Stock bot

July 22, 2020

1 Investment/Stock trading bot

The following code is from a Stock Market predictor that I have developed. The Accuracy of this predictor is 81% meaning that it predicts correctly in 81% of the cases if a certain stock's value will increase or decrease.

I hope you don't mind that I keep the variables that I have used as a secret, I have spent many months working on this project and now, I would like to obtain the benefits of this work. Thank you for your understanding :)

Oriol-Boris Monjo Farré

I start by importing the libraries.

[57]:
```
from collections import Counter
import numpy as np
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.utils.validation import column_or_1d
from sklearn.model_selection import cross_val_score
from sklearn.preprocessing import StandardScaler, LabelEncoder
```

Now I import the data that I have already have colected, cleaned and structured. During the process of obtaining the data, I have saved the registers in two different csv. Both documents have the same features in the same order and thus I can concatenate them.

Finally I replace the "Inf" values and drop NA's.

[58]:
```
df1=pd.read_csv("data_training1.csv",index_col=0)
df2=pd.read_csv("data_training2.csv",index_col=0)
df = pd.concat([df1,df2])
df=df.replace([np.inf,-np.inf],np.nan)
df = df.dropna(how='any',axis=0)
```

From all the data that I have I select 1,500,000 registers and I determine which are the most interesting features. I already know the variables that are more important, because prior to this machine learning project, I have done a statistical analysis of the data.

I define the X and Y variables and use the function "StandardScaler" to scale the data. Futhermore, I use the "train_test_split" to mix and separate the data in train/test data and features/target.

Regarding the choosen model, it is the RandomForestClassifier. I have used this model, because, the process that this classifier has to determine if a stock's price will increase, decrease or it will stay more or less the same, is very simillar to the procedures that expert traders use.

[60]:
```
df_train=df.sample(n = 1500000,random_state=5)

variables=[6,9,12,15,17,-5,-4,-3,-2,-1]

X = df_train.iloc[:,variables]
y = df_train["Target"].values
y2 = column_or_1d(y, warn=True)

sc=StandardScaler()
X=sc.fit_transform(X)

X_train, X_test, y_train, y_test =train_test_split(X,y2,test_size=0.25)

clf=RandomForestClassifier(criterion="gini",n_estimators=200)
clf.fit(X_train,y_train)
```

[60]:
```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=200,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

Finally I apply some techniques to see the accuracy of the model and its predictions.

[61]:
```
Accuracy= clf.score(X_test, y_test)
print("Accuracy:", Accuracy)
predictions=clf.predict(X_test)

print(pd.
 ↪crosstab(y_test,predictions,rownames=["Reality"],colnames=["Predictions"]))
```

```
Accuracy: 0.8111893333333333
Predictions     -1       0      1
Reality
-1           44496   30428   4375
 0            5596  227941   2682
 1            6395   21328  31759
```

Understanding the results:

-Accuracy: 0.8111893333333333 : It means that if the bot makes a prediction, it will be correct in more than 81% of the cases.

-Predictions/Reality: This table explains how many registers has been categorized as -1, 0 and 1 by the machine and how many were -1, 0 and 1 according to the real data.

```
-1 means that the stock's price will strongly decrease,
0 means that the price will be more or less the same,
1 means that the price will increase notoriously.
```

[ ]: