Supervised Learning: Market Prediction

Libraries

- Numpy
- Pandas
- Pandas_datareader
- Matplotlib
- Sklearn
  - Svm
  - Ensemble
  - Linear_model
  - Preprocessing
  - Model_selection

Steps

- Collect data on major country indexes
- Review/Model data
- Prep Data
- Choose a Models
- Train the Models
- Evaluate the Models
- Make Predictions

What is the S&P500 & index funds in general? --

```
S&P500:
   market-capitalization-weighted index of the 500 largest U.S. publicly traded companies.
   Strong gauge of large-cap U.S. equities.
```

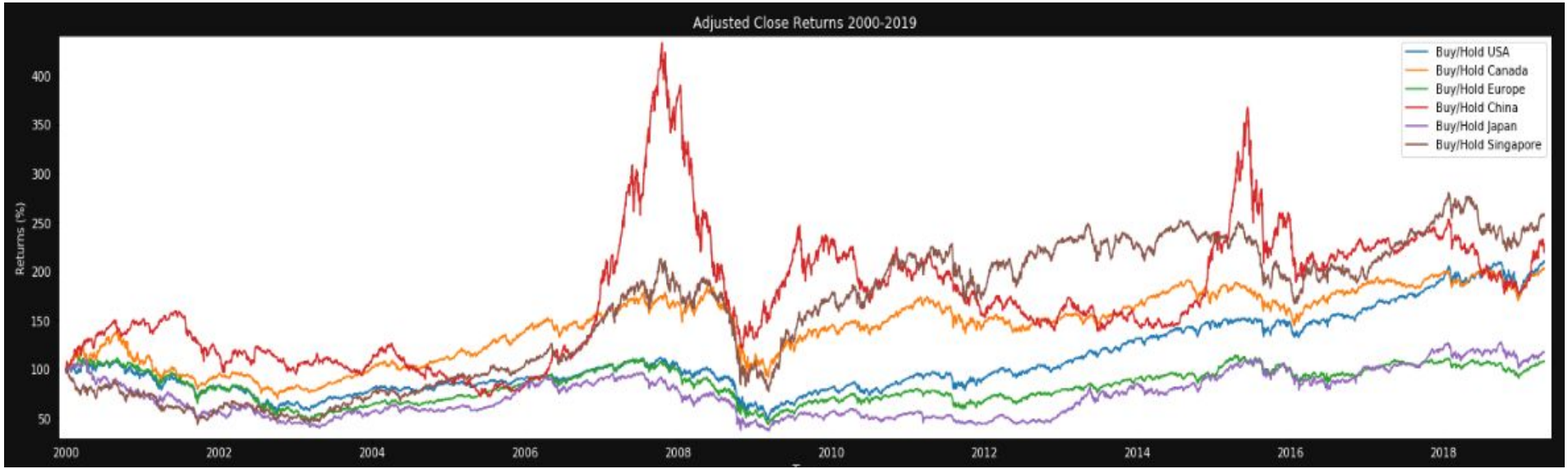
- market-capitalization-weighted in S&P $= \frac{\text{Company market cap}}{\text{Total S\&P market cap}}$
- Company market cap = current stock price * company's outstanding shares
- Most index funds are calculated in a similar manner

```
- Selected indexes:
    - S&P/TSX Composite index    (^GSPTSE)     - Canada      (CAD)                          1 CAD  =  .77   USD
    - S&P 500                    (^GSPC)       - USA         (USD)                          1 USD  = 1.00   USD
    - Euro Stoxx 50              (STOXX: SX5E) - Eurozone    (Euro)                         1 Euro = 1.12   USD
    - SSE Composite Index        (000001.SS)   - Shanghai    (CNY)     Chinese Yuan Renminbi 1 CNY  =  .15   USD
    - Nikkei 225                 (^N225)       - Japan       (JPY)     Japanese Yen          1 JPY  =  .0093 USD
    - MSCI Singapore Capped ETF  (EWS)         - Singapore   (SGD)     Singapore Dollar      1 SGD  =  .74   USD
```

