เรื่อง Iris data visualization and KNN classification

จัดทำโดย

นายพงศธร	พันธ์ศรี	รหัสนักศึกษา	056250204003-8
นางสาวจิรนุช	บูรณ์เจริญ	รหัสนักศึกษา	056250204027-7
นางสาววีระนุช	วุฒิ	รหัสนักศึกษา	056250204030-1
นางสาวกวินภัคร	อินทร์โพธิ์	รหัสนักศึกษา	056250204031-9
นายเกียรติศักดิ์	เสือใหล	รหัสนักศึกษา	056250204035-0
นางสาวธิญาดา	ไวยศร	รหัสนักศึกษา	056250204039-2
นางสาวอมลรดา	มีฉลาด	รหัสนักศึกษา	056250204041-8

วิทยาการข้อมูลและเทคโนโลยีสารสนเทศ ชั้นปีที่ 2 ห้อง 1

เสนอ

อาจารย์ เมธิญานินทร์ คำขาว

รายงานนี้เป็นส่วนหนึ่งของวิชา การทำเหมืองข้อมูล (Data Mining)

คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร

ภาคเรียนที่ 2 ปี การศึกษา 2565

หัวข้อโครงงาน : โปรแกรมการแบ่งกลุ่มไอริสด้วย K-NN

ประเภทของโครงงาน : รายงานทางวิชาการเพื่อการศึกษา

ผู้เสนอโครงงาน	: นายพงศธร	พันธ์ศรี	รหัสนักศึกษา	056250204003-8
	: นางสาวจิรนุช	บูรณ์เจริญ	รหัสนักศึกษา	056250204027-7
	: นางสาววีระนุช	วุฒิ	รหัสนักศึกษา	056250204030-1
	: นางสาวกวินภัคร	อินทร์โพธิ์	รหัสนักศึกษา	056250204030-1
	: นายเกียรติศักดิ์	เสือใหล	รหัสนักศึกษา	056250204035-0
	: นางสาวธิญาดา	ไวยศร	รหัสนักศึกษา	056250204039-2
	: นางสาวอมลรดา	มีฉลาด	รหัสนักศึกษา	056250204041-8

ครูที่ปรึกษาโครงงาน : ผศ.ดร.เมธญาณินฆ์ คำขาว

ปีการศึกษา : 2565

บทคัดย่อ

iris dataset มักจะใช้ในการเริ่มต้นเรียนรู้ กระบวนการสร้าง Machine Learning เพื่อการ Classification โดยในตัวอย่างนี้จะใช้ Support Vector Machine (SVM) โดยเมื่อสร้างและ Train Model เสร็จแล้ว สามารถนำ Model นี้ไปใช้ในการ จำแนก Species ได้ โดยการระบุ ความกว้างและความยาวของกลีบดอกไม้

สารบัญ

บทนำ

Iris เป็นพืชล้มลุกผลิดอกที่ได้รับความนิยมปลูกเป็นไม้ประดับอย่างแพร่หลาย เนื่องจากตัวดอกของไอริสนั้นมีรูปทรงแปลกตาและมีหลายสีเช่นสีม่วง สีน้ำเงินหรือสีขาว คำว่า tectorum ในชื่อภาษาละติน (Iris tectorum) แปลว่าหลังคาบ้าน ซึ่งมาจากในสมัยก่อนได้มีการนำไอริสไปปลูกประดับหลังคาบ้านนั่นเอง อย่างไรก็ตามทุกส่วนของต้นไอริสจะเป็นพิษต่อมนุษย์หากรับประทานเข้าไป

วัตถุประสงค์ของการวิจัย

1. เพื่อสร้างโปรแกรมการแบ่งสายพันธุ์ไอริสด้วยวิธีเหมืองข้อมูล
2. เพื่อวิเคราะห์ความแม่นยำของการแบ่งสายพันธุ์ไอริส
3. เพื่อพัฒนาระบบโปรแกรมการแบ่งสายพันธุ์ไอริสและนำไปพัฒนาระบบแอพพลิเคชันต่อไป

