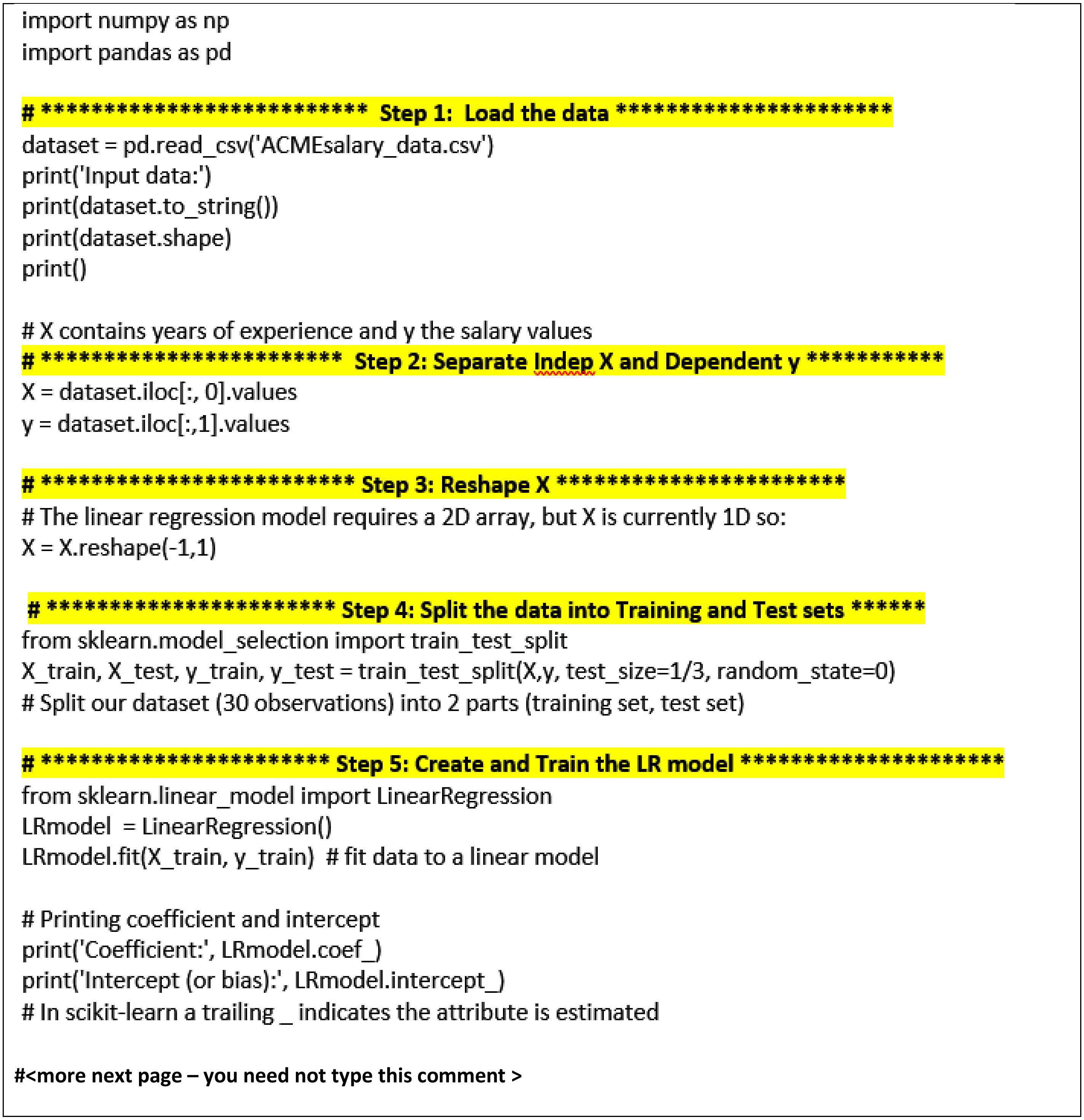
AIN 180

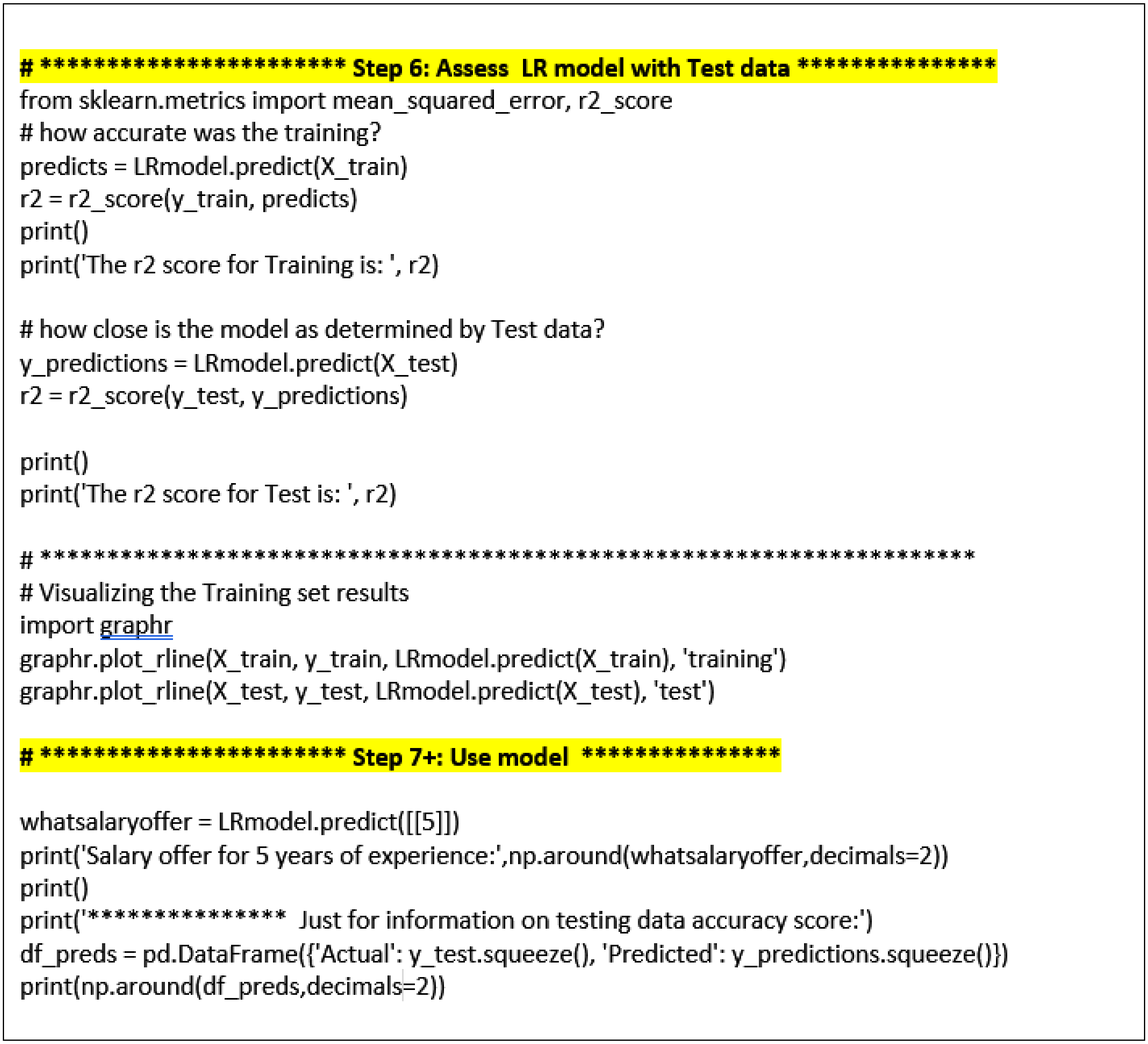
Inclass Assignment
Linear Regression HR Salary estimate

Worth: 30 pts (future code will be shorter! I promise 😊)

The purpose of this assignment is to become familiar with the steps involved in creating and then utilizing a Supervised Machine Learning Linear Regression Model. So, as you are typing consider the comments carefully so that the code is meaningful. Place comments of your own in the code to question anything that is unclear. We will subsequently discuss this same code in class.



```
import numpy as np
import pandas as pd

# ***************************  Step 1:  Load the data ***********************
dataset = pd.read_csv('ACMEsalary_data.csv')
print('Input data:')
print(dataset.to_string())
print(dataset.shape)
print()

# X contains years of experience and y the salary values
# *************************  Step 2: Separate Indep X and Dependent y ***********
X = dataset.iloc[:, 0].values
y = dataset.iloc[:,1].values

# ************************** Step 3: Reshape X ************************
# The linear regression model requires a 2D array, but X is currently 1D so:
X = X.reshape(-1,1)

 # ************************ Step 4: Split the data into Training and Test sets ******
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=1/3, random_state=0)
# Split our dataset (30 observations) into 2 parts (training set, test set)

# ************************ Step 5: Create and Train the LR model **********************
from sklearn.linear_model import LinearRegression
