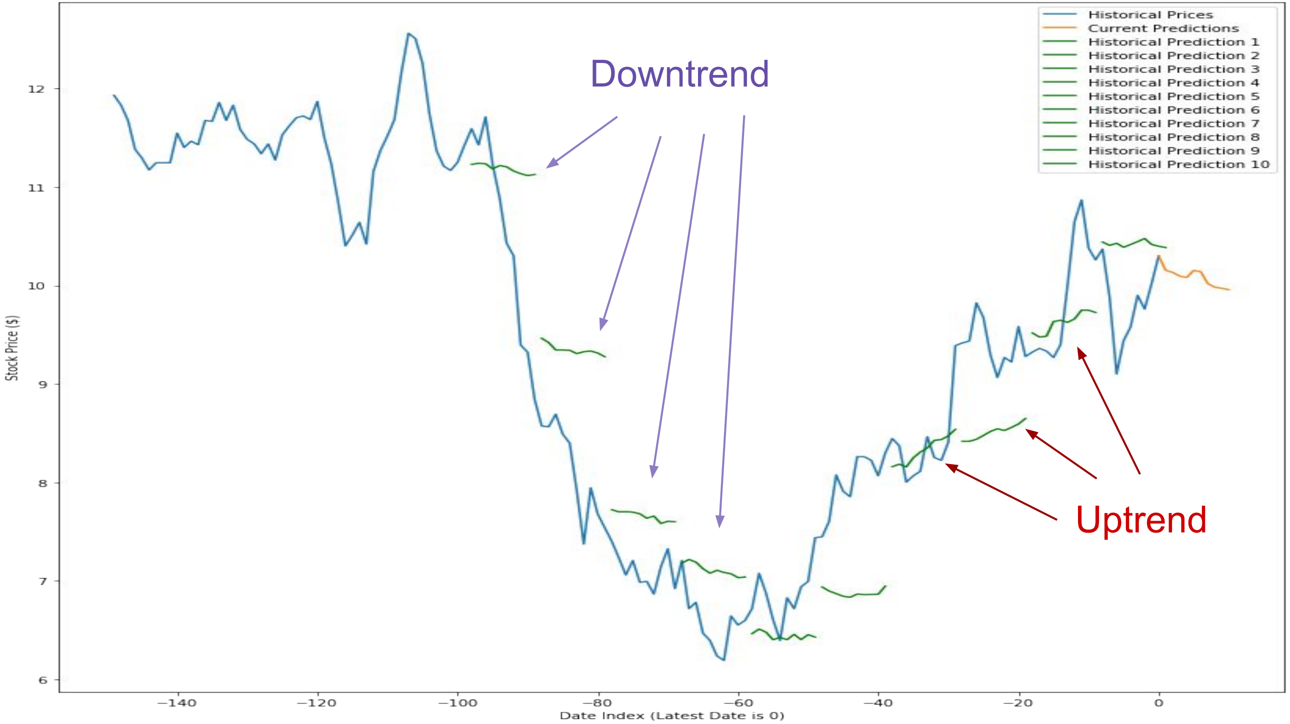
1-day Rolling Predict Approach + EMA-like Effect → Self-reinforcing Predictions

