รายงาน

เรื่อง

Assignment 2: สรุปการนำเสนอ, คะแนนการนำเสนอ, และแบบฝึกหัด

จัดทำโดย

นายกฤษณพงษ์ เพ็งบุญ 6330300038

นายจิรเมธ สุทธาวาณิชย์ 6330300119

นายชญานนท์ พูลวาสน์ 6330300151

นายชญานิน ตลับเงิน 6330300160

เสนอ

ผศ.ดร.กุลวดี สมบูรณ์วิวัฒน์

รายงานนี้เป็นส่วนหนึ่งของรายวิชา

03603351 วิทยาศาสตร์ข้อมูลเบื้องต้นหมู่เรียนบรรยาย 800

ภาคต้น ปีการศึกษา 2565

มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตศรีราชา

คำนำ

รายงานเล่มนี้เป็นส่วนหนึ่งของรายวิชา 03603351 วิทยาศาสตร์ข้อมูลเบื้องต้นเพื่อใช้ในการสรุปผลการศึกษา library สำหรับวิทยาศาสตร์ข้อมูล

ผู้จัดทำ

นายกฤษณพงษ์ เพ็งบุญ 6330300038

นายจิรเมธ สุทธาวาณิชย์ 6330300119

นายชญานนท์ พูลวาสน์ 6330300151

นายชญานิน ตลับเงิน 6330300160

สารบัญ

สรุปการนำเสนอ

Numpy

เป็น library สำหรับเก็บข้อมูลตัวเลขในรูปแบบของ array มีความสามารถในการทำงานที่รวดเร็วและประหยัด memory กว่า python list เนื่องจากมีขนาดที่ตายตัวโดยที่ข้อมูลทุกตัวเป็นข้อมูลชนิดดียวกันและเป็น library ที่เขียนขึ้นมาจากภาษา C

Scipy

scipy เป็นlibary ที่สร้างมาให้ปีนส่วนขยายของ numpy โดยมีฟังก์ชันสำหรับการทำงานระดับสูงทางคณิตศาสตร์ วิทยาศาสตร์ วิศวกรรม และการคำนวนทางเทคนิค

Matplotlib

เป็น library พื้นฐานสำหรับการสร้างกราฟเพื่อใช้สำหรับการวิเคราะห์ข้อมูล

Seaborn

เป็น library ที่พัฒนามาจาก matplotlib เพื่อให้การสร้างกราฟมีความสะดวกมากขึ้น เช่น การสร้างกราฟที่มีการทำ linear regression และแสดงผลออกมา

Pandas

เป็น library ที่มีความโดดเด่นในการทำ data analysis, data cleaning เก็บข้อมูลในลักษณะ dictionary + list ช้ากว่า numpy แต่สามารถเก็บข้อมูลที่ไม่ใช่ตัวเลขได้ สามารถใช้งานกับวิทยาศาสตร์ข้อมูลได้ดีกว่า Excel เนื่องจากสามารถจัดการกับข้อมูลจำนวนมากได้

Scikit-supervise

เป็นการใช้ library scikit สำรับสร้างโมเดล ML แบบ supervise ที่มีความง่ายในการใช้งาน และมี dataset ที่เตรียมไว้สำหรับฝึกใช้งาน

supervise learning สามารถแบ่งได้ 2 แบบคือ classification และ regresstion

Algorithm สำหรับ supervise learning -> KNN, decision tree, linear regression, logistic regression, SVM ฯลฯ

Scikit-unsupervise

เป็นการใช้ library scikit สำรับสร้างโมเดล ML แบบ unsupervise ที่มีความง่ายในการใช้งาน และมี dataset ที่เตรียมไว้สำหรับฝึกใช้งาน

unsupervised learning แบ่งได้ 2 แบบคือ Clustering, Dimensionality reduction

Algorithm สำหรับ supervise learning -> KNN, decision tree, linear regression, logistic regression, SVM ฯลฯ

