Group Members

- Vaibhav Uppal
- Shivang Ganjoo

Professor Name

- Mr. Venkatesh Gauri Shankar

Subject Name

- Python Programming

19 November 2017

STOCK MARKET PREDICTION

- This project is on stock market prediction of Bharti Airtel for the next day
- We have used two features for our project :-

1, Rival Performance

This feature vector is calculated by taking the mean of three big rivals.

- Tata Communications
- Idea Cellular
- Reliance Communications

2, Self Performance

This feature vector has been calculated by using the k-mean algorithm by forming four clusters of the past 52 days

RESEARCH

K-MEANS ALGORITHM

- This algorithm belongs to unsupervised learning.
- In unsupervised learning the output is not provided in the data set.
- The algorithm itself learns the output by the training examples given.

SUPPORT VECTOR MACHINE

- After our feature matrix is ready we move to the prediction step. Here we use the best algorithm for ‘complex non-linear decision boundary’ as quoted by Andrew NG, the support vector machine.
- SVM is a large margin classifier i.e. it chooses the best decision boundary at an equal distance from the data points of the classes involved.

K-MEAN ALGORITHM

- This algorithm belongs to unsupervised learning.
- In unsupervised learning the output is not provided in the data set.
- The algorithm itself learns the output by the training examples given.

WORK

- The first job we had did is to calculate the rival performance by taking mean of the rival companies and then subtracting the values which came to get the final rival performance values.
- The second job we did was to calculate self performance for that we used K-Means algorithm and formed 4 clusters.Then we sort our company stocks according to the clusters.
- Then we calculated the mean of the values of each cluster seperately and the highest mean was our self performance value.
- Then we used support vector machine to predict the profit or loss of our company in the form of 0(loss) or 1(profit) for the next days

EXPERIMENTAL WORK

CODE

INPUT

```
import pandas as pd;
from sklearn.cluster import KMeans
from sklearn import svm
f = pd.ExcelFile('C:/Users/admin/Downloads/tele_stocks.xlsx')
df = f.parse('Sheet1')

X = pd.DataFrame(df.iloc[:,5:8])
X.drop(['Government Policies'], axis = 1, inplace = True)
s = df.iloc[:,0]
rival = pd.DataFrame(df.iloc[:,1:4])
rival = rival.mean(axis=1)
rival,s = rival.diff(),s.diff()
rival.iloc[0],s.iloc[0] = 0,0
X.iloc[:,0] = rival
X.iloc[:,1] = s
print(X.iloc[:,:2])
#self1 = df.iloc[:,0]



def km():
    f = pd.read_excel('C:/Users/admin/Downloads/tele_stocks.xlsx')
    #print(f.head())
    f1 = f.drop(["Idea Cellular","Tata Communications","Reliance
Communications","Days","Rival","Government Policies","Self Performance"],axis=1)
```

```
    #print(f1.head())
    km = KMeans(n_clusters=4, init='k-means++', n_init=10)
    km.fit(f1)
    KMeans(copy_x=True,init='k-
means++',max_iter=300,n_clusters=4,n_init=10,n_jobs=1,precompute_distances='auto',random_
state=None,tol=0.0001,verbose=0)
    x = km.fit_predict(f1)
    #print(x)
    f["Cluster"]= x
   # print(f.head())
    f1 = f.sort_values(by=['Cluster'])
    #print(f1)
    formean=pd.DataFrame(f1.iloc[:,[0,8]])
    print(formean)
    cluster0=formean[formean.Cluster==0]
    print(cluster0)
    selfmean1=cluster0[['Bharti Airtel']].mean(axis=0)
    print(selfmean1)
    cluster1=formean[formean.Cluster==1]
    print(cluster1)
    selfmean2=cluster1[['Bharti Airtel']].mean(axis=0)
    print(selfmean2)
    cluster2=formean[formean.Cluster==2]
```

```
print(cluster2)

selfmean3=cluster2[['Bharti Airtel']].mean(axis=0)

print(selfmean3)

cluster3=formean[formean.Cluster==3]

print(cluster3)

selfmean4=cluster3[['Bharti Airtel']].mean(axis=0)

print(selfmean4)

s1=float(selfmean1)

s2=float(selfmean2)

s3=float(selfmean3)

s4=float(selfmean4)

if s1>s2 and s1>s3 and s1>s4:

    highest=s1

elif s2>s1 and s2>s3 and s2>s4:

    highest=s2

elif s3>s1 and s3>s2 and s3>s4:

    highest=s3

else:

    highest=s4

print(highest)

x = []

for i in range(1,52):

    x.append(highest)
    X['Self Performance'] = x


    print(X)


k = km()


y = []
for i in range(len(s)):
    if s[i]>=0:
        y.append(1)
    else:
        y.append(0)
model=svm.SVC(kernel='linear',C=1000,gamma=1)
model.fit(X,y)
model.score(X,y)
x = pd.DataFrame([3,2])
x = x.values.reshape(1,-1)
predicted=model.predict(x)
print("predicted =",predicted)
```