1-day Rolling Prediction

4-day Exponential Moving Average

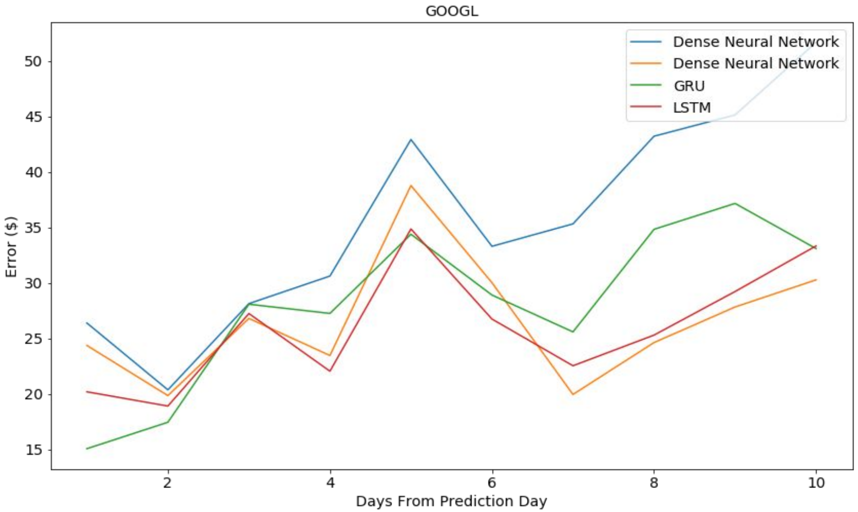
Prediction Findings 2

Difficult to Predict Exact Price Values, but Follow Trends

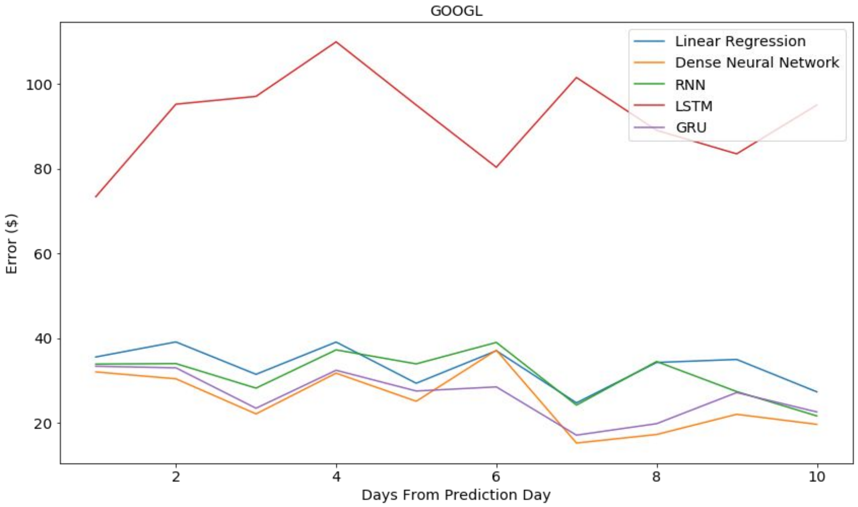


Prediction Findings 3

1-day Prediction is More Accurate for Closer Future,
but Accuracy Lowers over Time to Future