การนำเสนอ

Numpy

9

จุดแข็ง ข้อมูลมีความละเอียดครบถ้วน

จุดอ่อน อ่านสไลด์เป็นส่วนมาก

Scipy

8.5

จุดแข็ง ข้อมูลมีความครบถ้วน

จุดอ่อน ขาดตัวอย่างการใช้งานที่มากพอและมีบางจุดที่อ่านยาก

Matplotlib

9.5

จุดแข็ง ข้อมูลมีความละเอียดครบถ้วน

จุดอ่อน สไลด์มีบางจุดที่อ่านยาก

Seaborn

9

จุดแข็ง ข้อมูลมีความครบถ้วน

จุดอ่อน มีการอ่านสไลด์

Pandas

8.5

จุดแข็ง ข้อมูลค่อนข้างมีความละเอียด

จุดอ่อน ไม่มีการเตรียมความพร้อม

Scikit-supervise

10

จุดแข็ง ข้อมูลมีความละเอียดครบถ้วน

จุดอ่อน -

Scikit-unsupervise

8.5

จุดแข็ง ข้อมูลมีความละเอียดครบถ้วน

จุดอ่อน มีข้อมูลบางส่วนที่มีความผิดพลาด อ่านสไลด์

การบ้าน

Numpy

สร้างอาเรย์ต่อไปนี้

ขนาด 10x10 ทุกช่องมีค่า -3

ขนาด 10x10 แถวคู่เป็น 6ทุกตัว แถวคี่เป็น 9ทุกตัว

ขนาด 10x10 ทุกช่องเป็น 0 ยกเว้นในแนวทแยงมุมจากซ้ายบนลงมาขวาล่างเป็น 1

- one((10,10)) เป็นการสร้าง array 10*10 ที่มีค่าภายในทุกตัว คือ 1 แล้วนำไปคูณกับลบ 3 เพื่อให้ทุกค่ามีค่าเป็น -3

- full((10,10),[6,9]*5) เป็นการสร้าง array 10*10 ที่มีค่าแถวแรก(แถวที่ 0)๖๗เป็น 6 แถวต่อมาเป็น 9 แล้วก็ 6 เป็นจำนวน 5 ชุด

- identity เป็นการสร้างอาเรย์ เมตริกเอกลักษณ์

```
import numpy as np
first = np.ones((10,10))*-3
print(first)

second = np.full((10,10),[6,9]*5)
print(second)

third = np.identity(10)
print(third)
```

Scipy

จงสร้างอาเรย์ 1-40และเก็บไฟล์เป็น .MATLABFrom

scipy.io import savemat

```
import numpy as np
from scipy.io import savemat
a = np.arange(1,41)
savemat('onetoforty.mat',{'a':a})
```

Matplotlib

1)จงสร้างกราฟเส้น(Line Plot)

ylabel ชื่อHomework

xlabel ชื่อDay

Titleชื่อMyHomework

y1 = [5,15,15,20,10]

y2 = [15,20,10,5,15]

y3 = [10,5,15,15,20]

x = [1,2,3,4,5]

กราฟเป็นแบบกริด(grid),สีGreen,alpha=0.1,lw=2,linestyle='--'

1ลักษณะเส้นเป็นเส้นประสลับจุด,markerเป็นดาว,เส้นกราฟเป็นสีMagenta

2.ลักษณะเส้นเป็นเส้นประ,markerเป็นวงกลม,เส้นกราฟเป็นสีGreen

3.ลักษณะเส้นเป็นเส้นจุดประ,markerเป็นสามเหลี่-ยม,เส้นกราฟเป็นสีRed

- เซตกราฟโดย ใช้คำสั่ง plt.grid(True,color='green',alpha=0.1,lw=2,linestyle= '--')

- เซตลักษณะเส้นแรกโดย ใช้คำสั่ง plt.plot(x,y1,linestyle= '-.',marker='*',color='Magenta')

- เซตลักษณะเส้นสองโดย ใช้คำสั่ง plt.plot(x,y2,linestyle= '--',marker='o',color='green')

- เซตลักษณะเส้นสามโดย ใช้คำสั่ง plt.plot(x,y3,linestyle= ':',marker='^',color='red')