ทฤษฎีที่เกี่ยวข้อง

Iris Species คืออะไร

การจำแนกสายพันธุ์ของพืชตระกูล Iris ออกเป็น 3 กลุ่ม คือ Sentosa, Versicolor, และ Virginica โดยมีข้อมูลอยู่ 4 Feature คือ ความยาวกลีบเลี้ยง (Sepal length), ความกว้างกลีบเลี้ยง (Sepal width), ความยาวกลีบดอก (Petal length), และความกว้างกลีบดอก (Petal width) โดยทั้งหมดมีหน่วยวัดเป็นเซนติเมตร

K Nearest Neighbor

ทฤษฎี KNN

ประเภทของอัลกอริทึม KNN สามารถใช้สำหรับปัญหาการพยากรณ์การถดถอยและการจำแนกประเภท KNN อยู่ในตระกูลอัลกอริทึมการเรียนรู้ภายใต้การดูแล อย่างไม่เป็นทางการ หมายความว่าเราได้รับชุดข้อมูลที่มีป้ายกำกับซึ่งประกอบด้วยการสังเกตการฝึกอบรม (x,y) และต้องการจับความสัมพันธ์ระหว่าง x และ y เป้าหมายของเราคือการเรียนรู้ฟังก์ชัน h:X⟶Y นั่นทำให้ได้ข้อสังเกตที่มองไม่เห็น x , ชั่วโมง(x) สามารถทำนายผล y ที่สอดคล้องกันได้อย่างมั่นใจ

การวัดระยะทาง

ในการตั้งค่าการจำแนกประเภท อัลกอริทึมเพื่อนบ้านที่ใกล้ที่สุดของ K จะลดความสำคัญลงเพื่อสร้างการลงคะแนนเสียงข้างมากระหว่างกรณี K ที่ใกล้เคียงที่สุดกับข้อสังเกตที่ "มองไม่เห็น" ที่กำหนด ความคล้ายคลึงกันถูกกำหนดตามเมตริกระยะทางระหว่างจุดข้อมูลสองจุด ลักษณนาม k-ใกล้ที่สุด-เพื่อนบ้านโดยทั่วไปอิงตามระยะห่างแบบยุคลิดระหว่างตัวอย่างทดสอบกับตัวอย่างการฝึกที่ระบุ ให้ xi เป็นตัวอย่างอินพุตด้วย p คุณสมบัติ (xi1,xi2,...,xip) , น เป็นจำนวนตัวอย่างอินพุตทั้งหมด (i=1,2,...,n). ระยะห่างแบบยุคลิดระหว่างตัวอย่าง xi และ xl ถูกกำหนดเป็น:

$$d(x_i, x_l) = \sqrt{(x_{i1} - x_{l1})^2 + (x_{i2} - x_{l2})^2 + \ldots + (x_{ip} - x_{lp})^2}$$

บางครั้งมาตรการอื่นๆ อาจเหมาะสมกว่าสำหรับสถานที่หนึ่งๆ และรวมถึงระยะทางแมนฮัตตัน เชบีเชฟ และแฮมมิง

การทำเหมืองข้อมูล (Data Mining)

คือเทคนิคที่ใช้คอมพิวเตอร์ช่วยในการวิเคราะห์เพื่อประมวลผลและสำรวจชุดข้อมูลขนาดใหญ่ เมื่อใช้เครื่องมือและวิธีการทำเหมืองข้อมูล องค์กรสามารถค้นพบรูปแบบและความสัมพันธ์ที่ซ่อนอยู่ในข้อมูลของตน การทำเหมืองข้อมูลแปลงข้อมูลดิบเป็นความรู้เชิงปฏิบัติ บริษัทใช้ความรู้นี้ในการแก้ไขปัญหา วิเคราะห์ผลกระทบในอนาคตของการตัดสินใจทางธุรกิจ และเพิ่มขอบเขตกำไรของบริษัท