Graph of adjusted close prices from 2000 to 2019:

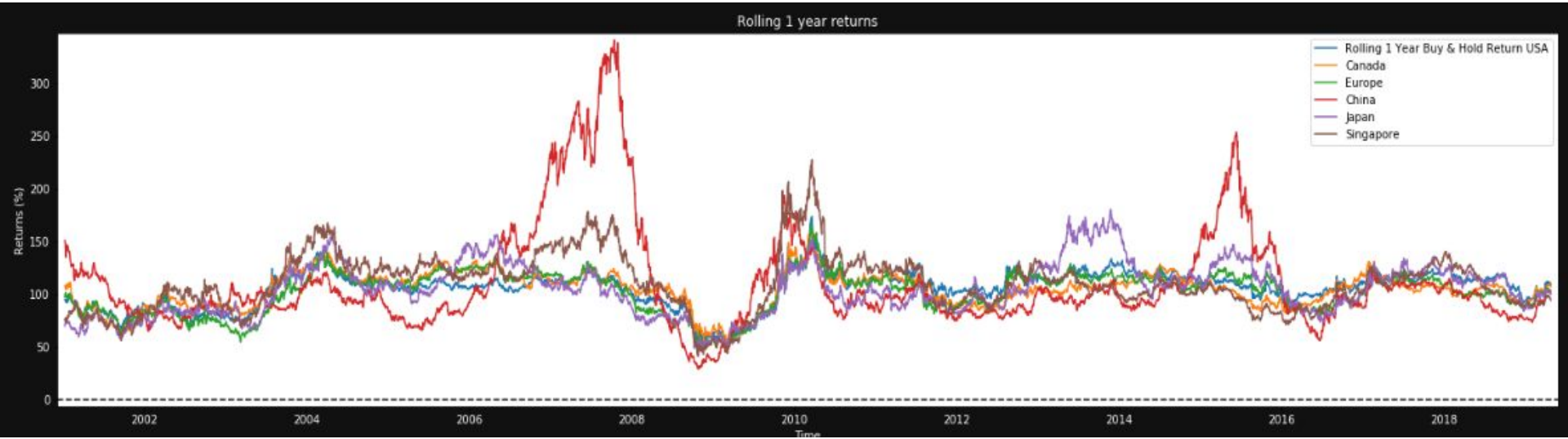
- Adjusted close = closing price after adjustments for all applicable splits and dividend distributions

Note: These returns do not account for exchange rates. This is only to review a general trend of the movement of different countries.



Graph of rolling 1 year returns from 2000 to 2019:

- $\text{Non cumulative returns} = \frac{\text{Adjusted Close}}{\text{Adjusted Close from 1 year ago}} = 1 \text{ year rolling \% return}$



Sharpe ratio for indexes' returns:

- $\text{Sharpe Ratio} = \frac{\text{(average index return) - (countries' risk free rate)}}{\text{indexes' standard deviation}}$

- Risk free rate of a country is the return on treasury bonds ranging from 5 to 30 years.
- Average Risk free rate per country:

```
- USA:       2.76
- Canada:    0.375
- Eurozone:  1.38
- China:     3.754
- Japan:     1.936
- Singapore: 2.41
```

```
SP500 1yr Buy/Hold Sharpe Ratio= -10.08
TSX 1yr Buy/Hold Sharpe Ratio= 4.07
STOXX600 1yr Buy/Hold Sharpe Ratio= -1.9
SSE 1yr Buy/Hold Sharpe Ratio= -5.41
NKE 1yr Buy/Hold Sharpe Ratio= 4.07
TSI 1yr Buy/Hold Sharpe Ratio= 4.07
```

Feature Engineering:

- Features include:
  - % Daily change
  - 3 month moving average
  - 7 month moving average
  - 12 month moving average
  - 90 day moving average as a percentage - this normalizes the data
  - 5 year treasury
  - 10 year treasury
  - 30 year treasury

Create Target Variable:

