Quiz: Decision Trees with Python

Due Feb 17 at 3:30pm **Points** 11 **Questions** 4 **Time Limit** None

Instructions

This exercise is a continuation of the quiz "OLS with Python". Make sure that you complete that quiz correctly first. Then follow the instructions in the document (https://app.box.com/embed_widget/s/vk8159bxwdlflc30sr5kxyumw9uwvogi?view=list&sort=name&direction=ASC&theme=dark) Fintech___Fracassi___Assignment___Decision_Trees_with_Python.pdf (https://app.box.com/embed_widget/s/if89p4wxoa1okap00l9cial8n60wtkcb?view=list&sort=name&direction=ASC&theme=dark) , and finally answer the questions below.

This is an individual exercise, and has no time limit.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	4 minutes	9 out of 11 *

* Some questions not yet graded

Score for this quiz: **9** out of 11 *
Submitted Feb 8 at 11:11am
This attempt took 4 minutes.

Question 1 — 3 / 3 pts

What is the mean squared error of the decision tree model?

- 0.7718440195526606
- 1.0718440195526606
- 0.9718440195526606 — Correct!

0.8718440195526606

Question 2

3 / 3 pts

Take a look at the graph you plotted using the tree.plot_tree function. How many levels are there (including the first main level)?

Correct!

- (selected) 22
- 26
- 20
- 24

Question 3

3 / 3 pts

What is the mean squared error of the random forest algorithm?

- 0.9514334662236628
- 0.9614334662236628
- 0.9714334662236628

Correct!

- (selected) 0.9814334662236628

Question 4

Not yet graded / 2 pts

Post the code below.

Your Answer:

```
# # OLS
#

# In[102]:


import pandas as pd
import numpy as np
from sklearn.datasets import make_classification
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_curve
from sklearn.metrics import roc_auc_score
from sklearn.metrics import mean_squared_error
import math
from matplotlib import pyplot
import os
import statsmodels.api as sm


# In[103]:


df=pd.read_csv('Stock Return Data.csv')
df.columns


# In[104]:


df.head()


# In[105]:


df['Mkt_rf_lead'] = df['Mkt_rf'].shift(-1)
df['Mkt_rf_lead']
```

```
# In[106]:


#df['Mkt_rf_lead'].dropna(how='any', inplace=True)
df.dropna(how='any', inplace=True)
df


# In[107]:


X = np.asarray(df.drop(['Date','Mkt_rf','Mkt_rf_lead'],axis=1))
y = np.asarray(df['Mkt_rf_lead'])


# Create linear regression object
reg = linear_model.LinearRegression()
# Train the model using the training sets
reg.fit(X, y)

# Make predictions using the testing set
pred_x = reg.predict(X)

# The mean absolute error

MSE = mean_squared_error(y, pred_x, squared=True)
RMSE = mean_squared_error(y, pred_x, squared=False)
print(MSE)
print(RMSE)


# In[108]:


dif=y-y_pred
dif_sq=dif*dif
n=len(y)
mse=np.sum(dif_sq)/n
mse2=mean_squared_error(y, y_pred)

rmse=math.sqrt(mse)
print("mse:", mse)
print("mse2:", mse2)
print("rmse:", rmse)
```

```
# In[109]:


df.describe()


# In[110]:


X = sm.add_constant(X)
fit = sm.OLS(y, X).fit()

y_pred=np.asarray(fit.predict())

# Print out the statistics
fit.summary()


# In[111]:


dif=y-y_pred
dif_sq=dif*dif
n=len(y)
mse=np.sum(dif_sq)/n
mse2=mean_squared_error(y, y_pred)

rmse=math.sqrt(mse)
print(mse)
print(mse2)
print(rmse)



# # Decision Trees
#

# In[137]:


from sklearn import tree
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeRegressor
```

```
from matplotlib import pyplot
from sklearn.ensemble import RandomForestRegressor


# In[138]:


Tree = DecisionTreeRegressor(random_state=42,
min_samples_leaf=100)
Tree.fit(X,y)
pred_x = Tree.predict(X)
MSE = mean_squared_error(y, pred_x, squared=True)
RMSE = mean_squared_error(y, pred_x, squared=False)
print(MSE)
print(RMSE)


# In[139]:


tree.plot_tree(Tree)


# In[142]:


from sklearn.ensemble import RandomForestRegressor


# In[145]:


forest = RandomForestRegressor(random_state=42, n_estimators=100,
min_samples_leaf=100, max_features='sqrt')
forest.fit(X,y.ravel())
pred_x = forest.predict(X)
MSE = mean_squared_error(y, pred_x, squared=True)
RMSE = mean_squared_error(y, pred_x, squared=False)
print(MSE)
print(RMSE)
```

Quiz Score: **9** out of 11