Stock Prediction　程序运行手册 MAC

Last edited: 29/12/2020 by Sangdi Gu

Part1: Setting Environment

Install Anaconda:

Go to Anaconda installation page: https://www.anaconda.com/distribution/#download-section
Follow the instructions on the following link to install Anaconda:
https://docs.anaconda.com/anaconda/install/
One reason we recommend using Anaconda is that you can create virtual environments for your demand. Each virtual environment will not affect each other.

Set virtual environment for CPU-only TensorFlow:

Run the following commands in terminal:

```
$ conda create -n tf_cpu tensorflow python=x.x (python part is optional)
$ conda activate tf_cpu
```

For deactivate your virtual environment:

```
$ conda deactivate
```

Install required Modules:

In Anaconda and your virtual environment, using "conda install" to install all the package. (Modules includes: matplotlib, pandas, PIL, keras, sklearn, alpha_vantage)

```
$ conda install matplotlib
$ conda install python=3.6.4 // older version of python is necessary for
installing older versions of pandas and tensorflow
$ conda install tensorflow=1.15.0
$ conda install pandas=0.24 // the newest version is not compatible
$ conda install pillow
$ conda install keras
$ conda install scikit-learn
$ conda install requests
$ pip install alpha_vantage
```

Part 2: Running the Program

Run the Program file:

Use the following commands in the terminal to run the main program file.
Change directory to file path :

```
$ cd <file dir>
```

Run file:

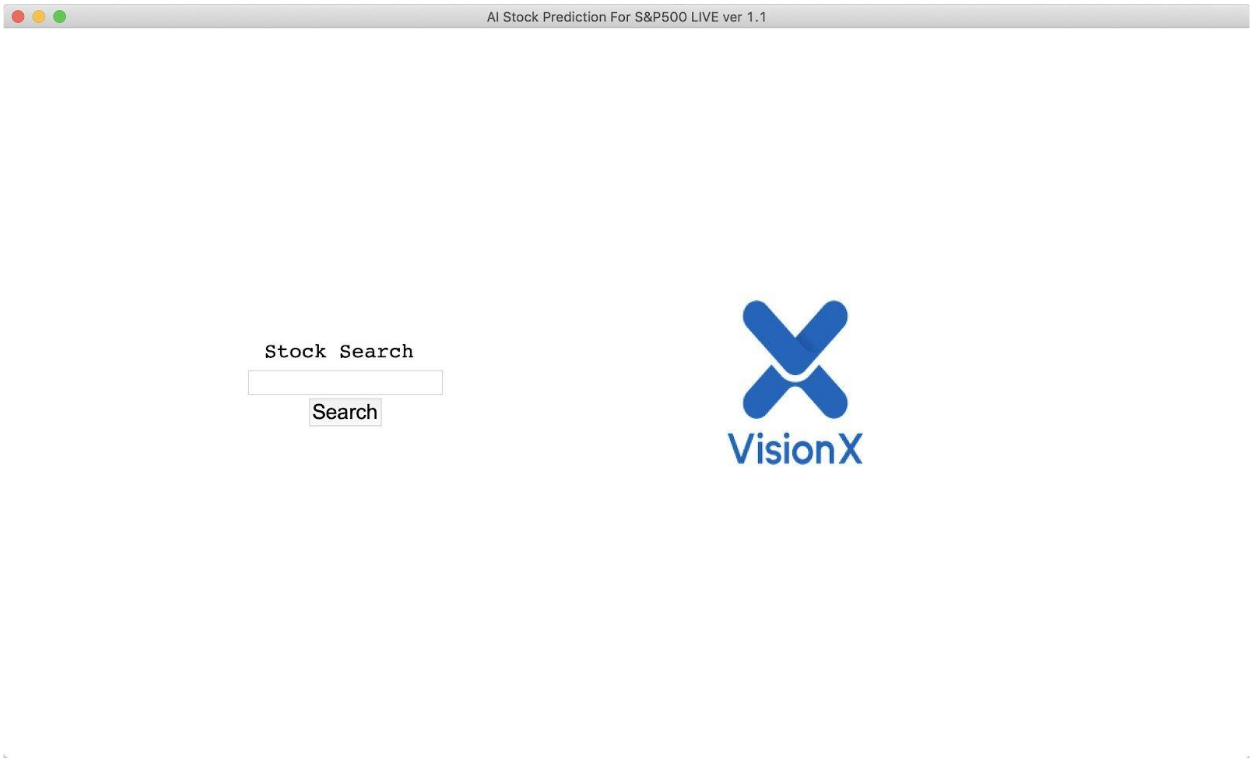
```
$ python3 <file dir>
```

Example:

```
$ cd /Users/jing/Desktop/Version1.1/
$ python3 Stock_Prediction_LIVE_ver1.1.py
```