- Target variable:
  - When Close > yesterdays close.
  - 1: True 0: False

Split train/test sets:

- Shuffle data to ensure randomization
- test size = 30%

Split train/test sets:

- Shuffle data to ensure randomization
- test size = 30%

Preprocessing:

- Use sklearns' StandardScaler():
  - Standardize features by removing the mean and scaling to unit variance.
  - z = (x - u) / s:
    - u = mean of the training samples
    - s = standard deviation of training samples
    - x = independent variables

Note: scale train and test data separately

(Receiver operating characteristic) Roc Curve function + AUC(area under the curve):

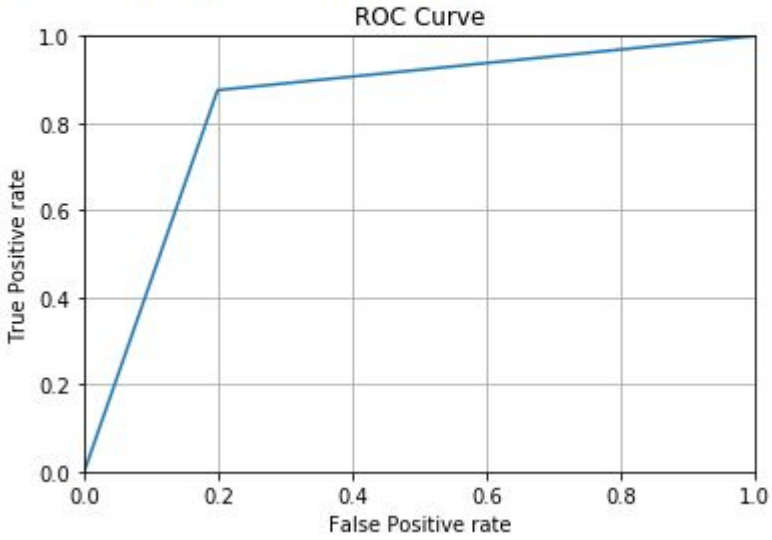
- Features true positive rate on the Y axis, and false positive rate on the X axis.

- Top left corner of plot is 'ideal' - false positive rate of zero & a true positive rate of one.
  - Not very realistic, but larger AUC is usually better.
  - 'Steepness' of ROC is important, ideal to max true positive rate w/ min false positive rate.

Logistic Regression:

- Calculate:
  - Accuracy: $\frac{TP + TN}{TP + TN + FP + FP}$
  - Precision: $\frac{TP}{TP + FP}$
  - Recall: $\frac{TP}{TP + FN}$
  - AUC

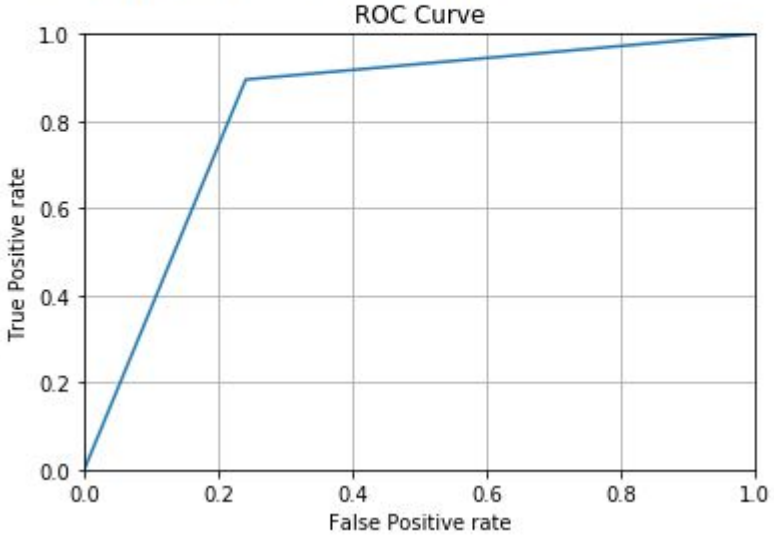
```
LR:
Accuracy:  0.8405797101449275
Precision:  0.8321513002364066
Recall:  0.8756218905472637
AUC of the model is 0.8383711694
```