เทคนิคการทำเหมืองข้อมูลมีอะไรบ้าง

เทคนิคการทำเหมืองข้อมูลอิงจากสาขาวิชาต่างๆ ที่ทับซ้อนกัน รวมถึงการวิเคราะห์ทางสถิติ แมชชีนเลิร์นนิง (ML) และคณิตศาสตร์ เป็นต้น เช่น

- การทำเหมืองตามกฎความเกี่ยวข้อง

การทำเหมืองกฎการเชื่อมโยงเป็นกระบวนการในการค้นหาความสัมพันธ์ระหว่างชุดข้อมูลสองชุดที่ดูเหมือนไม่เกี่ยวข้องกัน คำสั่ง if-then แสดงให้เห็นถึงความน่าจะเป็นของความสัมพันธ์ระหว่างจุดข้อมูลสองจุด นักวิทยาศาสตร์ข้อมูลจะวัดความถูกต้องของผลลัพธ์โดยใช้เกณฑ์การสนับสนุนและความมั่นใจ การสนับสนุนวัดความถี่ที่องค์ประกอบที่เกี่ยวข้องปรากฏในชุดข้อมูล ในขณะที่ความมั่นใจจะแสดงจำนวนครั้งที่คำสั่ง if-then นั้นถูกต้อง

- การจัดหมวดหมู่

การจัดหมวดหมู่เป็นเทคนิคการทำเหมืองข้อมูลที่ซับซ้อนซึ่งฝึกอัลกอริทึม ML เพื่อจัดเรียงข้อมูลเป็นหมวดหมู่ที่แตกต่างกัน ใช้วิธีการทางสถิติ เช่น ผังการตัดสินใจต้นไม้และส่วนที่ใกล้ที่สุดเพื่อระบุหมวดหมู่ สำหรับวิธีการทั้งหมดเหล่านี้ อัลกอริธึมได้รับการตั้งโปรแกรมไว้ล่วงหน้าด้วยการจัดหมวดหมู่ข้อมูลที่รู้จักเพื่อคาดเดาชนิดขององค์ประกอบข้อมูลใหม่

- การทำคลัสเตอร์

การทำคลัสเตอร์คือการจัดกลุ่มจุดข้อมูลหลายจุดเข้าด้วยกันตามความคล้ายคลึง แตกต่างจากการจัดหมวดหมู่เพราะไม่สามารถแยกแยะข้อมูลตามหมวดหมู่เฉพาะ แต่สามารถค้นหารูปแบบในความคล้ายคลึง ผลการทำเหมืองข้อมูลคือชุดของคลัสเตอร์ที่แต่ละคอลเลกชันแตกต่างจากกลุ่มอื่น แต่อ็อบเจกต์ในแต่ละคลัสเตอร์มีความคล้ายคลึงกันในทางใดทางหนึ่ง

- การวิเคราะห์ลำดับและเส้นทาง

ซอฟต์แวร์การทำเหมืองข้อมูลยังสามารถค้นหารูปแบบที่เหตุการณ์หรือค่าชุดหนึ่งนำไปสู่เหตุการณ์ในภายหลัง สามารถรับรู้การเปลี่ยนแปลงบางอย่างในข้อมูลที่เกิดขึ้นในช่วงเวลาปกติหรือในการลดลงและการไหลของจุดข้อมูลในช่วงเวลาหนึ่ง

ผลการวิจัย

นำเข้าและเตรียมข้อมูล

นำเข้าไลบรารี

```
[ ]  import numpy as np
     import pandas as pd
```

โหลดชุดข้อมูล

```
[ ]  # นำเข้าชุดข้อมูล
     dataset = pd.read_csv('Iris.csv')
```

สรุปชุดข้อมูล

```
[ ]  # เราสามารถทราบได้อย่างรวดเร็วว่ามีกี่อินสแตนซ์ (แถว) และจำนวนแอตทริบิวต์ (คอลัมน์) ที่ข้อมูลประกอบด้วยคุณสมบัติรูปร่าง
     dataset.shape
```

```
(150, 6)
```

```
[ ]  dataset.head(5)
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