Show User Interface:

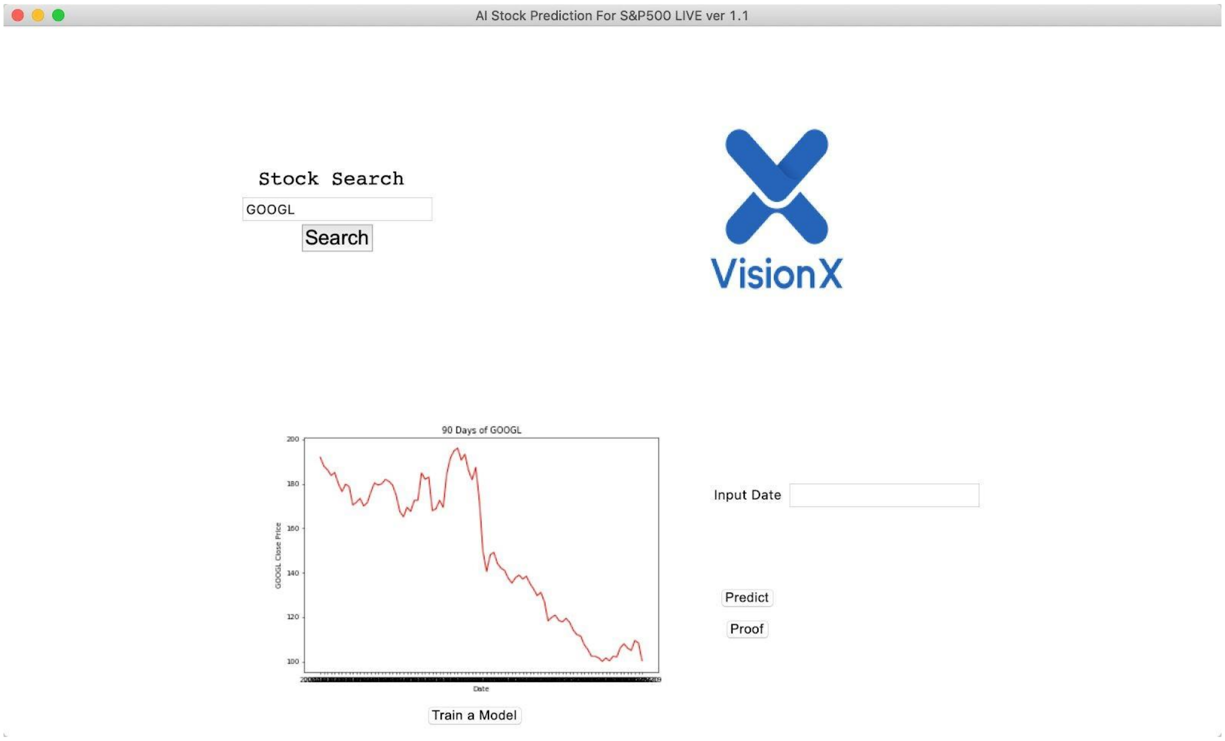
The interface contains a search bar for stock input.



Input Stock Symbol:

Example : “AAPL” for apple, “GOOGL” for google, “AMZN” for amazon.