Support Vector Classifier:

- Calculate:
  - Accuracy: $\frac{TP + TN}{TP + TN + FP + FP}$
  - Precision: $\frac{TP}{TP + FP}$
  - Recall: $\frac{TP}{TP + FN}$
  - AUC

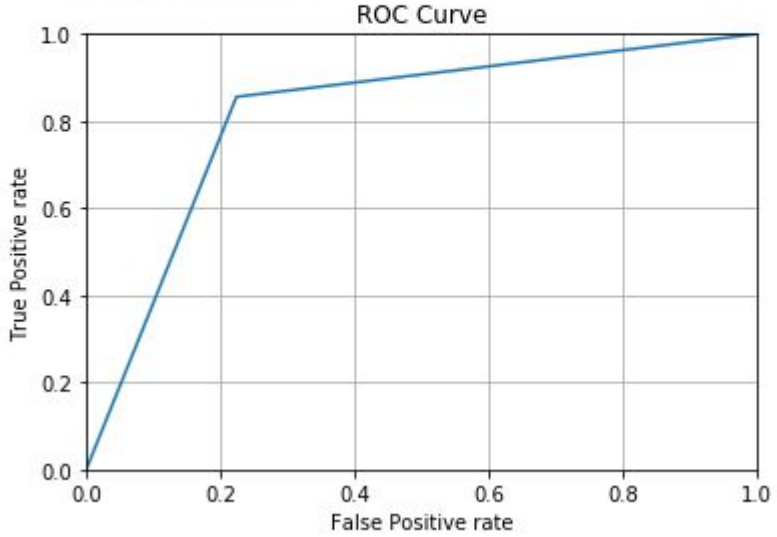
```
SVC:
Accuracy:  0.8313570487483531
Precision:  0.8071748878923767
Recall:  0.8955223880597015
AUC of the model is 0.8273130148
```



Random Forest Classifier:

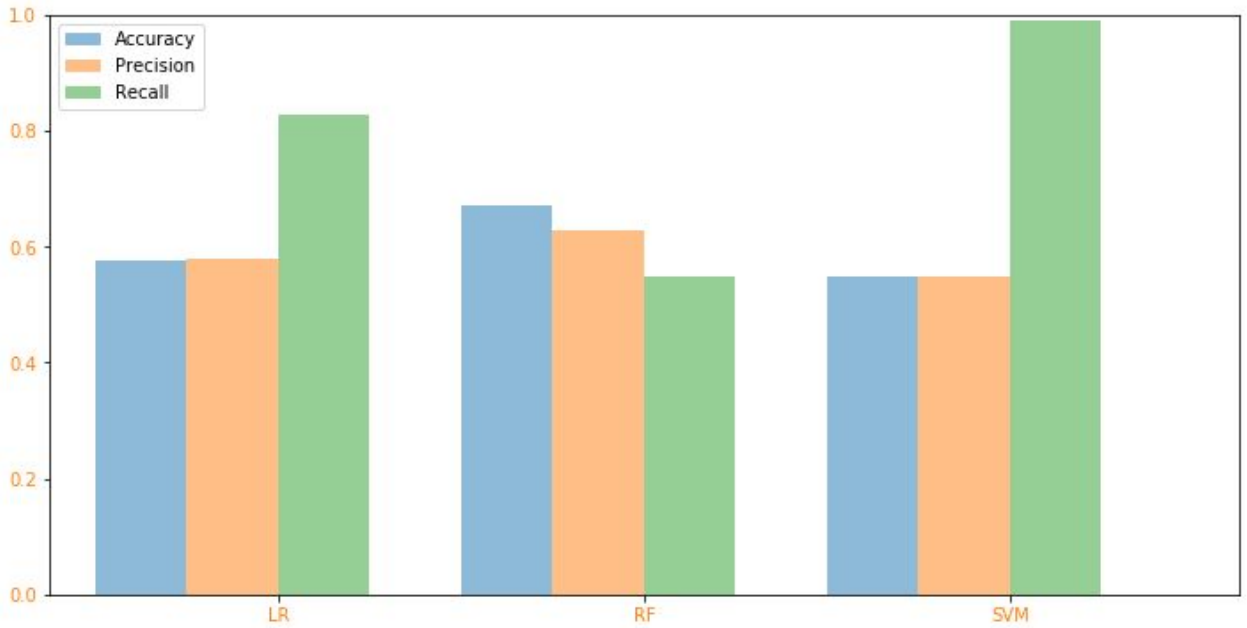
- Calculate:
  - Accuracy: $\frac{TP + TN}{TP + TN + FP + FP}$
  - Precision: $\frac{TP}{TP + FP}$
  - Recall: $\frac{TP}{TP + FN}$
  - AUC

```
rf
Accuracy:  0.8181818181818182
Precision:  0.8113207547169812
Recall:  0.8557213930348259
AUC of the model is 0.8158158786
```