OUTPUT

```
      Rival  Self Performance
0  0.000000              0.00
```

1	-2.533333	-31.55
2	-5.316667	3.35
3	5.933333	-19.35
4	0.283333	-4.65
5	5.933333	2.90
6	-3.250000	-4.55
7	-2.066667	-0.80
8	0.900000	8.70
9	-1.550000	3.00
10	-4.133333	-6.25
11	-2.150000	2.95
12	7.566667	1.95
13	-0.683333	2.15
14	-1.150000	5.65
15	8.650000	4.40
16	-3.433333	-3.25
17	4.883333	-1.55
18	0.816667	2.10
19	-1.450000	-1.15
20	0.316667	2.55
21	-6.916667	2.80
22	4.033333	3.90
23	-7.650000	-3.00

24	-3.900000	1.30
25	2.050000	-5.75
26	-2.583333	5.50
27	1.216667	3.55
28	-1.316667	9.35
29	5.700000	6.60
30	-4.366667	5.40
31	-1.016667	1.35
32	-2.066667	-0.80
33	6.683333	6.55
34	-7.716667	-2.10
35	-1.216667	-1.05
36	-8.800000	-10.15
37	6.283333	-2.65
38	3.616667	2.60
39	-0.116667	-5.05
40	-5.316667	-9.90
41	0.400000	1.90
42	-16.566667	6.10
43	2.950000	1.60
44	7.666667	0.35
45	7.366667	1.40
46	8.516667	-0.30

```
47  -4.083333              3.25
48  -0.733333              3.30
49   7.516667             -8.35
50  -0.533333              0.55
    Bharti Airtel  Cluster
50         416.45        0
49         415.90        0
36         421.65        0
37         419.00        0
38         421.60        0
39         416.55        0
42         414.65        0
43         416.25        0
44         416.60        0
45         418.00        0
46         417.70        0
29         422.45        0
28         415.85        0
47         420.95        0
48         424.25        0
11         385.35        1
10         382.40        1
9          388.65        1
```

8	385.65	1
7	376.95	1
6	377.75	1
5	382.30	1
4	379.40	1
3	384.05	1
12	387.30	1
13	389.45	1
41	408.55	2
40	406.65	2
25	397.45	2
26	402.95	2
1	400.05	2
2	403.40	2
14	395.10	2
15	399.50	2
16	396.25	2
17	394.70	2
18	396.80	2
19	395.65	2
20	398.20	2
21	401.00	2
22	404.90	2

```
23      401.90     2
24      403.20     2
27      406.50     2
35      431.80     3
34      432.85     3
33      434.95     3
32      428.40     3
31      429.20     3
30      427.85     3
0      431.60     3
   Bharti Airtel  Cluster
50      416.45     0
49      415.90     0
36      421.65     0
37      419.00     0
38      421.60     0
39      416.55     0
42      414.65     0
43      416.25     0
44      416.60     0
45      418.00     0
46      417.70     0
29      422.45     0
```