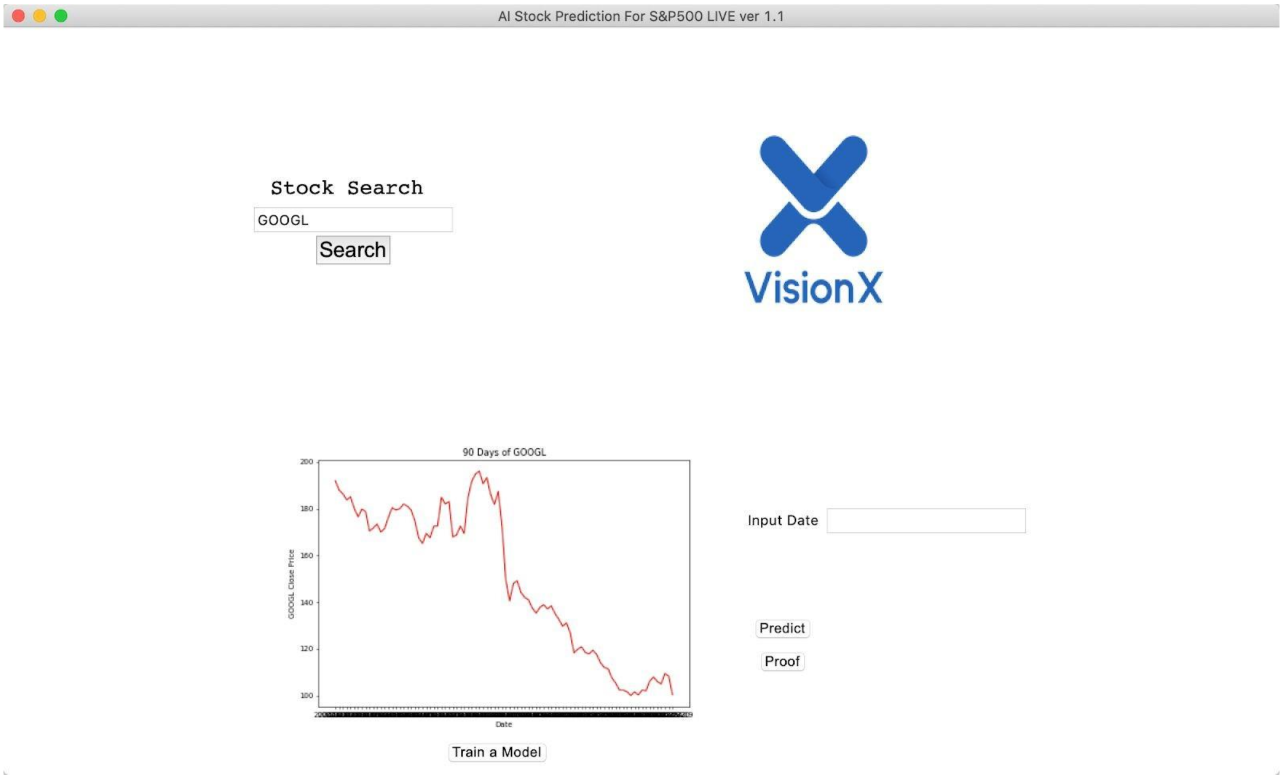
Output: plot of 90- days stock price data.



Stock Search: GOOGL

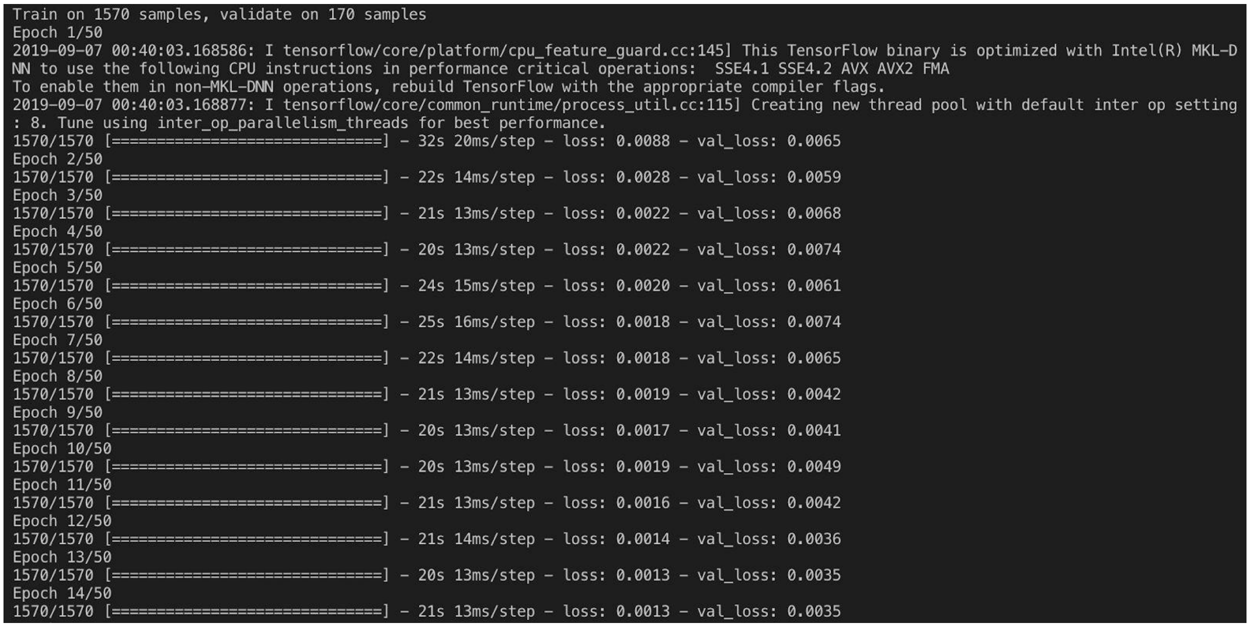
Train the Model:

