greatlearning

Post Graduate Program in Big Data & Machine Learning

Capstone Project Group- 4

Final Report

Bangalore
2018-2019

Team Members

Sandeep R Diddi
Sanjay Kumar Tiwary
Joyeeta Mallik
Sasirekha Sathasivam

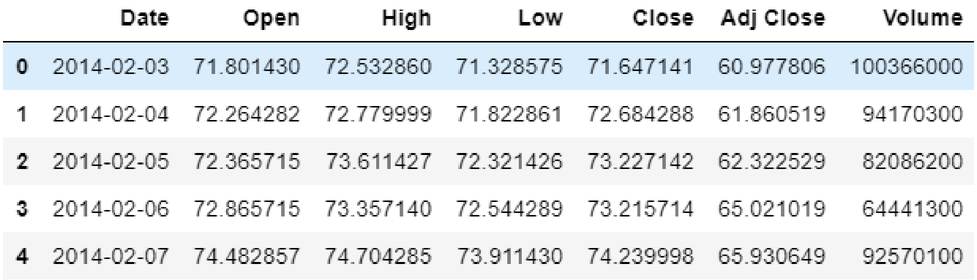
Table of Contents

1. Summary of problem statement, data and findings

Prediction for stock direction prediction is the project executed. For preparing the solution we have selected the google stock data for 5 year historical data with stock open, close, High, Low.

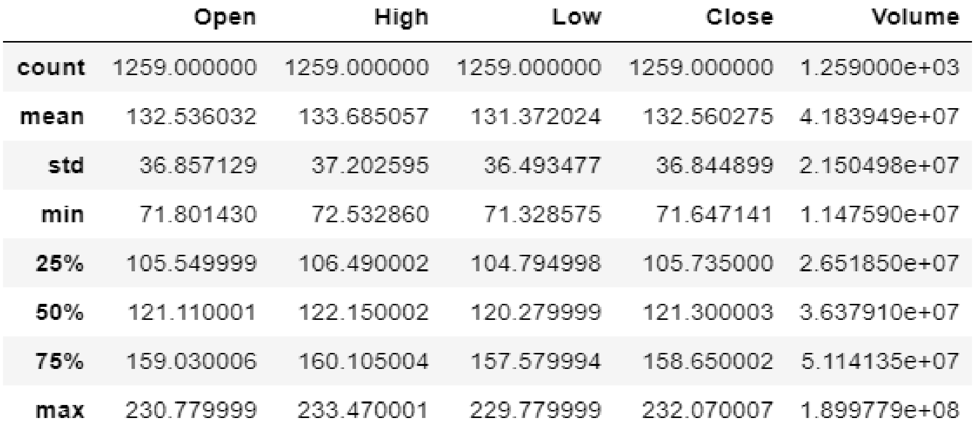
For this we have selected few of alpha signal from word quant company. With the help of various alpha, we prediction the next day price. Alpha 101 was picked as the best calculation to predict the Target of the stock if it tends positive or negative

Data selected for Apple for five years, sample dataset is shown below



	Date	Open	High	Low	Close	Adj Close	Volume
0	2014-02-03	71.801430	72.532860	71.328575	71.647141	60.977806	100366000
1	2014-02-04	72.264282	72.779999	71.822861	72.684288	61.860519	94170300
2	2014-02-05	72.365715	73.611427	72.321426	73.227142	62.322529	82086200
3	2014-02-06	72.865715	73.357140	72.544289	73.215714	65.021019	64441300
4	2014-02-07	74.482857	74.704285	73.911430	74.239998	65.930649	92570100

Table 1: Apple stock price (Raw data)



	Open	High	Low	Close	Volume
count	1259.000000	1259.000000	1259.000000	1259.000000	1.259000e+03
mean	132.536032	133.685057	131.372024	132.560275	4.183949e+07
std	36.857129	37.202595	36.493477	36.844899	2.150498e+07
min	71.801430	72.532860	71.328575	71.647141	1.147590e+07
25%	105.549999	106.490002	104.794998	105.735000	2.651850e+07
50%	121.110001	122.150002	120.279999	121.300003	3.637910e+07
75%	159.030006	160.105004	157.579994	158.650002	5.114135e+07
max	230.779999	233.470001	229.779999	232.070007	1.899779e+08

