นายฟาริซ หะยีมะสาและ รหัส 61050258

งานวิชา Machine Learning (แทนข้อสอบ Final)

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	DrugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	DrugY
...	...	...	...	...	...	...
195	56	F	LOW	HIGH	11.567	drugC
196	16	M	LOW	HIGH	12.006	drugC
197	52	M	NORMAL	HIGH	9.894	drugX
198	23	M	NORMAL	NORMAL	14.020	drugX
199	40	F	LOW	NORMAL	11.349	drugX

200 rows × 6 columns

อธิบาย dataset

Dataset ที่เลือกใช้เกี่ยวกับการแยกประเภทยาของผู้ป่วยแต่ละคนโดย จะแยกประเภทของยาออกเป็น 5 ประเภท คือ DrugY , drugX , drugA , drugC , drugB และมีคอลัมที่ใช้ประกอบการแยกประเภท 5 คอลัม คือ Age(อายุของผู้ป่วย) , Sex(เพศของผู้ป่วย) , BP(ระดับความดันโลหิต) โดยความดันโลหิตจะแบ่งเป็น ความดันต่ำ , ความดันปกติ และความดันสูง , Cholesterol(ระดับคอเลสเตอรอล) โดยระดับคอเลสเตอรอลแบ่งเป็น คอเลสเตอรอลปกติ และคอเลสเตอรอลสูง , Na_to_K(อัตราส่วนโซเดียมต่อโพแทสเซียมในเลือด)

หมายเหตุ : ตัว dataset ไม่ได้ระบุว่าประเภทของยาแต่ละประเภทหมายถึงยาอะไร

อธิบายการ dummy ข้อมูล จาก String เป็น ตัวเลข

Colume Sex(เพศของผู้ป่วย) แทนค่า M เป็น 0 และ F เป็น 1

Colume BP(ระดับความดันโลหิต) แทนค่า LOW เป็น 0 , NORMAL เป็น 1 และ HIGH เป็น 2

Colume Cholesterol(ระดับคอเลสเตอรอล) แทนค่า NORMAL เป็น 0 และ HIGH เป็น 1

Colume Drug(ประเภทของยา) แทนค่า DrugY เป็น 1 , drugX เป็น 2 , drugA เป็น 3 , drugC เป็น 4 และ drugB เป็น 5

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	1	2	1	25.355	1
1	47	0	0	1	13.093	4
2	47	0	0	1	10.114	4
3	28	1	1	1	7.798	2
4	61	1	0	1	18.043	1
...	...	...	...	...	...	...
195	56	1	0	1	11.567	4
196	16	0	0	1	12.006	4
197	52	0	1	1	9.894	2
198	23	0	1	0	14.020	2
199	40	1	0	0	11.349	2

200 rows × 6 columns

หลังจาก dummy ข้อมูล

อธิบายการทดลอง

Direct Classification .ใช้โมเดล K-nearest-neighbors โดยทดลองทำโดยการหา N ที่เหมาะสมกับ dataset ที่สุด ทดลองใช้ N เป็น 3, 5, 10 และดูผลลัพธ์

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.metrics import accuracy_score

#N=3
classifier_3 = KNeighborsClassifier(n_neighbors=3)
classifier_3.fit(X_train, y_train)
y_pred_3 = classifier_3.predict(X_test)
print("Accuracy N=3 => "+ str(accuracy_score(y_test,y_pred_3)))
#print(confusion_matrix(y_test, y_pred_3))
print(classification_report(y_test, y_pred_3))

#N=5
classifier_5 = KNeighborsClassifier(n_neighbors=5)
classifier_5.fit(X_train, y_train)
y_pred_5 = classifier_5.predict(X_test)
print("Accuracy N=5 => "+ str(accuracy_score(y_test,y_pred_5)))
#print(confusion_matrix(y_test, y_pred_5))
print(classification_report(y_test, y_pred_5))

#N=10
classifier_10 = KNeighborsClassifier(n_neighbors=10)
classifier_10.fit(X_train, y_train)
y_pred_10 = classifier_10.predict(X_test)
print("Accuracy N=10 => "+ str(accuracy_score(y_test,y_pred_10)))
#print(confusion_matrix(y_test, y_pred_10))
print(classification_report(y_test, y_pred_10))
```

Traditional Classification .ใช้โมเดล Decision Tree และ Random Forest ในการทดลอง

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier

#DecisionTree
clf_NDT = DecisionTreeClassifier()
clf_NDT = clf_NDT.fit(X_train,y_train)
y_pred_NDT = clf_NDT.predict(X_test)
print('Accuracy DecisionTree =',accuracy_score(y_test,y_pred_NDT))
print(confusion_matrix(y_pred_NDT, y_test))

#RandomForest
clf_NRF = RandomForestClassifier()
clf_NRF = clf_NRF.fit(X_train,y_train)
y_pred_NRF = clf_NRF.predict(X_test)
print('Accuracy RandomForest =',accuracy_score(y_test,y_pred_NRF))
print(confusion_matrix(y_test, y_pred_NRF))
```