Click “train the model” to train current model using stock data.



Training Progress:

The training progress takes about 10 minutes.

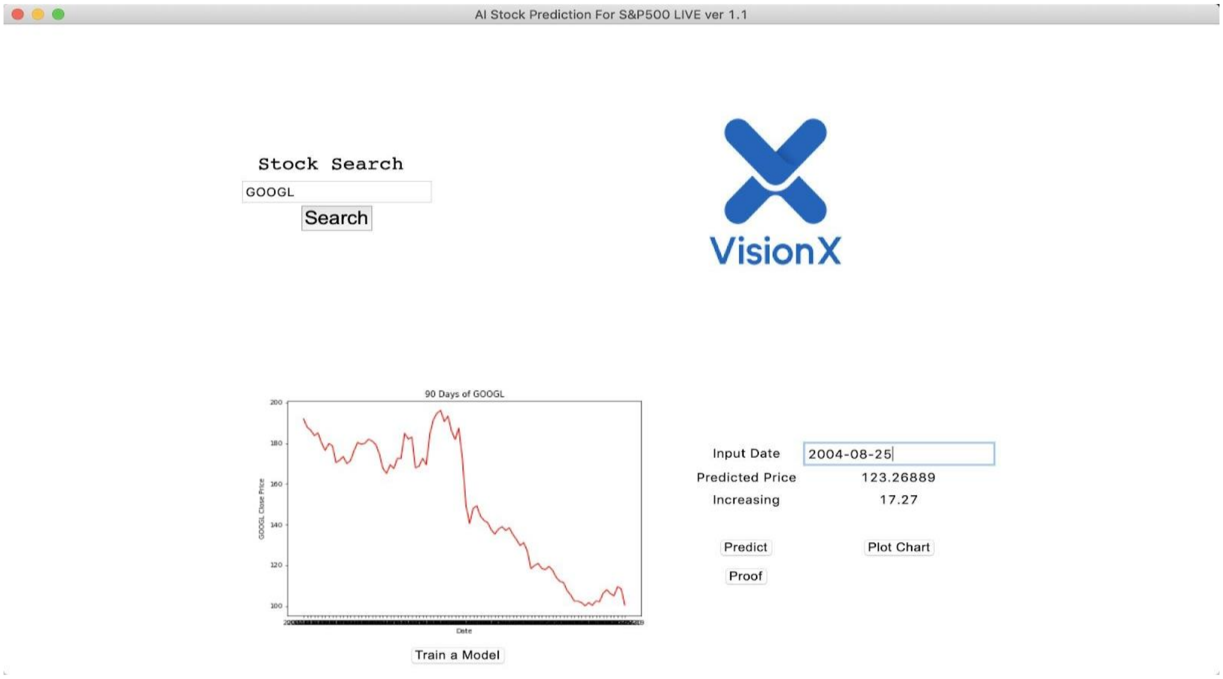


```
Train on 1570 samples, validate on 170 samples
Epoch 1/50
2019-09-07 00:40:03.168586: I tensorflow/core/platform/cpu_feature_guard.cc:145] This TensorFlow binary is optimized with Intel(R) MKL-D
NN to use the following CPU instructions in performance critical operations:  SSE4.1 SSE4.2 AVX AVX2 FMA
To enable them in non-MKL-DNN operations, rebuild TensorFlow with the appropriate compiler flags.
2019-09-07 00:40:03.168877: I tensorflow/core/common_runtime/process_util.cc:115] Creating new thread pool with default inter op setting
: 8. Tune using inter_op_parallelism_threads for best performance.
1570/1570 [==============================] - 32s 20ms/step - loss: 0.0088 - val_loss: 0.0065
Epoch 2/50
1570/1570 [==============================] - 22s 14ms/step - loss: 0.0028 - val_loss: 0.0059
Epoch 3/50
1570/1570 [==============================] - 21s 13ms/step - loss: 0.0022 - val_loss: 0.0068
Epoch 4/50
1570/1570 [==============================] - 20s 13ms/step - loss: 0.0022 - val_loss: 0.0074
Epoch 5/50
1570/1570 [==============================] - 24s 15ms/step - loss: 0.0020 - val_loss: 0.0061
Epoch 6/50
1570/1570 [==============================] - 25s 16ms/step - loss: 0.0018 - val_loss: 0.0074
Epoch 7/50
1570/1570 [==============================] - 22s 14ms/step - loss: 0.0018 - val_loss: 0.0065
Epoch 8/50
1570/1570 [==============================] - 21s 13ms/step - loss: 0.0019 - val_loss: 0.0042
Epoch 9/50
1570/1570 [==============================] - 20s 13ms/step - loss: 0.0017 - val_loss: 0.0041
Epoch 10/50
1570/1570 [==============================] - 20s 13ms/step - loss: 0.0019 - val_loss: 0.0049
Epoch 11/50
1570/1570 [==============================] - 21s 13ms/step - loss: 0.0016 - val_loss: 0.0042
Epoch 12/50
1570/1570 [==============================] - 21s 14ms/step - loss: 0.0014 - val_loss: 0.0036
Epoch 13/50
1570/1570 [==============================] - 20s 13ms/step - loss: 0.0013 - val_loss: 0.0035
Epoch 14/50
1570/1570 [==============================] - 21s 13ms/step - loss: 0.0013 - val_loss: 0.0035
```