Table 2: Basic Statistics of stock price

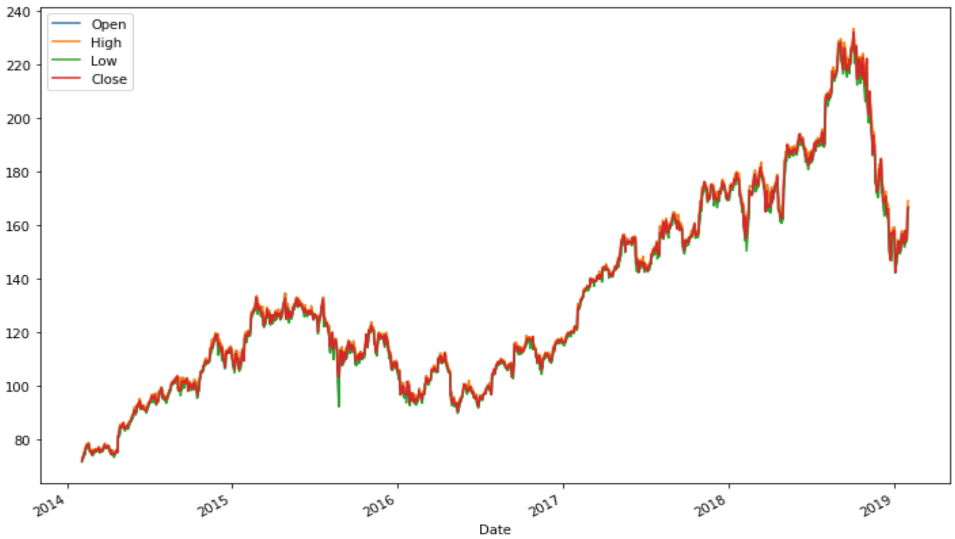


Figure 1: Trend of 5 year stock price

2. Overview of the final process

Calculation of alphas, there are many alpha and we have selected Alpha 101 for our price predictions.

Formula 1:

Alpha#101: ((close - open) / ((high - low) + .001))

Formula 2:

The target label or Y of the machine learning model is created based on this formula.

target = sign(Today close price - Yesterday close price)

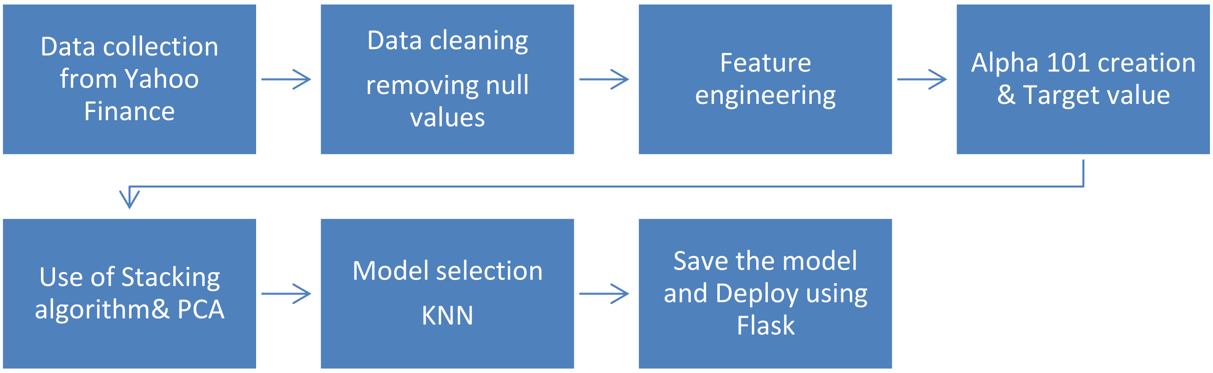
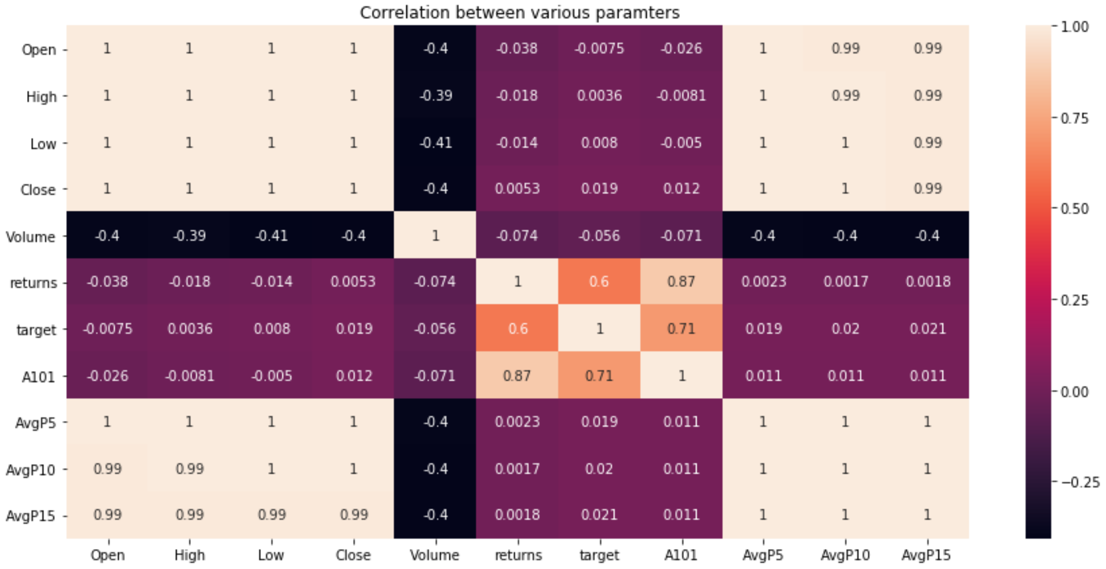

