Computing Strategy Returns

```
trade_dataset['Tomorrows Returns'] = 0.
trade_dataset['Tomorrows Returns'] =
np.log(trade_dataset['Close']/trade_dataset['Close'].shift(1))
trade_dataset['Tomorrows Returns'] = trade_dataset['Tomorrows Returns'].shift(-1)
```

Now that we have the predicted values of the stock movement. We can compute the returns of the strategy. We will be taking a long position when the predicted value of y is true and will take a short position when the predicted signal is False.

We first compute the returns that the strategy will earn if a long position is taken at the end of today, and squared off at the end of the next day. We start by creating a new column named 'Tomorrows Returns' in the trade_dataset and store in it a value of 0. We use the decimal notation to indicate that floating point values will be stored in this new column. Next, we store in it the log returns of today, i.e. logarithm of the closing price of today divided by the closing price of yesterday. Next, we shift these values upwards by one element so that tomorrow's returns are stored against the prices of today.

```
trade_dataset['Strategy Returns'] = 0.
trade_dataset['Strategy Returns'] = np.where(trade_dataset['y_pred'] == True,
trade_dataset['Tomorrows Returns'], - trade_dataset['Tomorrows Returns'])
```

Next, we will compute the Strategy Returns. We create a new column under the header 'StrategyReturns' and initialize it with a value of 0. to indicate storing floating point values. By using the np.where() function, we then store the value in the column 'Tomorrows Returns' if the value in the 'ypred' column stores True (a long position), else we would store negative of the value in the column 'Tomorrows Returns' (a short position); into the 'Strategy Returns' column.

```
trade_dataset['Cumulative Market Returns'] = np.cumsum(trade_dataset['Tomorrows
Returns'])
trade_dataset['Cumulative Strategy Returns'] = np.cumsum(trade_dataset['Strategy
Returns'])
```

We now compute the cumulative returns for both the market and the strategy. These values are computed using the cumsum() function. We will use the cumulative sum to plot the graph of market and strategy returns in the last step.

Plotting the graph of returns

```
import matplotlib.pyplot as plt
plt.figure(figsize=(10,5))
plt.plot(trade_dataset['Cumulative Market Returns'], color='r', label='Market
Returns')
plt.plot(trade_dataset['Cumulative Strategy Returns'], color='g', label='Strategy
Returns')
plt.legend()
plt.show()
```