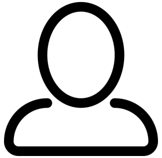
1-day Rolling Prediction



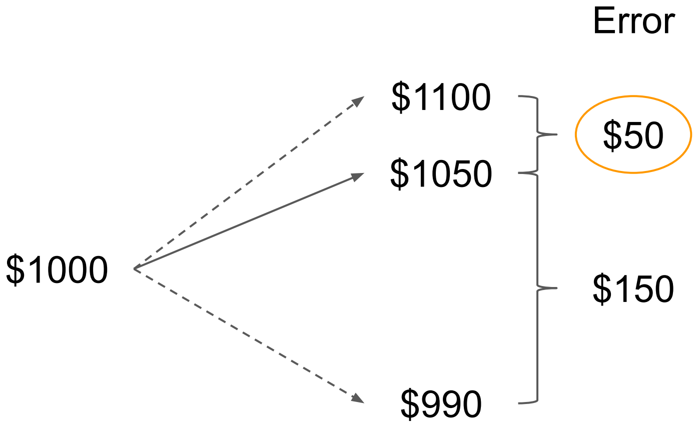
10-day Direct Prediction

Comparison Baseline

Historical Holding
Period Return
e.g. 10%



Hypothesized
Investor

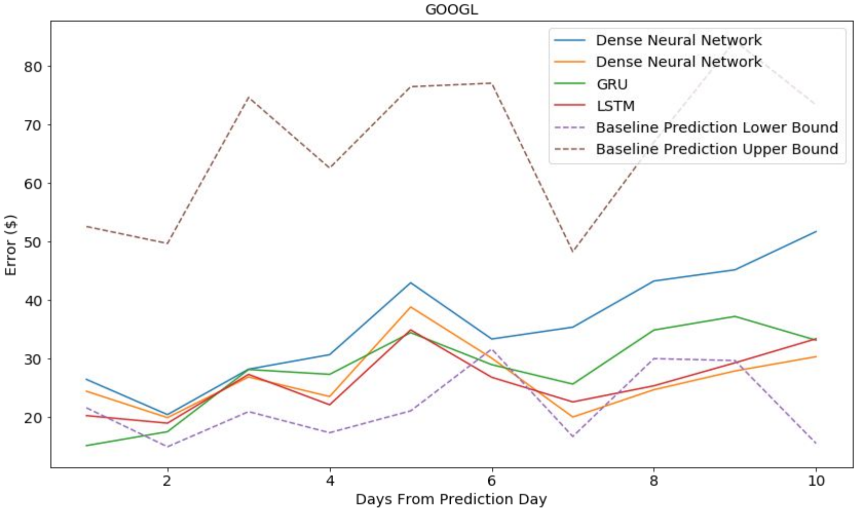


Assume always guesses right direction

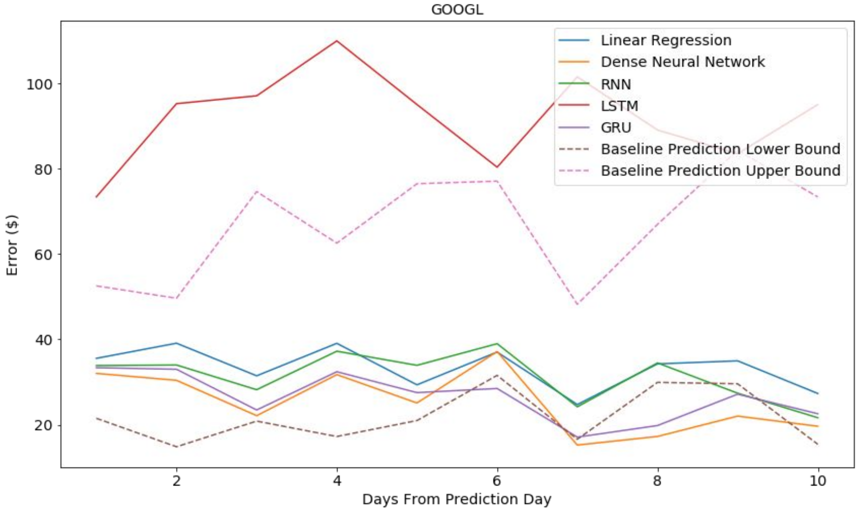
$$\text{Baseline Strategy Error} = min\{P_0 \cdot (1 + \overline{|HPR_i|}) - P_1,\ P_0 \cdot (1 - \overline{|HPR_i|}) - P_1)\}$$