```
[ ]  #ค่าสถิติเชิงพรรณ
     dataset.describe()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

```
[ ]  # สามารถดูจำนวนอินสแตนซ์ (แถว) ที่เป็นของแต่ละคลาส เราสามารถมองว่านี่เป็นจำนวนที่แน่นอน
     dataset.groupby('Species').size()
```

```
Species
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
dtype: int64
```

แบ่งข้อมูลออกเป็น features และ labels

```
[ ]  feature_columns = ['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm','PetalWidthCm']
     X = dataset[feature_columns].values
     y = dataset['Species'].values

     # วิธีอื่นในการเลือกคุณสมบัติและอาร์เรย์ป้ายกำกับ:
     # X = dataset.iloc[:, 1:5].values
     # y = dataset.iloc[:, 5].values
```

Label encoding

```
[ ]  from sklearn.preprocessing import LabelEncoder
     le = LabelEncoder()
     y = le.fit_transform(y)
```

```
[ ]  print(y)
```

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2]
```

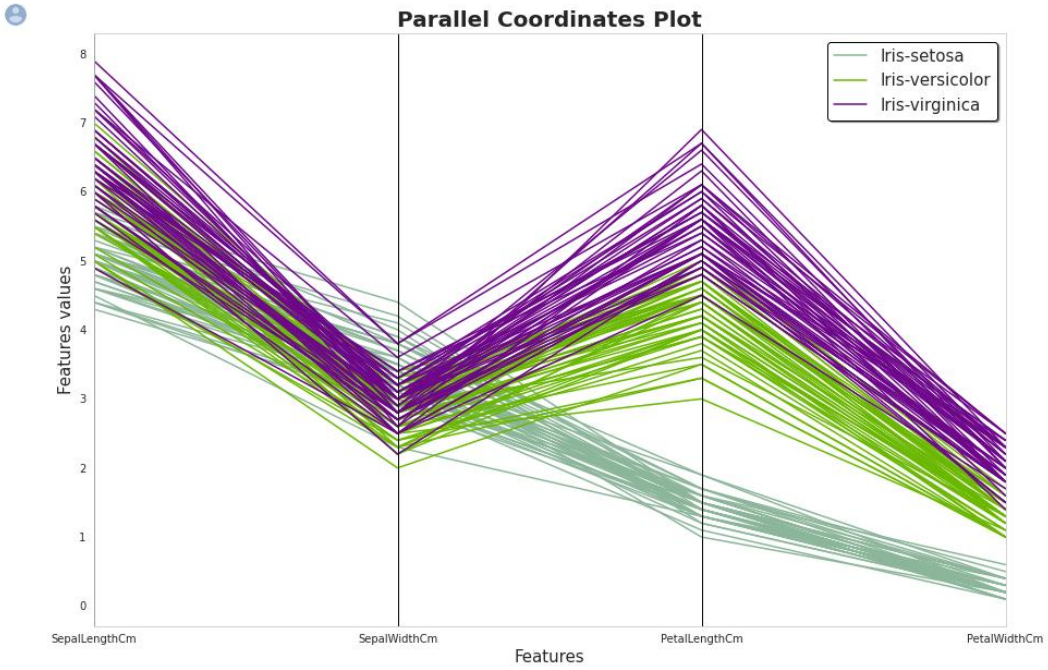
แยกชุดข้อมูลออกเป็นชุดฝึกและชุดทดสอบ

```
[ ]  from sklearn.model_selection import train_test_split
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
```

Data Visualization

Parallel Coordinates (พิกัดคู่ขนาน)

```
from pandas.plotting import parallel_coordinates
plt.figure(figsize=(15,10))
parallel_coordinates(dataset.drop("Id", axis=1), "Species")
plt.title('Parallel Coordinates Plot', fontsize=20, fontweight='bold')
plt.xlabel('Features', fontsize=15)
plt.ylabel('Features values', fontsize=15)
plt.legend(loc=1, prop={'size': 15}, frameon=True,shadow=True, facecolor="white", edgecolor="black")
plt.show()
```



Pairplot