Input Predict Date and Click “Predict”:

Input Date should follow the format: %Y-%m-%d (eg. 2019-09-05), and weekends and holidays are invalid date. The predicted value corresponds to the closing price of the next working date relative to the input date.

Prediction can only be made for the next unknown closing price. Predictions further into the future is not allowed due to poor accuracy using the current model. The current model is a simple multi-layer LSTM.



AI Stock Prediction For S&P500 LIVE ver 1.1

Stock Search: GOOGL

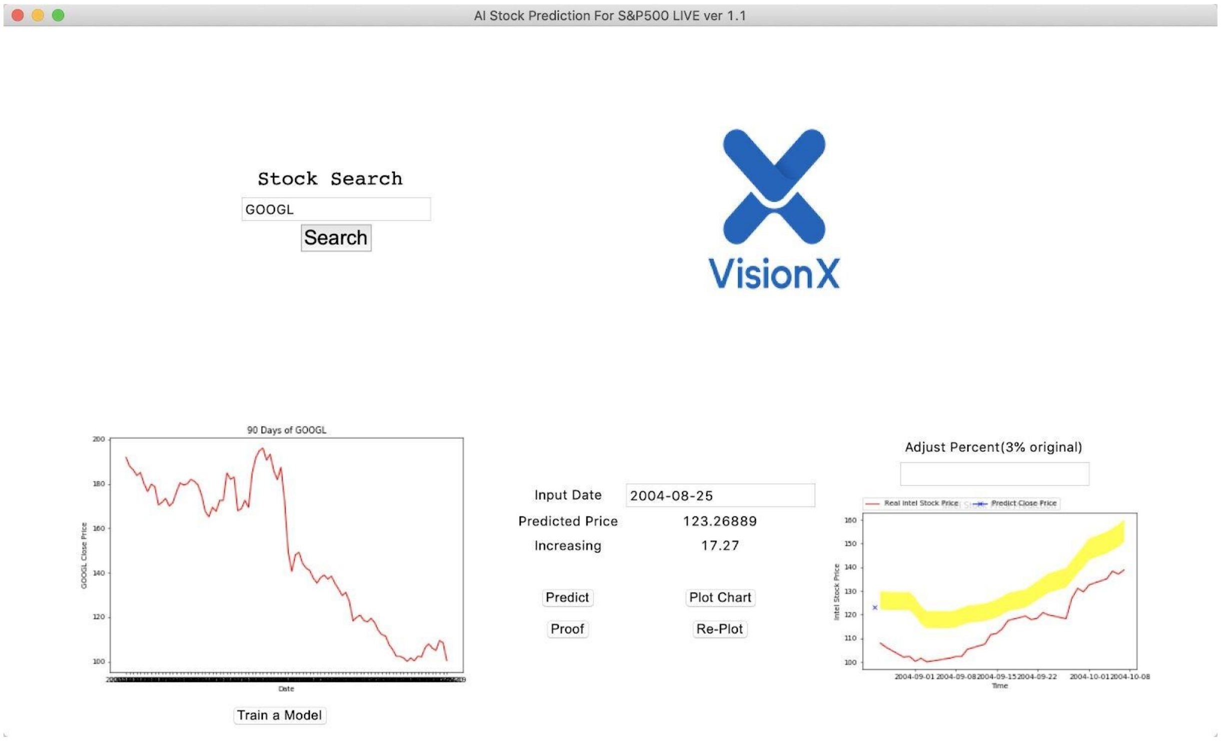
Input Date: 2004-08-25

Predicted Price: 123.26889

Increasing: 17.27

Click “Plot Chart” to Plot Chart:

The output chart indicates predicted trends.



AI Stock Prediction For S&P500 LIVE ver 1.1

Stock Search: GOOGL

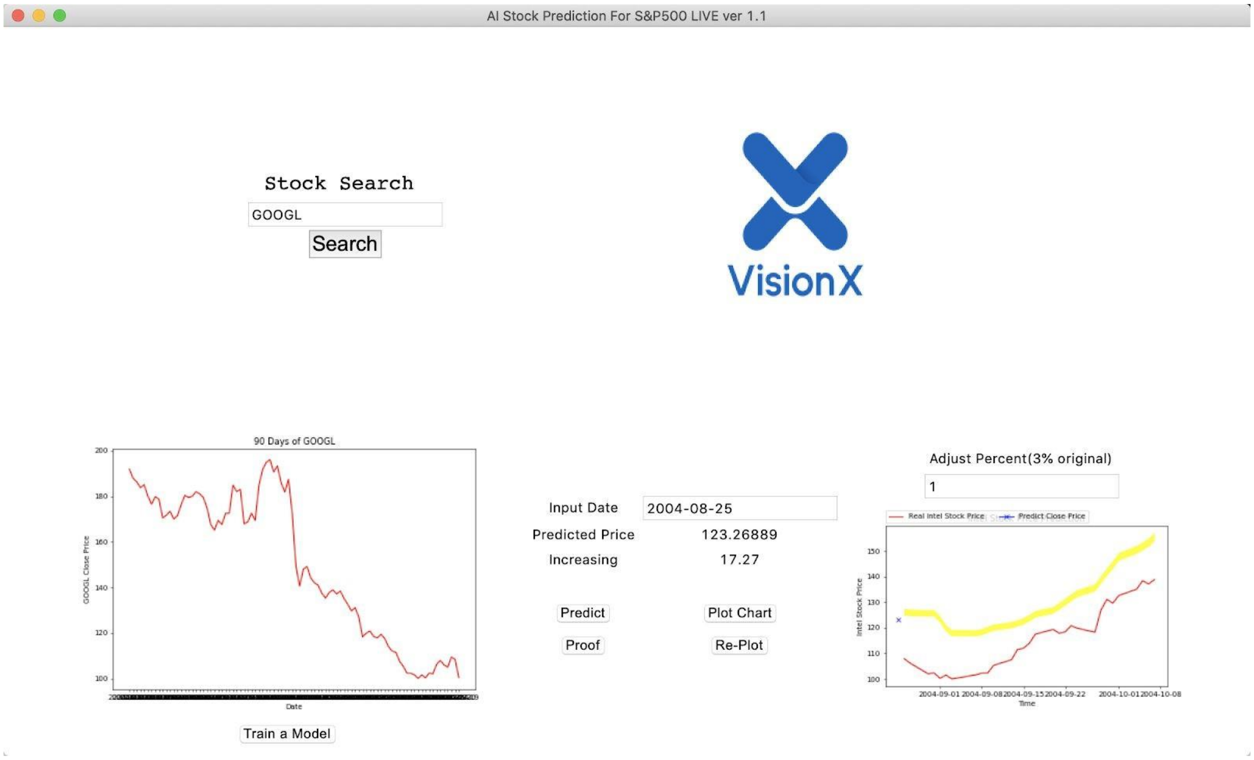
Input Date: 2004-08-25

Predicted Price: 123.26889

Increasing: 17.27

Adjust Percent(3% original):

