Quiz: Shrinkage with Python

Due Feb 15 at 3:30pm	**Points** 7	**Questions** 5	**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
Fintech___Fracassi___Assignment___Shrinkage_with_Python.pdf
(https://app.box.com/embed_widget/s/vk8159bxwdlflc30sr5kxyumw9uwvogi?
view=list&sort=name&direction=ASC&theme=dark) , and finally answer the questions below.

This exercise is individual, and has no time limit.

Attempt History

	Attempt	**Time**	**Score**
LATEST	**Attempt 1**	4 minutes	7 out of 7 *

* Some questions not yet graded

Score for this quiz: **7** out of 7 *
Submitted Feb 6 at 7:39pm
This attempt took 4 minutes.

Question 1   1 / 1 pts

What is the MSE for a lasso model with alpha = 1?

Correct!

- ● 1.0503518694865406

- ○ 1.0403518694865406

- ○ 1.0303518694865406

- ○ 1.0203518694865406

Question 2   2 / 2 pts

What is the MSE for a lasso model with alpha = 0.1?

○ 1.0296251769116054

Correct!

◉ 1.0496251769116054

○ 1.0396251769116054

○ 1.0196251769116054

Question 3   2 / 2 pts

What is the MSE for a lasso model with alpha = 0.01?

○ 1.0112636277086589

○ 1.0212636277086589

Correct!

◉ 1.0312636277086589

○ 1.0412636277086589

Question 4   2 / 2 pts

What is the MSE for a Ridge model with alpha = 1?

○ 1.0479229007247566

Correct!

    ◉ 1.0279229007247566

    ○ 1.0379229007247566

    ○ 1.0579229007247566

Question 5   **Not yet graded / 0 pts**

Copy and paste the code below.

Your Answer:

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[72]:

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

In[73]:

df.head()

```python
# In[74]:


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


# In[75]:


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


# In[76]:


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[77]:


dif=y-y_pred
dif_sq=dif*dif
n=len(y)
```

```python
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[78]:


df.describe()


# In[79]:


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

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

# Print out the statistics
fit.summary()


# In[80]:


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)


# # Shrinkage

# In[97]:
```

```python
from sklearn.linear_model import LinearRegression, Lasso, Ridge
import numpy as np
import matplotlib.pyplot as plt


# In[91]:


lasso = Lasso(alpha=1)
lasso.fit(X, y)
pred_x = lasso.predict(X)
MSE = mean_squared_error(y, pred_x)
RMSE = mean_squared_error(y, pred_x, squared=False)
print(MSE)
print(RMSE)


# In[92]:


lasso = Lasso(alpha=.1)
lasso.fit(X, y)
pred_x = lasso.predict(X)
MSE = mean_squared_error(y, pred_x)
RMSE = mean_squared_error(y, pred_x, squared=False)
print(MSE)
print(RMSE)


# In[93]:


lasso = Lasso(alpha=.01)
lasso.fit(X, y)
pred_x = lasso.predict(X)
MSE = mean_squared_error(y, pred_x)
RMSE = mean_squared_error(y, pred_x, squared=False)
print(MSE)
print(RMSE)


# In[99]:
```

```python
from sklearn.linear_model import Ridge
ridge = Ridge(alpha=1.0)
ridge.fit(X, y)
pred_x = ridge.predict(X)
MSE = mean_squared_error(y, pred_x)
RMSE = mean_squared_error(y, pred_x, squared=False)
print(MSE)
print(RMSE)


# In[100]:


from sklearn.linear_model import Ridge
Ridge = Ridge(alpha=0.1)
Ridge.fit(X, y)
pred_x = Ridge.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[101]:


from sklearn.linear_model import Ridge
Ridge = Ridge(alpha=.01)
Ridge.fit(X, y)
pred_x = Ridge.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: **7** out of 7