Data collection from Yahoo Finance
Data cleaning removing null values
Feature engineering
Alpha 101 creation & Target value
Use of Stacking algorithm& PCA
Model selection KNN
Save the model and Deploy using Flask

Figure 2:Flow of work

3. Step-by-step walk through of the solution

For this we have pair plot to compare the overall signal correlation amongst various parameters. This pair plot shows that all parameters are highly correlated.

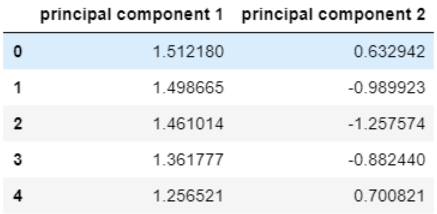


	Open	High	Low	Close	Volume	returns	target	A101	AvgP5	AvgP10	AvgP15
Open	1	1	1	1	-0.4	-0.038	-0.0075	-0.026	1	0.99	0.99
High	1	1	1	1	-0.39	-0.018	0.0036	-0.0081	1	0.99	0.99
Low	1	1	1	1	-0.41	-0.014	0.008	-0.005	1	1	0.99
Close	1	1	1	1	-0.4	0.0053	0.019	0.012	1	1	0.99
Volume	-0.4	-0.39	-0.41	-0.4	1	-0.074	-0.056	-0.071	-0.4	-0.4	-0.4
returns	-0.038	-0.018	-0.014	0.0053	-0.074	1	0.6	0.87	0.0023	0.0017	0.0018
target	-0.0075	0.0036	0.008	0.019	-0.056	0.6	1	0.71	0.019	0.02	0.021
A101	-0.026	-0.0081	-0.005	0.012	-0.071	0.87	0.71	1	0.011	0.011	0.011
AvgP5	1	1	1	1	-0.4	0.0023	0.019	0.011	1	1	1
AvgP10	0.99	0.99	1	1	-0.4	0.0017	0.02	0.011	1	1	1
AvgP15	0.99	0.99	0.99	0.99	-0.4	0.0018	0.021	0.011	1	1	1

Figure 3: Heatmap of stock parameters

Principal component analysis (PCA):

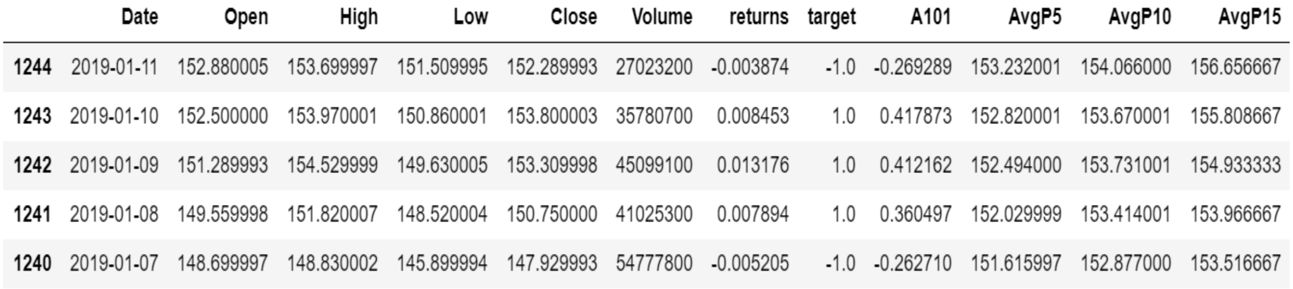
Apply PCA on all the feature for dimension reduction: Linear dimensionality reduction using Singular Value Decomposition of the data to project it to a lower dimensional space.