Adjust Percent and Click “Re-Plot”:



AI Stock Prediction For S&P500 LIVE ver 1.1

Stock Search

GOOGL

Input Date 2004-08-25

Predicted Price 123.26889

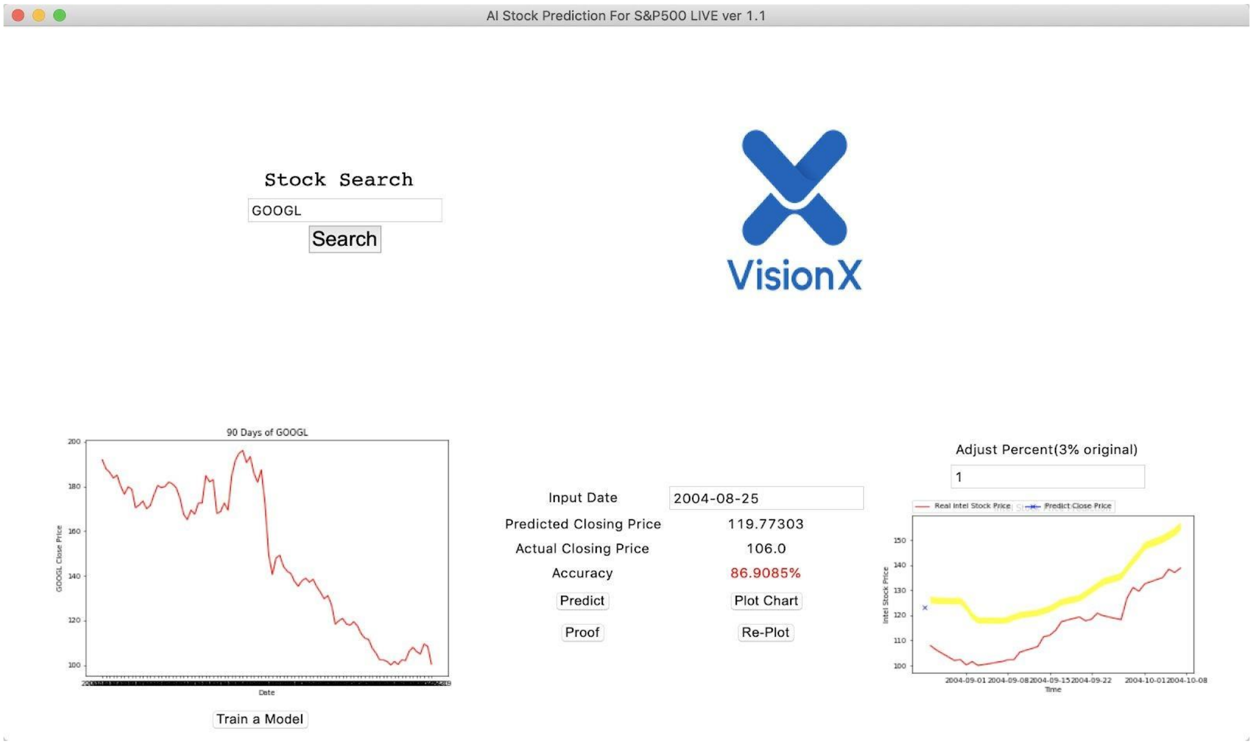
Increasing 17.27

Adjust Percent(3% original)

1

Click Proof and Show Proof:

Show actual closing price and accuracy. The date of the closing price corresponds to the input date.



AI Stock Prediction For S&P500 LIVE ver 1.1

Stock Search

GOOGL

Input Date 2004-08-25

Predicted Closing Price 119.77303

Actual Closing Price 106.0

Accuracy 86.9085%

Adjust Percent(3% original)

1