(https://www.quantinsti.com)

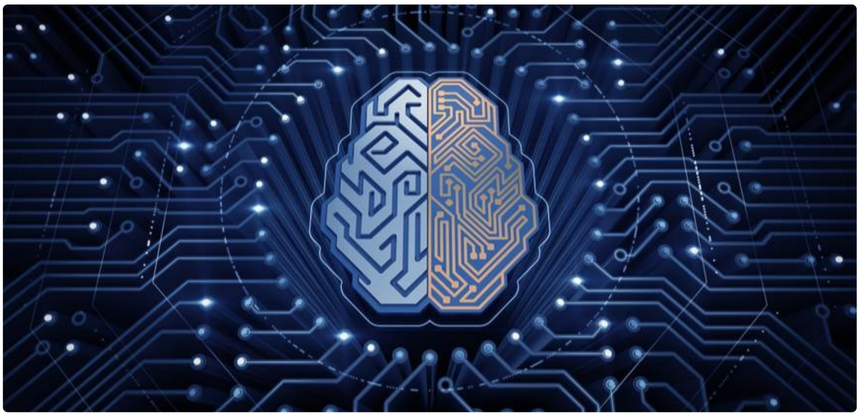
MAIN NAVIGATION

Home (https://www.quantinsti.com) > Blog (https://www.quantinsti.com/blog) > Trading Strategies (https://www.quantinsti.com/category/quantitative-trading-strategies/)

Machine Learning Classification Strategy In Python ()

On December 18, 2017 ▪ 0 Comment (https://www.quantinsti.com/blog/machine-learning-classification-strategy-python/#respond)

G+ (h | 1 | 6 | in 14 | (htt | 21 SHARES



By Ishan Shah (https://www.linkedin.com/in/ishan-shah-18393828/)

In this blog, we will step by step implement a machine learning classification (https://quantra.quantinsti.com/course/trading-machine-learning-classification-svm) algorithm on S&P500 using Support Vector Classifier (SVC). SVCs are supervised learning classification models. A set of training data is provided to the machine learning classification algorithm, each belonging to one of the categories. For instance, the categories can be to either buy or sell a stock. The classification algorithm builds a model based on the training data and then, classifies the test data into one of the categories.

> *Machine Learning Classification Strategy In Python*
>
> CLICK TO TWEET

(https://twitter.com/share?text=Machine+Learning+Classification+Strategy+In+Python+%40QuantInsti&url=https://www.quantinsti.com/blog/machine-learning-classification-strategy-python/)

Now, let's implement the machine learning classification strategy in Python.

Step 1: Import the libraries

In this step, we will import the necessary libraries that will be needed to create the strategy.

# machine learning classification

- from sklearn.svm import SVC
- from sklearn.metrics import scorer
- from sklearn.metrics import accuracy_score

# For data manipulation

- import pandas as pd
- import numpy as np

# To plot

- import matplotlib.pyplot as plt
- import seaborn

# To fetch data

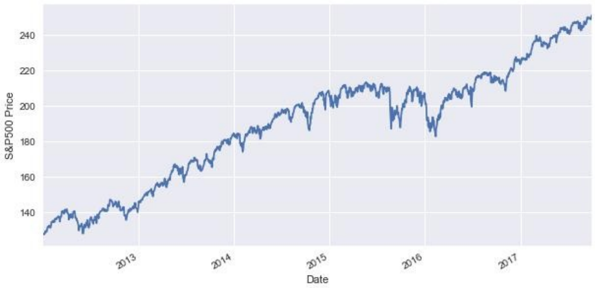
- from pandas_datareader import data as pdr

Step 2: Fetch data

We will download the S&P500 data from google finance using pandas_datareader.

After that, we will drop the missing values from the data and plot the S&P500 close price series.

```
Df = pdr.get_data_google('SPY', start="2012-01-01", end="2017-10-01")

Df= Df.dropna()

Df.Close.plot(figsize=(10,5))

plt.ylabel("S&P500 Price")

plt.show()
```



Step 3: Determine the target variable

The target variable is the variable which themachine learning (https://www.quantinsti.com/blog/overview-machine-learning-trading/) classification algorithm will predict. In this example, the target variable is whether S&P500 price will close up or close down on the next trading day.