```
plt.figure()
sns.pairplot(dataset.drop("Id", axis=1), hue = "Species", size=3, markers=["o", "s", "D"])
plt.show()
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/axisgrid.py:2076: UserWarning: The `size` parameter has been renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
<Figure size 432x288 with 0 Axes>
```

การใช้ KNN ในการจำแนกประเภท

การทำนาย

```
# Fitting clasifier ให้เข้ากับ Training set
# เรียกใช้ไลบรารี
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix, accuracy_score
from sklearn.model_selection import cross_val_score

# โมเดลการเรียนรู้แบบยกตัวอย่าง (k = 3)
classifier = KNeighborsClassifier(n_neighbors=3)

# Fitting โมเดล
classifier.fit(X_train, y_train)

# การทำนายผลชุดการทดสอบ
y_pred = classifier.predict(X_test)
```

การประเมินการคาดการณ์

```
[ ] cm = confusion_matrix(y_test, y_pred)
    cm
```

```
array([[11,  0,  0],
       [ 0, 12,  1],
       [ 0,  0,  6]])
```

การคำนวณความแม่นยำของโมเดล

```
[ ] accuracy = accuracy_score(y_test, y_pred)*100
    print('Accuracy of our model is equal ' + str(round(accuracy, 2)) + ' %.')
```

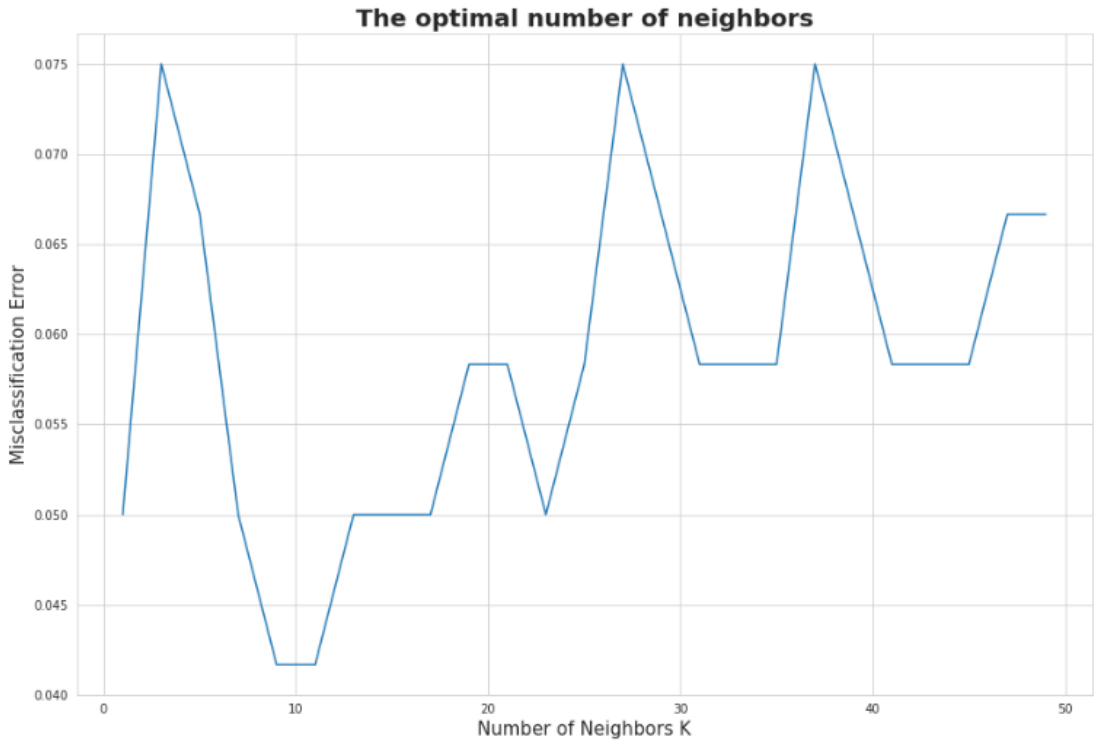
```
Accuracy of our model is equal 96.67 %.
```

การใช้การ cross validation สำหรับการปรับพารามิเตอร์