LRmodel  = LinearRegression()
LRmodel.fit(X_train, y_train)  # fit data to a linear model

# Printing coefficient and intercept
print('Coefficient:', LRmodel.coef_)
print('Intercept (or bias):', LRmodel.intercept_)
# In scikit-learn a trailing _ indicates the attribute is estimated

#<more next page – you need not type this comment >
```



```
# *********************** Step 6: Assess  LR model with Test data ***************
from sklearn.metrics import mean_squared_error, r2_score
# how accurate was the training?
predicts = LRmodel.predict(X_train)
r2 = r2_score(y_train, predicts)
print()
print('The r2 score for Training is: ', r2)

# how close is the model as determined by Test data?
y_predictions = LRmodel.predict(X_test)
r2 = r2_score(y_test, y_predictions)

print()
print('The r2 score for Test is: ', r2)

# **************************************************************************
# Visualizing the Training set results
import graphr
graphr.plot_rline(X_train, y_train, LRmodel.predict(X_train), 'training')
graphr.plot_rline(X_test, y_test, LRmodel.predict(X_test), 'test')

# *********************** Step 7+: Use model  ***************

whatsalaryoffer = LRmodel.predict([[5]])
print('Salary offer for 5 years of experience:',np.around(whatsalaryoffer,decimals=2))
print()
print('***************  Just for information on testing data accuracy score:')
df_preds = pd.DataFrame({'Actual': y_test.squeeze(), 'Predicted': y_predictions.squeeze()})
print(np.around(df_preds,decimals=2))
```