Prediction Findings 4

Comparable Performance with Baseline Strategy



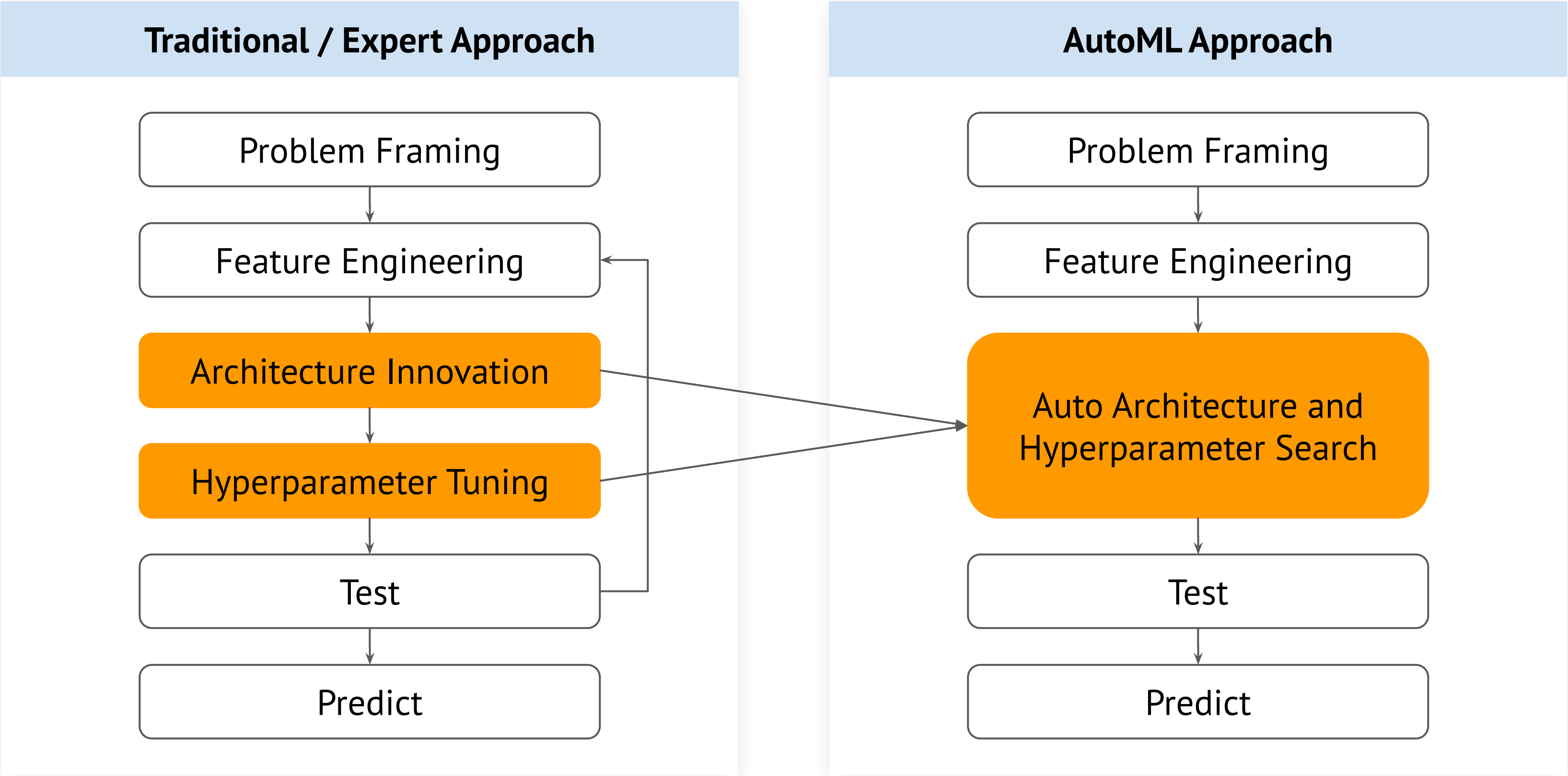
1-day Rolling Predict



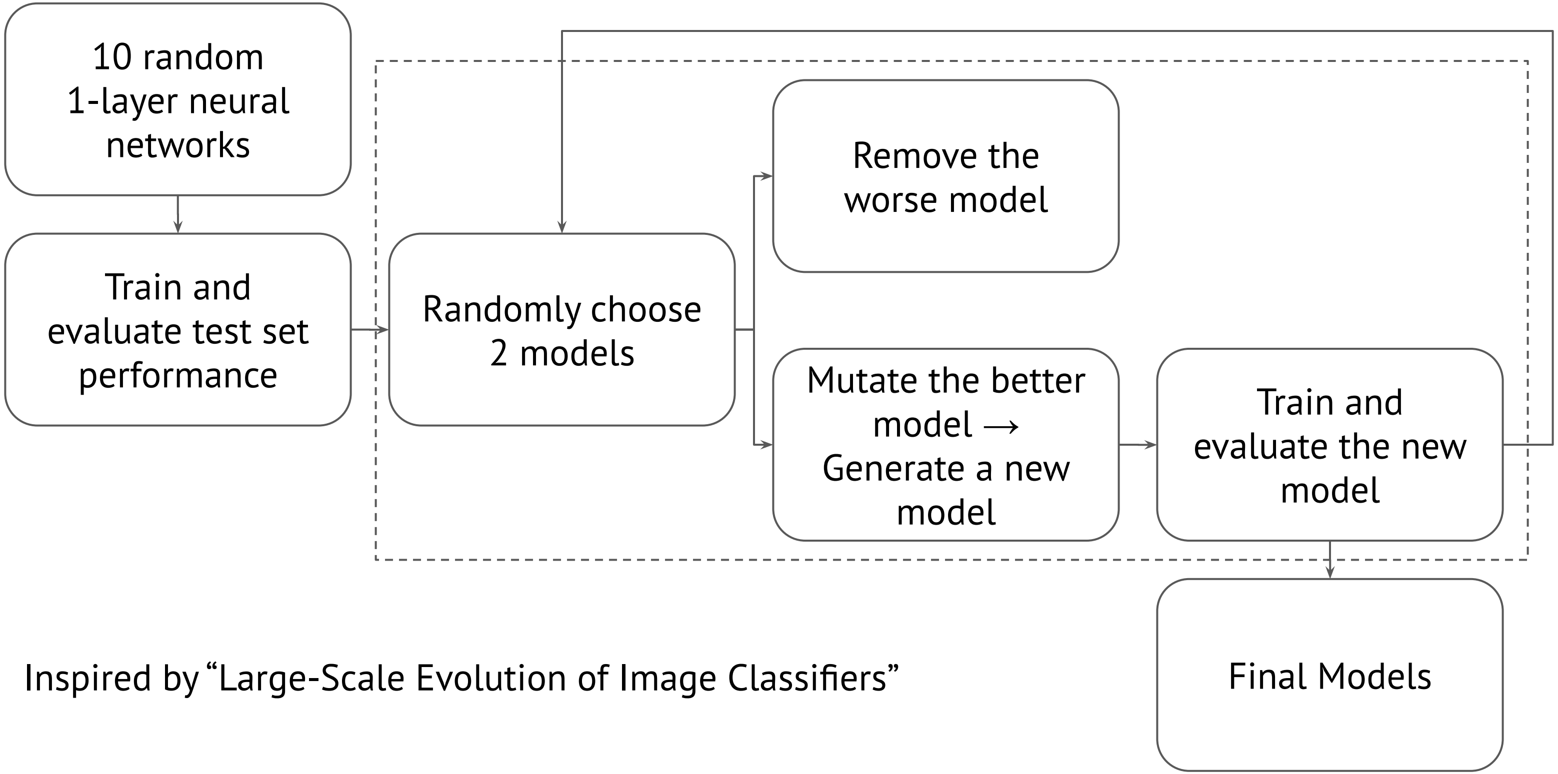
10-day Direct Predict

How Do We Achieve Such Models?

Machine Learning Pipeline



Evolution Algorithm



Inspired by “Large-Scale Evolution of Image Classifiers”