	principal component 1	principal component 2
0	1.512180	0.632942
1	1.498665	-0.989923
2	1.461014	-1.257574
3	1.361777	-0.882440
4	1.256521	0.700821

Table 3: Principal component

This is final data frame which is going to be pass into model, feature extraction



	Date	Open	High	Low	Close	Volume	returns	target	A101	AvgP5	AvgP10	AvgP15
1244	2019-01-11	152.880005	153.699997	151.509995	152.289993	27023200	-0.003874	-1.0	-0.269289	153.232001	154.066000	156.656667
1243	2019-01-10	152.500000	153.970001	150.860001	153.800003	35780700	0.008453	1.0	0.417873	152.820001	153.670001	155.808667
1242	2019-01-09	151.289993	154.529999	149.630005	153.309998	45099100	0.013176	1.0	0.412162	152.494000	153.731001	154.933333
1241	2019-01-08	149.559998	151.820007	148.520004	150.750000	41025300	0.007894	1.0	0.360497	152.029999	153.414001	153.966667
1240	2019-01-07	148.699997	148.830002	145.899994	147.929993	54777800	-0.005205	-1.0	-0.262710	151.615997	152.877000	153.516667

Table 4: Final Data frame for model development

4. Model evaluation

We have use stacking algorithms to find the best model.

Stacking Classifiers:

Stacking is an ensemble learning technique to combine multiple classification models via a meta-classifier. The individual classification models are trained based on the complete training set; then, the meta-classifier is fitted based on the outputs -- meta-features -- of the individual classification models in the ensemble. The meta-classifier can either be trained on the predicted class labels or probabilities from the ensemble.

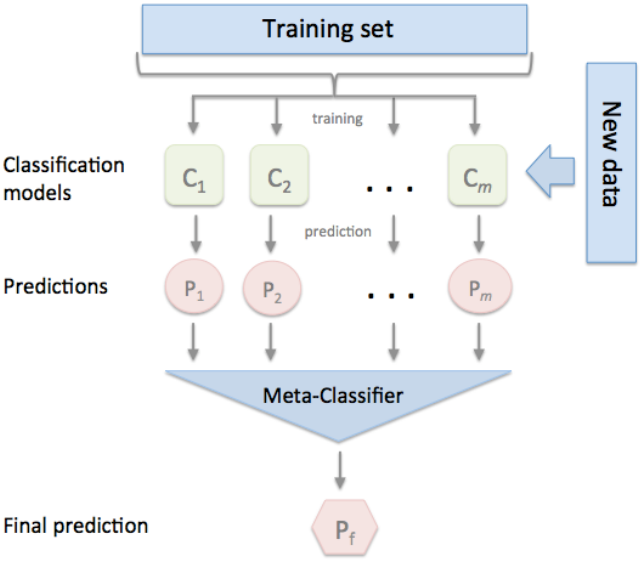


Figure 4: Stacking Classifier

Output is shown below.

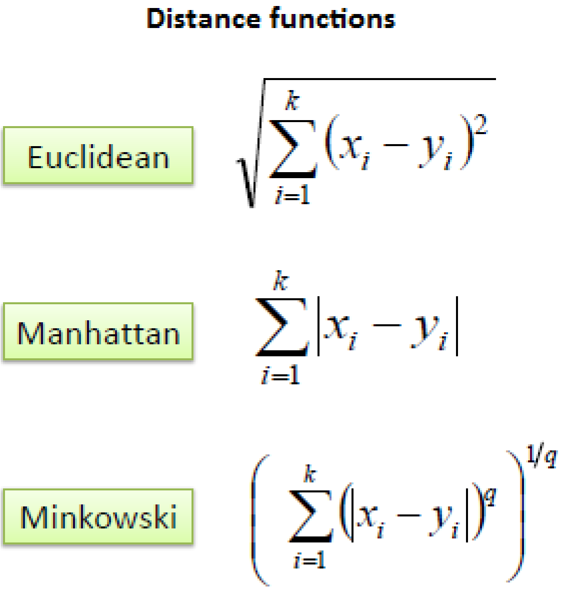
```
3-fold cross validation:

Accuracy: 0.83 (+/- 0.03) [KNN]
Accuracy: 0.81 (+/- 0.04) [Random Forest]
Accuracy: 0.83 (+/- 0.03) [Support vector]
Accuracy: 0.81 (+/- 0.03) [Stacking Classifier]
```

K- Nearest Neighbors:

K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). KNN has been used in statistical estimation and pattern recognition already in the beginning of 1970's as a non-parametric technique.

