Prediction of Daily Stock Returns

The purpose of this report is to explain the thought process and methodology for predicting daily stock returns. This project uses historical stock data to predict future price movements and calculate daily percent returns.

Historical stock data was gathered from Yahoo Finance using the yfinance Python library. The yfinance library is an open-source tool, not affiliated with Yahoo, used to download market data from the Yahoo Finance API.

From yfinance we downloaded Daily Open, High, Low, Close, Adjusted Close , and Volume for SPY from 01Jan2009 – 29Oct2021. Data was stored in a Pandas dataframe.

Daily Price range was calculated as the difference between High – Low prices for the day and, was added to the dataframe. Differences between daily adjusted closing prices and daily high/low was also added to the dataframe. The daily percent change in Adjusted Closing price was calculated using the Pandas pct_change function and also added to the dataframe.

The Technical Analysis Library, TA-Lib, was used to perform various technical analyses on historical daily adjusted close SPY prices.

The technical indicators used in this project include: Simple 10 day Moving Average, 9 pt Relative Strength Index, 10 pt Up/Middle/Down Bollinger Bands. Differences between the current day's adjusted close and individual Bollinger Bands were calculated and added to the dataframe.

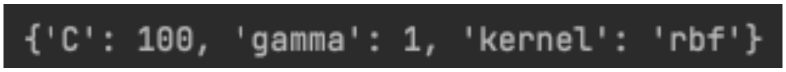
A '*Signal*'' column was added to the dataframe in which daily price percent changes greater than 2% were characterized by 1, all other changes were denoted by a 0.

Because price is specific to the security being evaluated, a numpy array, X, was created from the existing dataframe – dropping asset specific data (Date, Open, High, Low, Close, etc.). The information included in the numpy array aimed to capture price movement characterized by a change in the difference between daily adjusted close and the respective Moving Averages, Bollinger Bands.

Data processing consisted of min-max normalizing and splitting data into Test/Train/Development.

The sklearn gridsearch function was used to find best parameters for training a support vector machine.

The resulting parameters are shown below:

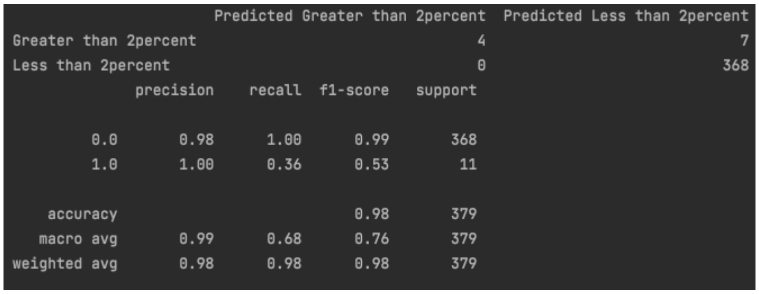


```
{'C': 100, 'gamma': 1, 'kernel': 'rbf'}
```

Figure 1. GridSearch optimal parameters

Using the optimal parameters provided from sklearn GridSearch, a Support Vector Machine was trained to predict daily price movement greater than 2%.

Given the somewhat random nature of financial markets, the model preformed better than expected when predicting 2% jumps in price from day to day. The resulting confusion matrix is shown below:



```
                      Predicted Greater than 2percent  Predicted Less than 2percent
Greater than 2percent                                4                             7
Less than 2percent                                   0                           368
              precision    recall  f1-score   support

         0.0       0.98      1.00      0.99       368
         1.0       1.00      0.36      0.53        11

    accuracy                           0.98       379
   macro avg       0.99      0.68      0.76       379
weighted avg       0.98      0.98      0.98       379
```

Figure 2. Confusion Matrix

Overall the support vector machine did quite a good job at predicting future price movement based on differences in current adjusted closing prices and different technical indicators. Although the results are promising, more work should be done to backtest investment strategies based on ML as the market history is not always the best predictor of future price movement.