We will first determine the actual trading signal using the following logic – if next trading day's close price is greater than today's close price then, we will buy the S&P500 index, else we will sell the S&P500 index. We will store +1 for the buy signal and -1 for the sell signal.

```
y = np.where(Df['Close'].shift(-1) > Df['Close'],1,-1)
```

Step 4: Creation of predictors variables

The X is a dataset that holds the predictor's variables which are used to predict target variable, 'y'. The X consists of variables such as *'Open – Close'* and *'High – Low'*. These can be understood as indicators based on which the algorithm will predict the option price.

```
Df['Open-Close'] = Df.Open - Df.Close

Df['High-Low'] = Df.High - Df.Low

X=Df[['Open-Close','High-Low']]
```

In the later part of the code, themachine learning (https://www.quantinsti.com/blog/machine-learning-trading-predict-stock-prices-regression/) classification algorithm will use the predictors and target variable in the training phase to create the model and then, predict the target variable in the test dataset.

Step 5: Test and train dataset split

In this step, we will split data into the train dataset and the test dataset.

1. First, 80% of data is used for training and remaining data for testing
2. X_train and y_train are train dataset
3. X_test and y_test are test dataset

```
split_percentage = 0.8

split = int(split_percentage*len(Df))
```

```
# Train data set

X_train = X[:split]

y_train = y[:split]
```

```
# Test data set

X_test = X[split:]

y_test = y[split:]
```

Step 6: Create the machine learning classification model using the train dataset

We will create the machine learning classification model based on the train dataset. This model will be later used to predict the trading signal in the test dataset.

```
cls = SVC().fit(X_train, y_train)
```

Step 7: The classification model accuracy

We will compute the accuracy of the classification model on the train and test dataset, by comparing the actual values of the trading signal with the predicted values of the trading signal. The function accuracy_score() will be used to calculate the accuracy.

Syntax: accuracy_score(target_actual_value,target_predicted_value)

1. target_actual_value: correct signal values
2. target_predicted_value: predicted signal values

```
accuracy_train = accuracy_score(y_train, cls.predict(X_train))

accuracy_test = accuracy_score(y_test, cls.predict(X_test))
```

```
print('\nTrain Accuracy:{: .2f}%'.format(accuracy_train*100))

print('Test Accuracy:{: .2f}%'.format(accuracy_test*100))
```

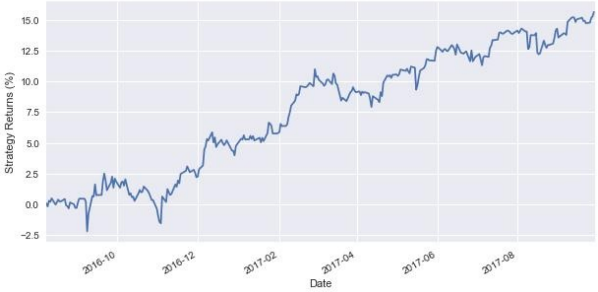
An accuracy of 50%+ in test data suggests that the classification model is effective.

Step 8: Prediction

We will predict the signal (buy or sell) for the test data set, using the cls.predict() function. Then, we will compute the strategy returns based on the signal predicted by the model in the test dataset. We save it in the column 'Strategy_Return' and then, plot the cumulative strategy returns.

Df['Predicted_Signal'] = cls.predict(X)

```
# Calculate log returns

Df['Return'] = np.log(Df.Close.shift(-1) / Df.Close)*100

Df['Strategy_Return'] = Df.Return * Df.Predicted_Signal

Df.Strategy_Return.iloc[split:].cumsum().plot(figsize=(10,5))

plt.ylabel("Strategy Returns (%)")

plt.show()
```



As seen from the graph, the machine learning classification strategy generates a return of around 15% in the test data set.

Next Step

If you want to learn various aspects of Algorithmic trading (https://www.quantinsti.com/) then check out the Executive Programme in Algorithmic Trading (EPAT™) (https://www.quantinsti.com/epat/). The course covers training modules like Statistics & Econometrics, Financial Computing & Technology, and Algorithmic & Quantitative Trading. EPAT™ equips you with the required skill sets to be a successful trader. Enroll now (https://www.quantinsti.com/epat/)!

Related Posts:

Build Technical Indicators in Python

Connecting FXCM over FIX – A detailed Tutorial

Trading Using Machine Learning In Python
SVM (Support Vector Machine)
Trading Using Machine Learning In Python – SVM

Development of Cloud-Based Automated Trading System

(https://www.quantinsti.com/blog/epat-project-

G+ (h | 1 | 6 | 14 | (htt | 21 SHARES

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

reCAPTCHA

Please upgrade to a supported browser to get a reCAPTCHA challenge.