```
# สร้างรายการ K สำหรับ KNN
k_list = list(range(1,50,2))
# สร้างรายการ cv scores
cv_scores = []

# # ทำการ cross validation 10 เท่า
for k in k_list:
    knn = KNeighborsClassifier(n_neighbors=k)
    scores = cross_val_score(knn, X_train, y_train, cv=10, scoring='accuracy')
    cv_scores.append(scores.mean())
```

```
# เปลี่ยนเป็น misclassification error
import matplotlib.pyplot as plt
import seaborn as sns
MSE = [1 - x for x in cv_scores]

plt.figure()
plt.figure(figsize=(15,10))
plt.title('The optimal number of neighbors', fontsize=20, fontweight='bold')
plt.xlabel('Number of Neighbors K', fontsize=15)
plt.ylabel('Misclassification Error', fontsize=15)
sns.set_style("whitegrid")
plt.plot(k_list, MSE)

plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```




```
[ ]  # หาค่า k ที่ดีที่สุด
     best_k = k_list[MSE.index(min(MSE))]
     print("The optimal number of neighbors is %d." % best_k)
```

```
The optimal number of neighbors is 9.
```

ยอมรับ KNN ทำให้เป็นผล

```python
import numpy as np
import pandas as pd
import scipy as sp

class MyKNeighborsClassifier():
    """
    การใช้อัลกอริทึม KNN ทำให้เป็นผล
    """

    def __init__(self, n_neighbors=5):
        self.n_neighbors=n_neighbors

    def fit(self, X, y):
        """
        ปรับโมเดลให้พอดีโดยใช้ X เป็นอาร์เรย์ของ features และ y เป็นอาร์เรย์ของ labels
        """
        n_samples = X.shape[0]
        #จำนวน neighbors ไม่สามารถมากกว่าจำนวนตัวอย่างได้
        if self.n_neighbors > n_samples:
            raise ValueError("Number of neighbors can't be larger then number of samples in training set.")

        # X และ y ต้องมีจำนวนตัวอย่างเท่ากัน
        if X.shape[0] != y.shape[0]:
            raise ValueError("Number of samples in X and y need to be equal.")

        # ค้นหาและบันทึก labels คลาสที่เป็นไปได้ทั้งหมด
        self.classes_ = np.unique(y)

        self.X = X
        self.y = y

    def predict(self, X_test):

        # จำนวนการคาดการณ์ที่ต้องทำและจำนวนคุณสมบัติภายในตัวอย่างเดียว
        n_predictions, n_features = X_test.shape

        # การจัดสรรพื้นที่สำหรับอาร์เรย์ของการทำนาย
        predictions = np.empty(n_predictions, dtype=int)

        # loop วนรอบการสังเกตทั้งหมด
        for i in range(n_predictions):
            # การคำนวณของการทำนายเดี่ยว
            predictions[i] = single_prediction(self.X, self.y, X_test[i, :], self.n_neighbors)

        return(predictions)
```

```
def single_prediction(X, y, x_train, k):

    # จำนวนตัวอย่างในชุดฝึก
    n_samples = X.shape[0]

    # สร้างอาร์เรย์สำหรับระยะทางและเป้าหมาย
    distances = np.empty(n_samples, dtype=np.float64)

    # การคำนวณระยะทาง
    for i in range(n_samples):
        distances[i] = (x_train - X[i]).dot(x_train - X[i])

    # การรวมอาร์เรย์เป็นคอลัมน์
    distances = sp.c_[distances, y]
    # การเรียงลำดับอาร์เรย์ตามค่าของคอลัมน์แรก
    sorted_distances = distances[distances[:,0].argsort()]
    # celecting labels associeted with k smallest distances
    # การเลือก labels ที่เกี่ยวข้องกับระยะทาง k ที่เล็กที่สุด
    targets = sorted_distances[0:k,1]

    unique, counts = np.unique(targets, return_counts=True)
    return(unique[np.argmax(counts)])
```

```
[ ] # ยกตัวอย่างรูปแบบการเรียนรู้ (k = 3)
    my_classifier = MyKNeighborsClassifier(n_neighbors=3)

    # Fitting โมเดล
    my_classifier.fit(X_train, y_train)

    # การทำนายผลชุดการทดสอบ
    my_y_pred = my_classifier.predict(X_test)
```

```
[ ] accuracy = accuracy_score(y_test, my_y_pred)*100
    print('Accuracy of our model is equal ' + str(round(accuracy, 2)) + ' %.')
```

```
Accuracy of our model is equal 96.67 %.
```