- .show() เพื่อแสดงผลลัพธ์

```
import matplotlib.pyplot as plt


y1 = [5,15,15,20,10]
y2 = [15,20,10,5,15]
y3 = [10,5,15,15,20]
x = [1,2,3,4,5]


plt.title('My homework')
plt.xlabel('Day')
plt.ylabel('Homework')
g = plt.grid(True,color='green',alpha=0.1,lw=2,linestyle= '--')
plt.plot(x,y1,linestyle= '-.',marker='*',color='Magenta')
plt.plot(x,y2,linestyle= '--',marker='o',color='green')
plt.plot(x,y3,linestyle= ':',marker='^',color='red')
plt.show()
```

Seaborn

1. กราฟ scatter โดย

1.1) ความหนาของจุดบนกราฟเท่ากับ 100

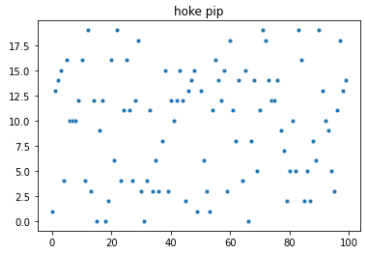
1.2) ตั้ง title ตามชื่อกลุ่มของตัวเอง

1.3) set แกน x และy

- arrange ให้ x = 0-99

- random y 100

- กำหนดชนิดกราฟด้วย .scatterplot() และ set หัวข้อด้วย .set(title = “name”)




```
import seaborn as sb
import matplotlib.pyplot as plt
import numpy as np
x = np.arange(100)
y = np.random.randint(0,20,100)
sb.scatterplot(x=x,y=y,s  = 20).set(title = "hoke pip")
```

2. กราฟ histrogram 2รูป โดย

2.1) รูปที่ 1 ให้พล็อตตามข้อมูลคอลลัมน์ tip เป็นแกนx และข้อมูลของคอลลัมน์smoker เป็นแกน y โดยใชค้า สั่ง facetgrid

2.3) รูปที่ 2 หาเส้นตรงที่เหมาะสมกับข้อมูลโดยใชค้า สั่ง regplot

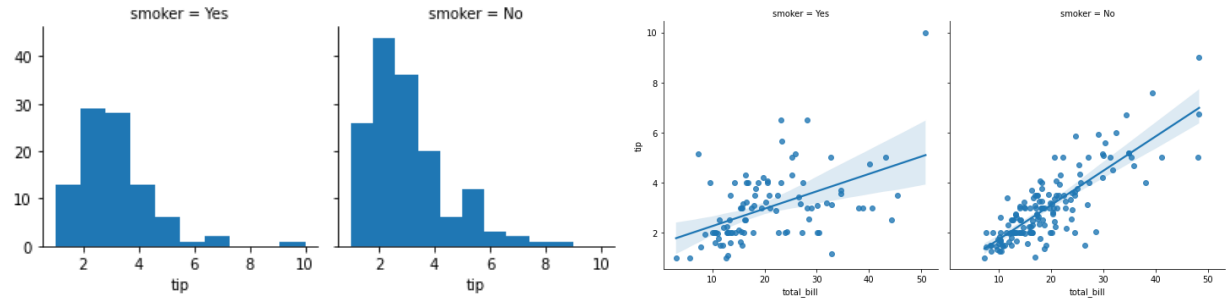
- load data set tips

- สั่งสร้าง facegrid

- สั่ง plot กราฟลงแต่ละช่อง

- regplot + facetgrid = lmplot ใช้คำสั่ง .lmplot และกำหนดค่าตามข้อก่อนหน้า

```
import seaborn as sb
import matplotlib.pyplot as plt
import numpy as np
x = np.arange(100)
y = np.random.randint(0,20,100)
sb.scatterplot(x=x,y=y,s  = 20).set(title = "hoke pip")

t = sb.load_dataset('tips')
g = sb.FacetGrid(t,col = "smoker")
g.map(plt.hist, "tip")


sb.lmplot(x="total_bill", y="tip", col="smoker", data=t)
```