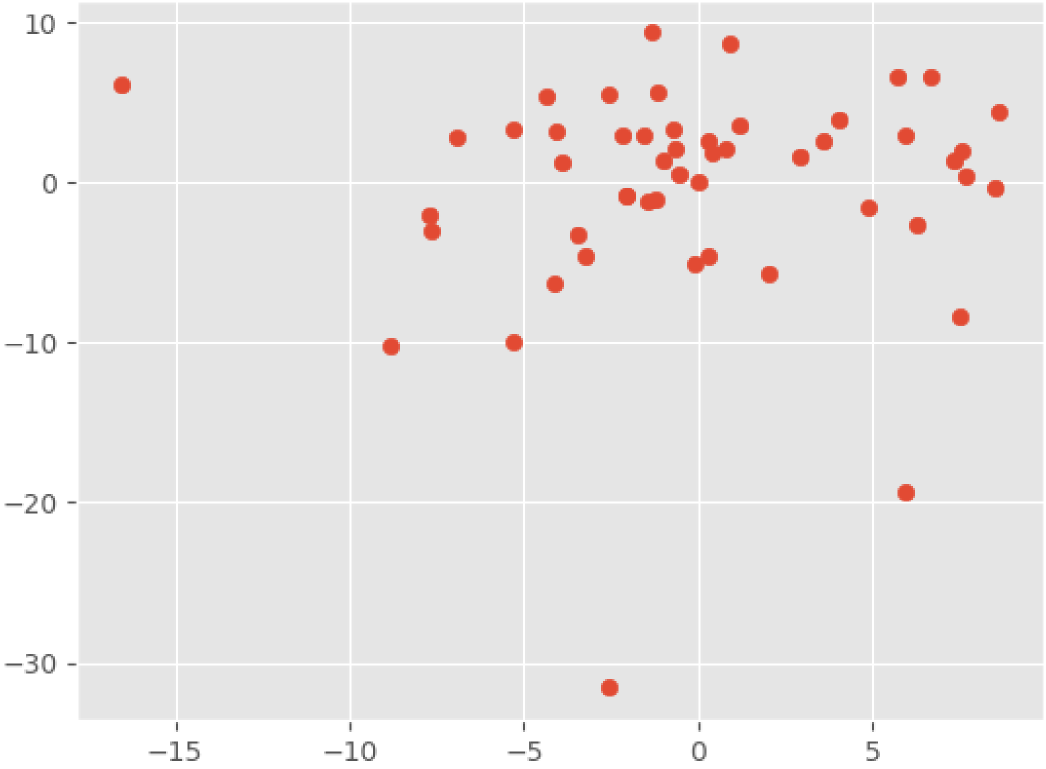
```
28        415.85       0
47        420.95       0
48        424.25       0
Bharti Airtel    418.523333
dtype: float64
    Bharti Airtel  Cluster
11        385.35       1
10        382.40       1
9         388.65       1
8         385.65       1
7         376.95       1
6         377.75       1
5         382.30       1
4         379.40       1
3         384.05       1
12        387.30       1
13        389.45       1
Bharti Airtel    383.568182
dtype: float64
    Bharti Airtel  Cluster
41        408.55       2
40        406.65       2
25        397.45       2
```

```
26         402.95        2
1          400.05        2
2          403.40        2
14         395.10        2
15         399.50        2
16         396.25        2
17         394.70        2
18         396.80        2
19         395.65        2
20         398.20        2
21         401.00        2
22         404.90        2
23         401.90        2
24         403.20        2
27         406.50        2
Bharti Airtel    400.708333
dtype: float64
    Bharti Airtel  Cluster
35         431.80        3
34         432.85        3
33         434.95        3
32         428.40        3
31         429.20        3
```

```
30         427.85        3

0          431.60        3

Bharti Airtel    430.95

dtype: float64

430.94999999999993

        Rival  Self Performance

0    0.000000            430.95

1   -2.533333            430.95

2   -5.316667            430.95

3    5.933333            430.95

4    0.283333            430.95

5    5.933333            430.95

6   -3.250000            430.95

7   -2.066667            430.95

8    0.900000            430.95

9   -1.550000            430.95

10  -4.133333            430.95

11  -2.150000            430.95

12   7.566667            430.95

13  -0.683333            430.95

14  -1.150000            430.95

15   8.650000            430.95

16  -3.433333            430.95
```

17	4.883333	430.95
18	0.816667	430.95
19	-1.450000	430.95
20	0.316667	430.95
21	-6.916667	430.95
22	4.033333	430.95
23	-7.650000	430.95
24	-3.900000	430.95
25	2.050000	430.95
26	-2.583333	430.95
27	1.216667	430.95
28	-1.316667	430.95
29	5.700000	430.95
30	-4.366667	430.95
31	-1.016667	430.95
32	-2.066667	430.95
33	6.683333	430.95
34	-7.716667	430.95
35	-1.216667	430.95
36	-8.800000	430.95
37	6.283333	430.95
38	3.616667	430.95
39	-0.116667	430.95

```
40  -5.316667            430.95

41   0.400000            430.95

42 -16.566667            430.95

43   2.950000            430.95

44   7.666667            430.95

45   7.366667            430.95

46   8.516667            430.95

47  -4.083333            430.95

48  -0.733333            430.95

49   7.516667            430.95

50  -0.533333            430.95

predicted = [1]
```


10
0
−10
−20
−30
−15
−10
−5
0
5

FUTURE SCOPE

- We have not used subjective features like government policies. We will find a way to quantize them and make our predictions more accurate.
- Our dataset is a manually made excel sheet. We will do web scraping and store that data from the web into our dataset.

CONCLUSION

We conclude that our prediction is based on the past data and according to us our project gives accurate result.

REFERENCES

MoneyControl