Perform Cross validation on models:

- 5 folds
- Metrics:
  - Accuracy: $\frac{TP + TN}{TP + TN + FP + FP}$
  - Precision: $\frac{TP}{TP + FP}$
  - Recall: $\frac{TP}{TP + FN}$

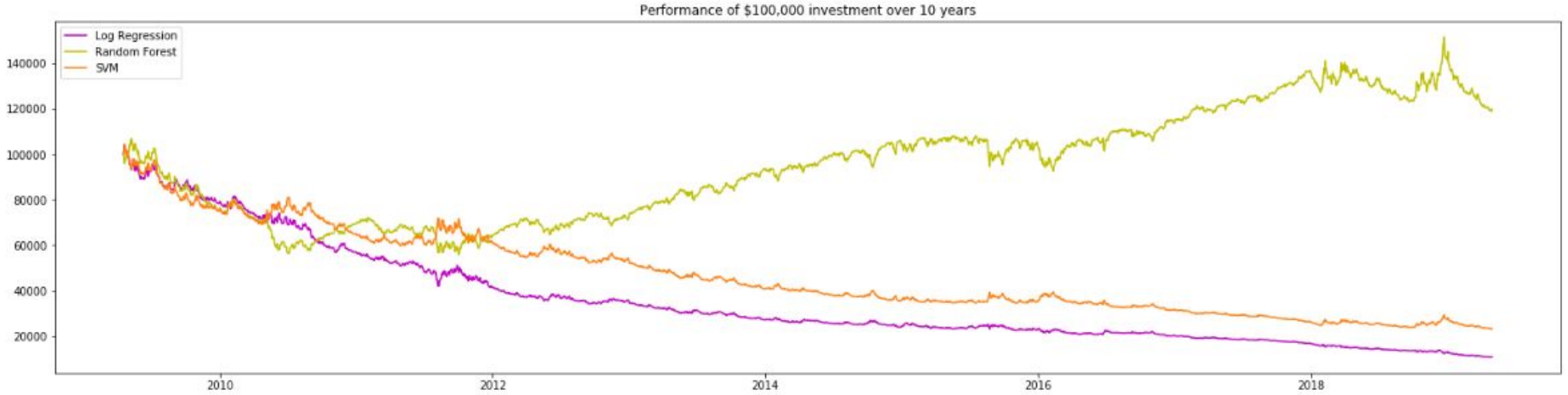


What happens if I place 100,000 USD into the S&P500 of 10 years?

- Will my models be able to predict?
- We've already seen the prediction above, they're not the best. But we can below shows a different visualization.

View Prediction as Graph:

- Compare using:
  - Logistic Regression
  - Random Forest Classifier
  - Support Vector Classifier


Performance of $100,000 investment over 10 years
Log Regression
Random Forest
SVM
140000
120000
100000
80000
60000
40000
20000
2010
2012
2014
2016
2018

Fin