Pandas

นำข้อมูลดังต่อไปนี้และตอบคำถาม.จากข้อมูลข้างต้นให้เขียนคำสั่งของ python และ SQL ให้ได้ผลดังนี้โดยสร้าง data frame จาก โค้ดนี้

```
import pandas as pd

city = {"name": ["Bangkok","Chonburi","Ayuthaya","Samuthprakarn","Lopburi","Korat"],

"population" : [19191919,28282828,37373737,46464646,5555555,12345678], "hospital":[100,20,10,35,69,56]}

 cities = pd.DataFrame(city,columns = ["population","hospital"],index =city["name"])
```

1.แสดงจังหวัดที่มีจำนวนโรงพยาบาลที่มีมากกว่า 50

- Code -> cities[cities["hospital"] > 50]

  Mysql -> select * from cities where hospital > 50;(สมมุติให้ตารางนี้ชื่อ cites)

2.ให้ลบคอลลัมชื่อ ‘population และแถวของ ‘Chonburi’ ออก

- Code -> cities = cities.drop(columns=['population'],index='Chonburi')

  Mysql -> alter table cities drop column population;

  delete from Department where name=’chonburi’;

3.ให้ทำการ insert column population คืนให้อยู่ตำแหน่ง เดิมโดยกำหนดให้

```
 population = [19191919,37373737,46464646,5555555,12345678]
```

- Code -> cities.insert(loc=0, column="population",value= population)

  Mysql -> alter table cities add column population int;

  update cities set population=19191919 where name=’Bangkok’;

  update cities set population=37373737 where name=’Ayuthaya’;

  update cities set population=46464646 where name= ‘Samuthprakarn’;

  update cities set population=5555555 where name= ‘Lopburi’;

  update cities set population=12345678 where name=’Korat’;

4.ให้ทำการอัพเดทข้อมูล

- Code -> cities["hospital"].replace({ 100:123, 10:45}, inplace=True)

  Mysql -> update cities set hospital=123 where hospital=100;

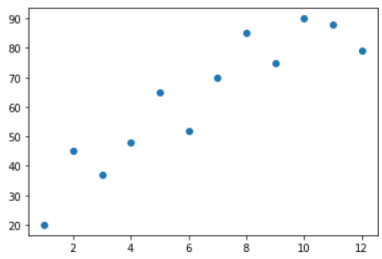
  update cities set hospital=45 where hospital=10;

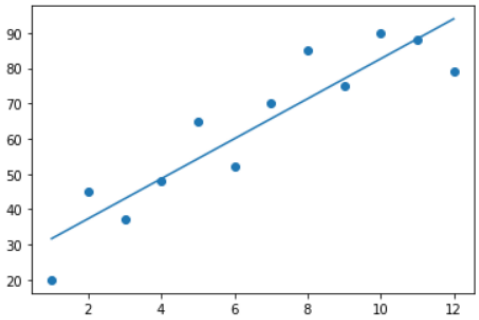
```
import pandas as pd
city = {"name": ["Bangkok","Chonburi","Ayuthaya","Samuthprakarn","Lopburi","Korat"]
        "population" : [19191919,28282828,37373737,46464646,5555555,12345678],
        "hospital":[100,20,10,35,69,56]}
cities = pd.DataFrame(city,columns = ["population","hospital"],index =city["name"])

print(cities[cities["hospital"] > 50])

cities = cities.drop(columns=['population'],index='Chonburi')
print(cities)

population = [19191919,37373737,46464646,5555555,12345678]
cities.insert(loc=0, column="population",value= population)
print(cities)


cities["hospital"].replace({ 100:123, 10:45}, inplace=True)
print(cities)
```

Scikit-supervise

1.จากโค้ดlinear Regressionให(ทดลองเปลี่ยนจำนวนสินค้าที่ขายเป็น 20,45,37,48,65,52,70,85,75,90,88,79แล้วรันโค้ด โดยต้องแสดงผลกราฟให้ดู(codeต้องไม่error)

```
[2]: x=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10,11,12] # เดือน
     y=[20,45,37,48,65,52,70,85,75,90,88,79] # จำนวนสินค้าที่ขายได้ในแต่ละเดือน
     # y = y[::-1]
     plt.scatter(x,y) # Scatter Plot

[2]: <matplotlib.collections.PathCollection at 0x25729677a60>
```