Theoretical Concerns

Search Bias

Heuristic Search → Simple Mutations

- Small Population Size (10)
- Few Iterations (100)
- Other Hyperparameters

→ Limited Search Space

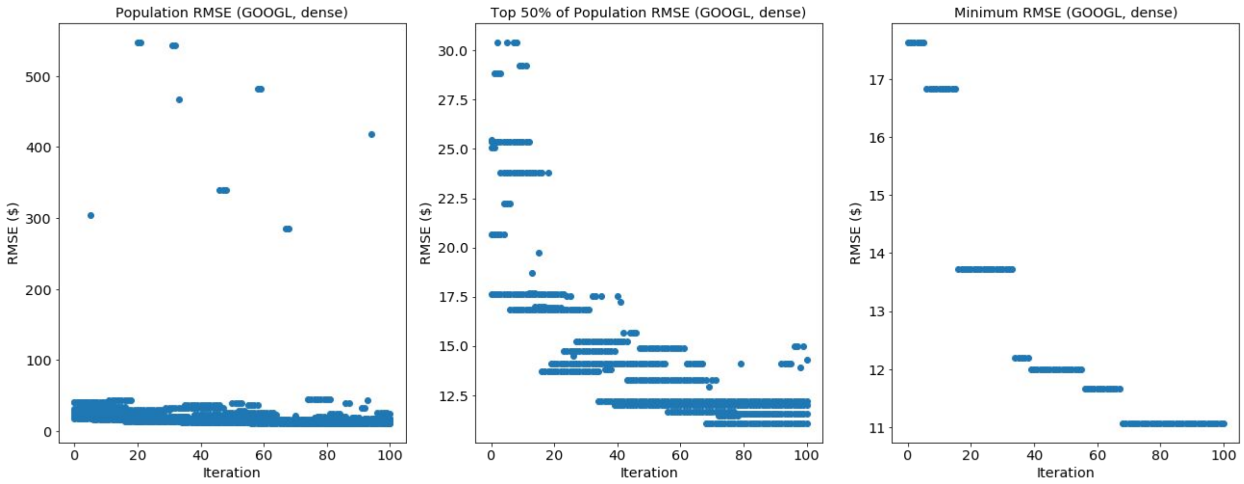
Language Bias

Feature Engineering

↓

Auto Architecture and Hyperparameter Search

Findings

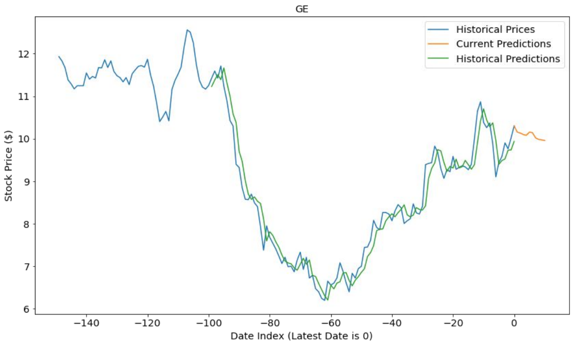


Algorithm Gradually Finds Better Models

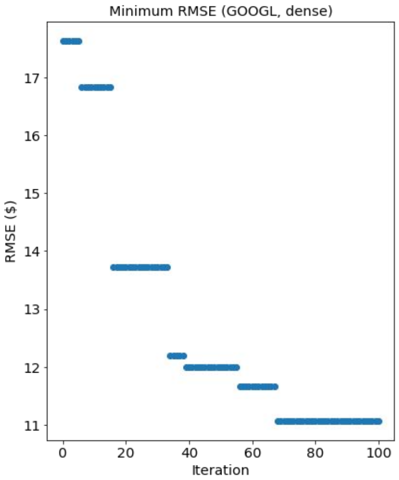
Simple Models (1-layer, linear)

- Too Few Data (just daily prices)
- Incorrect Problem Framing (input, output, objective function)

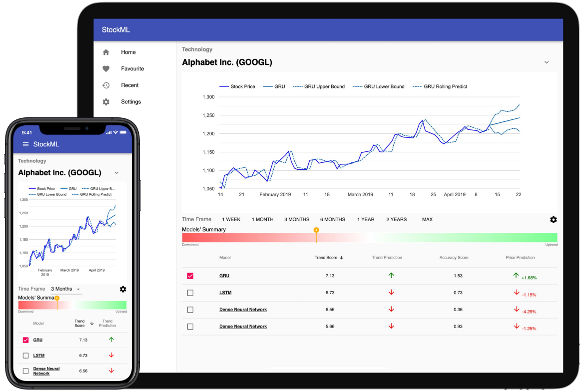
Democratize ML Technologies For Retail Investors



Reasonable Performance



AutoML (Evolution Algorithm)



Application with Good UX

Q & A

Appendix A - Snake

Definition

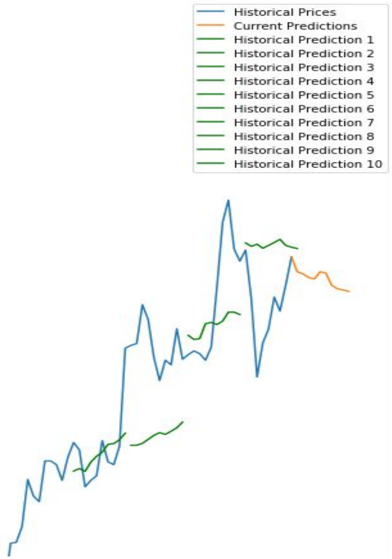
Snake: *a 10-day disjoint prediction segment given a test set. It refers to the historical predicted prices using a prediction model.*

$$Snakes = \{Snake_i \mid i \in [0, 9]\ \}$$
$$Snake_i = \{\ \hat{P}_{10i+j} \mid j \in [1, 10]\ \}$$

$$\text{, where } \hat{P}_{100} \text{ is the historical predicted price of today.}$$

Purpose

It involves in the evaluation of the model performance. By comparing the snake with the actual historical prices, we may calculate the accuracy of the model.



Green lines: Snakes

Appendix B - Model Accuracy Score (MAS)

Definition

MAS describes the accuracy of the price prediction regarding the actual price. It is a weighted sum of Model Prediction Score (MPS) and Model Direction Score (MDS), ranging in [0,1].