Deep Learning ใช้โมเดล แบบ Multi-layer Perceptron ซึ่งเป็น Model Neural network ประเภทหนึ่ง และได้ทำการทดลองด้วยการปรับค่า hidden layer เท่ากับ 5, 6, 8, 9 ตามลำดับและดูผลลัพธ์

```
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import cross_val_score

#hidden layer = 5
clf_MLP_5 = MLPClassifier(solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(5), random_state=1)
clf_MLP_5.fit(X_train, y_train)
y_pred_MLP_5 = clf_MLP_5.predict(X_test)
print('Accuracy MLPClassifier_5 =',accuracy_score(y_test,y_pred_MLP_5))
print(confusion_matrix(y_test, y_pred_MLP_5))

#hidden layer = 6
clf_MLP_6 = MLPClassifier(solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(6), random_state=1)
clf_MLP_6.fit(X_train, y_train)
y_pred_MLP_6 = clf_MLP_6.predict(X_test)
print('Accuracy MLPClassifier_6 =',accuracy_score(y_test,y_pred_MLP_6))
print(confusion_matrix(y_test, y_pred_MLP_6))

#hidden layer = 8
clf_MLP_8 = MLPClassifier(solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(8), random_state=1)
clf_MLP_8.fit(X_train, y_train)
y_pred_MLP_8 = clf_MLP_8.predict(X_test)
print('Accuracy MLPClassifier_8 =',accuracy_score(y_test,y_pred_MLP_8))
print(confusion_matrix(y_test, y_pred_MLP_8))

#hidden layer = 8
clf_MLP_9 = MLPClassifier(solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(9), random_state=1)
clf_MLP_9.fit(X_train, y_train)
y_pred_MLP_9 = clf_MLP_9.predict(X_test)
print('Accuracy MLPClassifier_9 =',accuracy_score(y_test,y_pred_MLP_9))
print(confusion_matrix(y_test, y_pred_MLP_9))
```

ผลการทดลองของแต่ละโมเดล

Direct Classification ได้ผลการทดลอง N ที่เหมาะสมกับ dataset ที่สุด คือ N=5 ได้ Accuracy เท่ากับ 0.76667

```
Accuracy N=3 => 0.7333333333333333

Accuracy N=5 => 0.7666666666666667

Accuracy N=10 => 0.75
```

Traditional Classification ได้ผลการทดลองของทั้ง Decision Tree และ Random Forest เท่ากัน ได้ค่า Accuracy เท่ากับ 1 มีความแม่นยำในการทำนาย 100%

```
Accuracy DecisionTree = 1.0

Accuracy RandomForest = 1.0
```

Deep Learning ได้ผลการทดลอง ของโมเดลแบบ Multi-layer Perceptron ได้ค่า Accuracy มากที่สุดเท่ากับ 0.9333 ในการใส่พารามิเตอร์ hidden layer เท่ากับ 8

```
Accuracy MLPClassifier_5 = 0.5

Accuracy MLPClassifier_6 = 0.75

Accuracy MLPClassifier_8 = 0.9333333333333333

Accuracy MLPClassifier_9 = 0.6166666666666667
```

สรุปผลและอภิปรายผลการทดลอง

สรุป Direst Classification

หลังจากนำ dataset ไปทำ Direct Classification พบว่ายิ่งใช้จำนวน N เพิ่มมากขึ้นมาโอกาศทำให้การทำนายผิดพลาดได้ เพราะจำนวนข้อมูลที่นำเข้ามาแต่ละ class มีไม่เท่ากันจึงอาจไปเจอ class ที่มีตัวนวนตัวอยู่ใกล้มากกว่าและทำให้การทำนายผิดพลายได้ และการที่จำนวน N น้อยเกินไปก็อาจทำให้ไม่ความแม่นยำลดลงได้เช่นเดียวกัน เพราะตัวที่อยู่ใกล้อาจมีจำนวนข้อมูลของแต่ละ class เท่ากัน จึงต้องเลือกใช้จำนวน N ที่เหมาะสมกับแต่ละ dataset โดยใน Dataset นี้ คือ N=5 จะให้ Accuracy เท่ากับ 0.76666667

สรุป Traditional Classification

ผลการนำ dataset ไปเข้า model Decision Tree และ Random Forest ได้ค่า Accuracy และ confusion matrix เท่ากัน ค่า Accuracy เท่ากับ 1 ซึ่งหมายความว่า model มีการทำนายที่แม่นยำ ไม่มีข้อผิดพลาดเลย ซึ่ง Decision Tree และ Random Forest เป็นการนำ dataset ไปเข้า model เพื่อสร้างต้นไม้ออกมา เพื่อแยกประเภทของแต่ละclass ซึ่งแสดงว่า dataset นี้เหมาะกับการทำ Traditional Classification แบบ Decision Tree และ Random Forest

สรุป Deep Learning

Deep Learning ได้ Accuracy มากที่สุดเท่ากับ 0.9333 ในการใส่พารามิเตอร์ hidden layer เท่ากับ 8 โดยจากการทดลองทำให้เห็นว่า dataset ชุดนี้เหมาะกับการใช้ hidden layer เท่ากับ 8 เพราะเมื่อเพิ่มจำนวน hidden layer ไปเป็น 9 ทำให้ค่า Accuracy ลดลง

สรุปผลการทดลองทั้งหมด

จากการทดลองนำ dataset ไปเข้าโมเดลทั้งแบบ Direst Classification , Traditional Classification และ Deep Learning ได้ข้อสรุปว่า dataset ชุดนี้เหมาะกับการทำ Traditional Classification ที่สุด เพราะได้ค่า Accuracy ดีที่สุด คือ 1 ซึ่งสามารถทำนายข้อมูล test ถูกต้อง 100%