*# 1. Import the library***import** numpy **as** np  
**import** pandas **as** pd  
**from** sklearn.metrics **import** root\_mean\_squared\_error  
**from** sklearn.metrics **import** mean\_squared\_error  
**from** sklearn.metrics **import** r2\_score  
**import** matplotlib.pyplot **as** plt  
**from** sklearn.metrics **import** mean\_absolute\_error  
  
  
*# 8. Evaluation model  
# 8.1 true and predict data*predict\_output = pd.read\_csv(**"D:/MY LECTURE 2025/DL - Week 6/W6\_Pred\_ANN.csv"**)  
actual\_output = pd.read\_csv(**"D:/MY LECTURE 2025/DL - Week 6/Actual\_Output.csv"**)  
  
y\_pred = np.array(predict\_output)  
y\_true = np.array(actual\_output)  
  
  
*# axis = 0 (row), axis = 1 (col)*y\_pred\_mean = np.mean(y\_pred, axis=1)  
y\_true\_mean = np.mean(y\_true, axis=1)  
  
*# 8.2 RMSE*RMSE = root\_mean\_squared\_error(y\_pred\_mean,y\_true\_mean)  
print(**"RMSE: "**, **"{:.2f}"**.format(RMSE))  
  
*# 8.3 MSE*MSE = mean\_squared\_error(y\_pred\_mean,y\_true\_mean)  
print(**"MSE: "**, **"{:.2f}"**.format(MSE))  
  
*# 8.4 R*r2 = r2\_score(y\_pred\_mean,y\_true\_mean)  
R = np.sqrt(r2)  
print(**"R: "**, **"{:.2f}"**.format(R))  
  
*# MAE*MAE = mean\_absolute\_error(y\_pred\_mean,y\_true\_mean)  
print(**"MAE: "**, **"{:.2f}"**.format(MAE))  
  
*# 8.5 check pattern*plt.plot(y\_pred\_mean)  
plt.plot(y\_true\_mean)  
plt.title(**'predict vs actual'**)  
plt.ylabel(**'angle (deg)'**)  
plt.xlabel(**'Data Point (%)'**)  
plt.legend([**'predict'**, **'actual'**], loc=**'upper right'**)  
plt.show()