$$\text{Model Accuracy Score (MAS)} = (1-\alpha)\cdot MPS \ + \ \alpha \cdot MDS$$

$$\text{Model Prediction Score (MPS)} = \frac{1}{|S|}\sum_{i=1}^{|S|} SPS_i$$

$$\text{Snake Prediction Score (SPS)} = \begin{cases} \left(\frac{e-\sigma}{\sigma}\right)^4 & \text{if } e \leq \sigma \\ 0 & \text{otherwise} \end{cases} \text{, where } e = \frac{1}{10}\sum_{i=1}^{10}\left|\frac{\hat{P}_i}{P_i} - 1\right|$$

$$\text{Model Direction Score (MDS)} = \frac{1}{|S|}\sum_{i=1}^{|S|} SDS_i$$

$$\text{Snake Direction Score (SDS)} = \begin{cases} 1 & \text{if } sgn(\hat{P}_{10}-\hat{P}_0) = sgn(P_{10}-P_0) \quad \text{and} \quad |\hat{P}_{10}-\hat{P}_0| \leq |P_{10}-P_0| \\ 0.8 & \text{if } sgn(\hat{P}_{10}-\hat{P}_0) = sgn(P_{10}-P_0) \quad \text{and} \quad |\hat{P}_{10}-\hat{P}_0| > |P_{10}-P_0| \\ 0 & \quad otherwise \end{cases}$$

Accuracy Score	Price Prediction
1.53	↑ +1.68%
0.73	↓ -1.15%
0.36	↓ -4.29%
0.93	↓ -1.25%

Appendix C - Model Prediction Score (MPS)

Definition

MPS is the average of Snake Prediction Scores (SPS). Each SPS is calculated by the prediction error in each of the 10-day disjoint segments,

where the error is basically an average of the absolute relative change between the predicted prices and the actual prices over the 10 days.

$$\text{Model Prediction Score (MPS)} = \frac{1}{|S|}\sum_{i=1}^{|S|} SPS_i$$

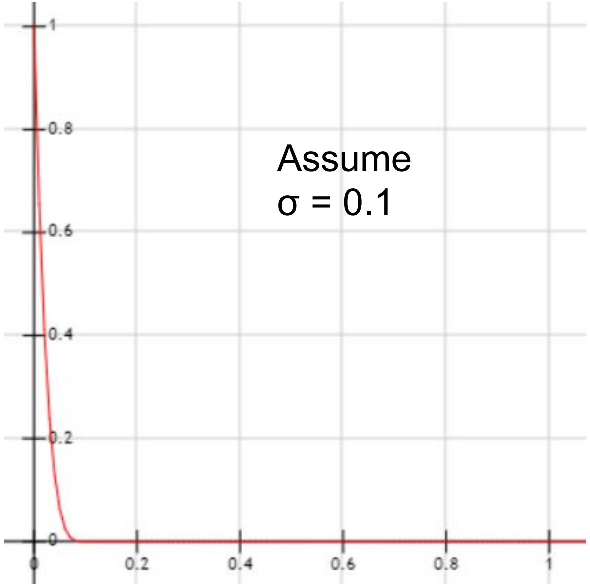
$$\text{Snake Prediction Score (SPS)} = \begin{cases} \left(\frac{e-\sigma}{\sigma}\right)^4 & \text{if } e \leq \sigma \\ 0 & \text{otherwise} \end{cases} \quad \text{, where } e = \frac{1}{10}\sum_{i=1}^{10}\left|\frac{\hat{P}_i}{P_i} - 1\right|$$

Appendix C - Model Prediction Score (MPS)

Definition

SPS is 0 If the error > σ of the stock.

Otherwise, a scoring concave upward function is applied to scale the error to a range of [0,1] based on the standard deviation.



Assume
$\sigma = 0.1$

$$\text{Model Prediction Score (MPS)} = \frac{1}{|S|}\sum_{i=1}^{|S|} SPS_i$$

$$\text{Snake Prediction Score (SPS)} = \begin{cases} \left(\frac{e-\sigma}{\sigma}\right)^4 & \text{if } e \le \sigma \\ 0 & \text{otherwise} \end{cases} \quad , \text{ where } e = \frac{1}{10}\sum_{i=1}^{10}\left|\frac{\hat{P}_i}{P_i} - 1\right|$$

Appendix D - Model Direction Score (MDS)

Definition

MDS is the average of Snake Direction Scores (SDS).

Each SDS is evaluated by the alignment of the prediction direction and the actual direction of the stock trend in each of the 10-day disjoint segments.

$$\text{Model Direction Score (MDS)} = \frac{1}{|S|}\sum_{i=1}^{|S|} SDS_i$$

$$\text{Snake Direction Score (SDS)} = \begin{cases} 1 & \text{if } sgn(\hat{P_{10}} - \hat{P_0}) = sgn(P_{10} - P_0) \quad \text{and} \quad |\hat{P_{10}} - \hat{P_0}| \leq |P_{10} - P_0| \\ 0.8 & \text{if } sgn(\hat{P_{10}} - \hat{P_0}) = sgn(P_{10} - P_0) \quad \text{and} \quad |\hat{P_{10}} - \hat{P_0}| > |P_{10} - P_0| \\ 0 & \qquad otherwise \end{cases}$$

Appendix D - Model Direction Score (MDS)

Definition

SDS is 0 if the prediction has a different direction with the actual direction, as it means the prediction is giving a false trend signal to the users.