Alternatively if you think you are getting this page in error, please check your internet connection and reload.

Why is this happening to me?

Search for posts

Search

Privacy - Terms

WANT TO LEARN ALGO TRADING?

Name *

Full Name

Email *

Your Email ID

Phone *

Your Phone Number

Country *

Singapore

Are you human?

3 T2XW

I'm Interested

(https://www.quantinsti.com/guest-post/)

CATEGORIES

Career Advice (https://www.quantinsti.com/category/career-advice/) (16)

Case Studies (https://www.quantinsti.com/category/case-studies/) (6)

Downloadables (https://www.quantinsti.com/category/downloadables/) (22)

Getting Started (https://www.quantinsti.com/category/quantitative-trading-getting-started/) (112)

Guest Posts (https://www.quantinsti.com/category/guest-posts/) (2)

News (https://www.quantinsti.com/category/news/) (46)

- Events (https://www.quantinsti.com/category/news/events-news/) (30)

Press Releases (https://www.quantinsti.com/category/press-releases/) (3)

Programming and Trading Tools (https://www.quantinsti.com/category/programming-and-trading-tools/) (92)

- Other Languages (https://www.quantinsti.com/category/programming-and-trading-tools/other-languages/) (10)
- Python (https://www.quantinsti.com/category/programming-and-trading-tools/python/) (38)
- R Programming (https://www.quantinsti.com/category/programming-and-trading-tools/r-programming/) (37)
- Trading Platforms (https://www.quantinsti.com/category/programming-and-trading-tools/trading-platforms/) (6)

Project Work EPAT (https://www.quantinsti.com/category/project-work-epat/) (14)

Trading Strategies (https://www.quantinsti.com/category/quantitative-trading-strategies/) (70)

Webinars (https://www.quantinsti.com/category/events/) (32)

- Previous Webinars (https://www.quantinsti.com/category/events/previous/) (32)

HELPFUL SOURCES

Quantocracy (http://quantocracy.com/?utm_source=quantinsti.com)

Quantsportal (http://www.quantsportal.com/?utm_source=quantinsti.com)

Quantpedia (http://quantpedia.com/?utm_source=quantinsti.com)

KDnuggets (http://www.kdnuggets.com/?utm_source=quantinsti.com)

R-bloggers (http://www.r-bloggers.com/?utm_source=quantinsti.com)

The Financial Hacker (http://www.financial-hacker.com/?utm_source=quantinsti.com)

Wall Street Oasis (http://www.wallstreetoasis.com/?utm_source=quantinsti.com)

Turing Finance (http://www.turingfinance.com/?utm_source=quantinsti.com)

Quick Links

About Us (https://www.quantinsti.com/about-us/)
EPAT™ (https://www.quantinsti.com/epat/)
Alumni (https://www.quantinsti.com/alumni/)
Testimonials (https://www.quantinsti.com/testimonials/)
Faculty (https://www.quantinsti.com/faculty/)
Careers (https://www.quantinsti.com/careers/)
Quantra™ (https://quantra.quantinsti.com/)
Placement Cell (https://www.quantinsti.com/placement-cell/)
BTRM (https://www.quantinsti.com/btrm)
Contact Us (https://www.quantinsti.com/contact-us/)
Disclaimer Policy (https://www.quantinsti.com/disclaimer-policy/)
Privacy Policy (https://www.quantinsti.com/privacy-policy/)
Terms & Conditions (https://www.quantinsti.com/terms-conditions/)
Pay Your Fees (https://www.quantinsti.com/payment/)
Student Login (http://quantinsti.talentlms.com/)

India
QuantInsti Quantitative Learning Pvt Ltd
A-309, Boomerang,
Chandivali Farm Road, Powai,
Mumbai – 400 072
Toll Free: 1800-266-5401
Phone: +91-22-61691400

Singapore
30 Cecil Street, #19-08,
Prudential Tower,
Singapore – 049712
Phone: +65-9057-8301

Connect with us...

(https://www.facebook.com/quantinsti) (https://twitter.com/quantinsti/) (https://www.linkedin.com/company/quantinsti) (https://plus.google.com/110772715968756646442/) (https://www.youtube.com/user/quantinsti)

QUANTOCRACY
POWER TO THE QUANTS

(http://quantocracy.com/)Show us some love on Quantocracy.