A case is classified by a majority vote of its neighbors, with the case being assigned to the class most common amongst its K nearest neighbors measured by a distance function. If K = 1, then the case is simply assigned to the class of its nearest neighbor


Distance functions
Euclidean
$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$
Manhattan
$\sum_{i=1}^{k}|x_i - y_i|$
Minkowski
$\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

Equation 1: Distance calculation

In our case KNN & Support vector are giving the best accuracy.

```
K Nearest Neighbors (NN = 20)
Accuracy Score: 79.62466487935657%
Confusion Matrix:
[[139  31]
 [ 45 158]]
Classification Report:
             precision    recall  f1-score   support

       -1.0       0.76      0.82      0.79       170
        1.0       0.84      0.78      0.81       203

avg / total       0.80      0.80      0.80       373
```

5. Comparison to benchmark

How does your final solution compare to the benchmark you laid out at theoutset? Did you improve on the benchmark? Why or why not?

6. Visualization(s)

Plot price return with Alpha 101

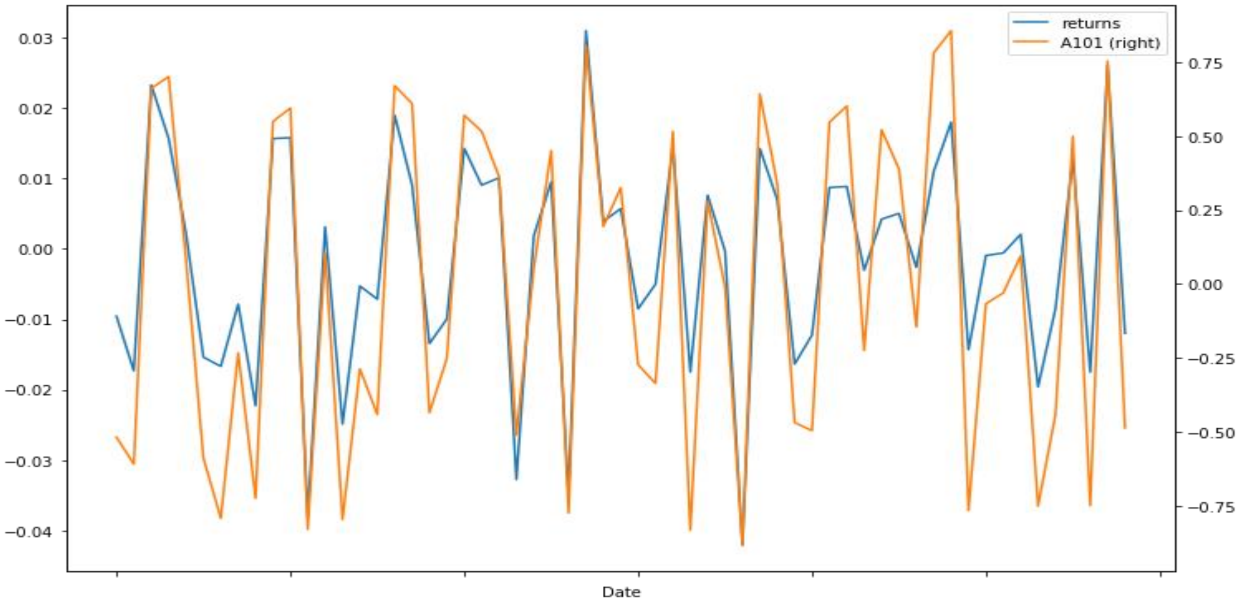


Figure 5: Returns vs Alpha 101

7. Implications

How does your solution affect the problem in the domain or business? What recommendations would you make, and with what level of confidence?

8. Limitations

What are the limitations of your solution? Where does your model fall short in the real world? What can you do to enhance the solution?

9. Closing Reflections