2. จากโค้ดของ Logistic Regresstion ให้ลองเปลี่ยนค่าตรง predict จากที่ predict เป็น สายพันธุ์ versicolor ให้เป็นสายพันธ์ setosa และ virginica แล้วนำรูปภาพมาแสดง

- เปลี่ยนค่า X_new ให้ใกล้เคียงกับสายพันธ์นั้นๆ ดังนี้ Setosa -> [1,2,3,1.5], Virginica -> [1,2,6,2.5]

```
[9]: # Make a prediction
     #X_new = np.array([[sepal length, sepal width ,petal length, petal width]])
     X_new = np.array([[6, 2.5, 4 ,1.2]])
     y_pred = logreg.predict(X_new)
     y_pred_prob = logreg.predict_proba(X_new)
     print("Prediction:", y_pred, "with the probability array:", y_pred_prob)
     print("Predicted target name:", iris["target_names"][y_pred])
     #setosa[1, 2, 3 ,1.5] versicolor[6, 2.5, 4 ,1.2] virginica[1, 2, 6 ,2.5]

Prediction: [1] with the probability array: [[0.0153178  0.95081079 0.03387141]]
Predicted target name: ['versicolor']
```

Scikit-unsupervise

1. ชุดข้อมูลแบบใดเหมาะกับการใช้ วิธีการ unsupervised learning

- ชุดข้อมูลที่เราต้องการเห็นภาพลักษณะรูปแบบการจับกลุ่มของข้อมูลหรือข้อมูลที่มีมิติของข้อมูลมาก

2. จากตัวอย่าง K-means clustering(ในสไลด์หน้า 28 เป็นต้นไป)จงแสดงตัวอย่างข้อมูล 15 รายการแรก

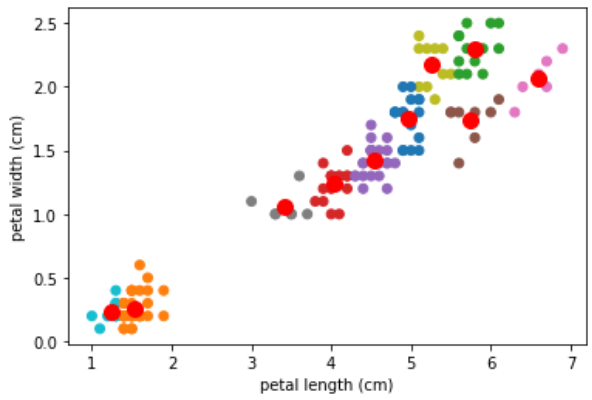
-

```
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.cluster import KMeans
iris = datasets.load_iris()
x = iris.data[:,2:]
y = iris.target
print(x[:15,:])
```

```
[[1.4 0.2]
 [1.4 0.2]
 [1.3 0.2]
 [1.5 0.2]
 [1.4 0.2]
 [1.7 0.4]
 [1.4 0.3]
 [1.5 0.2]
 [1.4 0.2]
 [1.5 0.1]
 [1.5 0.2]
 [1.6 0.2]
 [1.4 0.1]
 [1.1 0.1]
 [1.2 0.2]]
```

3. จากตัวอย่าง (ในสไลด์หน้า 28 เป็นต้นไป)ให้จัดกลุ่มข้อมูลออกเป็น 10 กลุ่ม (โดยใช้วิธี K-meansclustering)

-



```
<None>
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.cluster import KMeans
iris = datasets.load_iris()
x = iris.data[:,2:]
y = iris.target
print(x[:15,:])
plt.scatter(x[:,0],x[:,1],cmap='Paired_r')
plt.xlabel(iris.feature_names[2])
plt.ylabel(iris.feature_names[3])
plt.show()

km = KMeans(n_clusters=10).fit(x)
y_clustered = km.labels_
g=plt.scatter(x[:,0],x[:,1],c=y_clustered,cmap='tab10')
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1], s=100,c='red')
plt.xlabel(iris.feature_names[2])
plt.ylabel(iris.feature_names[3])
plt.show()
```