Otherwise, SDS would be evaluated based on the direction of the estimation error. In other words, if the prediction is overestimated, SDS is 0.8. Otherwise, it is 1.

Assumed that an underestimated prediction means the model is more reserved and is better off than an overestimating model.

$$\text{Model Direction Score (MDS)} = \frac{1}{|S|}\sum_{i=1}^{|S|} SDS_i$$

$$\text{Snake Direction Score (SDS)} = \begin{cases} 1 & \text{if } sgn(\hat{P_{10}} - \hat{P_0}) = sgn(P_{10} - P_0) \quad \text{and} \quad |\hat{P_{10}} - \hat{P_0}| \leq |P_{10} - P_0| \\ 0.8 & \text{if } sgn(\hat{P_{10}} - \hat{P_0}) = sgn(P_{10} - P_0) \quad \text{and} \quad |\hat{P_{10}} - \hat{P_0}| > |P_{10} - P_0| \\ 0 & \quad otherwise \end{cases}$$

Appendix E - Model Trend Score (MTS)

Definition

MTS describes the correctness of the trend predicted by the models regarding the actual price, ranging in [0,1].

Trend Score ↓	Trend Prediction
7.13	↑
6.73	↓
6.56	↓
5.66	↓

$$\text{Model Trend Score (MTS)} = \frac{1}{10}\sum_{i=1}^{10} TS_i$$

TS is the percentage of having a correct trend prediction of price i days later:

$$\text{Trend Score for i-day predict } (TS_i) = \frac{\sum_{j=1}^{100-i} f(\hat{P}_{i+j} - \hat{P}_j\ ,\ P_{i+j} - P_j)}{100 - i}, \text{ where}$$

$$f(\hat{D}, D) = \begin{cases} 1 & \text{if } sgn(\hat{D}) = sgn(D) \\ 0 & \text{otherwise} \end{cases}$$

Appendix E - Model Trend Score (MTS)

Purpose

Since price prediction accuracy is difficult to obtain, sometimes MAS might not be intuitive.

As a result, instead of observing the exact changes in prices, we could look at the trend of the predictions.

With MTS, the users could gain more accurate insight on the future price change of the stock.

Trend Score ↓	Trend Prediction
7.13	↑
6.73	↓
6.56	↓
5.66	↓

Appendix F - Buy/Sell Score

Definition

It indicates the likeness of the stock going up or down to assist the users in making decisions.

It ranges in [-1, 1], with 1 means expecting an uptrend, -1 means expecting a downtrend, and 0 means the prediction is inconclusive.

$$\text{Buy/Sell Score} = \frac{1}{|M'|}\sum_{i=1}^{|M'|}(MTS_i \cdot TD_i)$$

$$\text{Trend Direction (TD)} = \begin{cases} 1 & \text{if } \left|\{sgn(\hat{P}_i - P_0) = +ve \mid i \in [1, 10]\ \}\right| > 5 \\ 0 & \text{if } \left|\{sgn(\hat{P}_i - P_0) = +ve \mid i \in [1, 10]\ \}\right| = 5 \\ -1 & \text{otherwise} \end{cases}$$

M : set of all models $\quad M' = \{m \mid m \in M,\ MTS_m \geq T\}$ $\quad T$: score threshold

Appendix G - Upper Bounds & Lower Bounds

Definition

Stock prices are volatile in nature and predictions could almost impossibly be 100% accurate.

To give investors more information about how the stock price may fluctuate, upper bounds and lower bounds of prediction error range are also calculated.

$$upper_i = \hat{P}_i + stdev(\hat{P}_j - P_j)$$
$$lower_i = \hat{P}_i - stdev(\hat{P}_j - P_j)$$

$\hat{P}_j$ *is the collection of all i-th day prediction generated by the model in the test set, while*

P_j *is the corresponding collection of actual stock prices.*