สรุปและการอภิปราย

งานวิจัยนี้เป็นงานวิจัยที่ประยุกต์ใช้อัลกอริทึมการทำเหมืองข้อมูล เพื่อสร้างโปรแกรมการแบ่งสายพันธุ์ไอริสด้วยวิธี K-NN ทดสอบกับชุดข้อมูล จำนวน 150 ชุด โดยในข้อมูลจะประกอบไปด้วย ความกว้างกลีบเลี้ยง,ความยาวกลีบเลี้ยง,ความกว้างกลีบดอก,ความยาวกลีบดอก และ ชื่อสายพันธุ์ของไอริสและคลาสผลลัพธ์ใช้เป็นชื่อสายพันธุ์ของไอริส ได้แก่ Iris-setosa, Iris-verginica, Iris-versicolor การแบ่งชุดข้อมูลเป็นชุดข้อมูลทดสอบและชุดข้อมูลฝึกสอนแบบ 10-Fold Cross Validation วัดประสิทธิภาพด้วยตัวชี้วัด คือ ค่าความถูกต้องผลการทดสอบพบว่าโปรแกรมการแบ่งสายพันธุ์ไอริสด้วยวิธี K-NN มีค่าความแม่นยำสูงสุดเท่ากับ 96.67%

เอกสารอ้างอิง

Iris Species คืออะไร https://guopai.github.io/ml-blog04.html

https://sysadmin.psu.ac.th/2017/09/26/machine-learning-01-python-with-iris-dataset/

การทำเหมืองข้อมูลคืออะไร shorturl.at/gjA26

Iris data visualization and KNN classification https://www.kaggle.com/code/skalskip/iris-data-visualization-and-knn-classification

MIT Lecture: https://www.youtube.com/watch?v=09mb78oiPkA

Iris dataset: https://www.kaggle.com/uciml/iris

Theory: http://www.scholarpedia.org/article/K-nearest_neighbor

https://machinelearningmastery.com/tutorial-to-implement-k-nearest-neighbors-in-python-from-scratch/

https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/

https://www.analyticsvidhya.com/blog/2014/10/introduction-k-neighbours-algorithm-clustering/

ข้อมูลผู้จัดทำ

นายพงศธร พันธ์ศรี รหัสนักศึกษา 056250204003-8 ปวข.64/1

หน้าที่ คนสร้างโปรแกรมและนำเสนอ

นางสาวจิรนุช บูรณ์เจริญ รหัสนักศึกษา 056250204027-7 ปวข.64/1

หน้าที่ นำเสนอ

นางสาววีระนุช วุฒิ รหัสนักศึกษา 056250204030-1 ปวข.64/1

หน้าที่ คนสร้างโปรแกรมและนำเสนอ

นางสาวกวินภัคร อินทร์โพธิ์ รหัสนักศึกษา 056250204031-9 ปวข.64/1

หน้าที่ คนช่วยคิดโปรแกรมและนำเสนอ

นายเกียรติศักดิ์ เสือใหล รหัสนักศึกษา 056250204035-0 ปวข.64/1

หน้าที่ คนช่วยคิดโปรแกรมและนำเสนอ

นางสาวธิญาดา ไวยศร รหัสนักศึกษา 056250204039-2 ปวข.64/1

หน้าที่ จัดทำรูปเล่ม ค้นหาข้อมูลการวิจัยและทำสไลค์

นางสาวอมลรดา มีฉลาด รหัสนักศึกษา 056250204041-8 ปวข.64/1

หน้าที่ ค้นหาข้อมูลการวิจัยและนำเสนอ