

A background image showing a person's hands typing on a laptop keyboard. The person is wearing a blue and white checkered shirt. The laptop screen displays some data or charts, though they are out of focus. The overall scene is a blurred office environment.

Predicting Stock Direction

AGENDA

1. A Brief Overview of Stock Market.
2. Getting Familiar with Stock Market Data.
3. What does Prediction of Stock Direction mean?
4. How can we achieve this? (Methodology).
5. Pseudocode for stock direction prediction...
 - a. Import required Libraries
 - b. Get the data using pandas_datareader
 - c. Extract the price column and scale the column
 - d. Split the data in training and testing
 - e. Create Features (Feature Engineering)
 - f. Fit the model.
 - g. Test the model and plot the results.
6. Task for you.

Overview of Stock market

1. Every corporation need finance to run their operations.
2. Some finance need can be accomplished by Capital and some by taking Loan from banks.
3. But, still they need more. What they can do?
4. They can issue shares of their company for public to purchase.
5. When someone purchases the share (stock) they will have some part in the business profits too.
6. Now, there are many organisations who trade shares with public. How is the price of their stock is decided?
7. Yes, you guessed it right, It is decided by Stock Market where each corporation takes part in Stock Trading and they get the best price according to demand and supply equilibrium.

Stock Market Data

Let us take a corporation called A.

A let their shares be purchased or sold in stock market. When they do this, on a daily basis, for A's stock various values are generated which are important indicators for someone who is looking for investing in the stock of A. Those indicators are:

1. Highest Price in a Day (High)
2. Lowest Price in a Day (Low)
3. Volume Traded in a Day (Volume)
4. Opening Price (OPEN)
5. Closing Price (Close)
6. Adjusted Close

Stock Prediction Objective

Essentially, if we are today trading in a particular stock say A, we would be interested in knowing what will be the price of A tomorrow.

This is what precisely what we are trying to do in this video.. To predict whether the price of the stock will go up or down.

Methodology

The stock data is a sequential data (indexed by time) and we would like to know what will be the next term in this sequence of price. Here is what we are going to do....

1. For every price X_t , we will take a look at previous n number of prices as a feature to predict this X_t . So, we will essentially take $X_{t-1}, X_{t-2}, \dots, X_{t-60}$ to predict the price X_t .
2. In this way, we will make a training data where we will consider each price as dependent variable and corresponding past 60 values as independent variable and we will try to fit some models on this data.

Pseudocode for this task

Importing Required Libraries

```
# Import Pandas
# Import Matplotlib for plotting
# Import pandas_datareader
# Import StandardScaler import sklearn
# Import Some models for predicting continuous output
# Like LinearRegression, DecisionTreeRegressor etc
# from sklearn
```

We will check many different models **and** see which one **is** best

Get data using pandas_datareader

```
# Specify a symbol or ticker for a corporation
symbol = ""
# Specify the start date for the data
start_date = ""
# Specify the end date for the data
end_date = ""

# Get the data by passing these three info in
# pandas_datareader.get_data_yahoo()
```

Scaling

```
# Get the price column from the data
close = # Close Price column

# Scale the data
scale = StandardScaler()
scale.fit(close)

# Transform the data

new_close = scale.transform()
```

Get the train and test data

```
# First specify the training length
training_len = ""

# Then, split the scaled closing price
# in train and test data

train_data = ""
test_data = ""
```

Create Features

```
# Intialize empty X_train and y_train
X_train = []
y_train = []

# Run a loop from 60 to the end of the train_data
# append the last 60 values to X_train and current
# value to the y_train.

# Convert the X_train and y_train into numpy arrays
```

Fit and Test the model

Now, use the train test to fit as many as different models you can.

Test each model's score and measure the performance.

Task for you

We have just implemented various algorithms, and measured score for various score for a specific stock.

Now, your task is to repeat the same sequence of steps for many other stocks say Microsoft, Facebook, Amazon etc.



Thank you