



Star dataset

group 4

by Three “P” group



About data

Absolute Temperature - อุณหภูมิสัมบูรณ์ (เป็น K)

Relative Luminosity - ความส่องสว่างสัมพัทธ์ (L/L_0)

Relative Radius - รัศมีสัมพัทธ์ (R/R_0)

Absolute Magnitude - ขนาดสัมบูรณ์ (M_V)

Star Color - สีของดาว (ขาว, แดง, น้ำเงิน, เหลือง, เหลืองส้ม เป็นต้น)

Spectral Class - ระดับสเปกตรัม (O,B,A,F,G,K,,M)

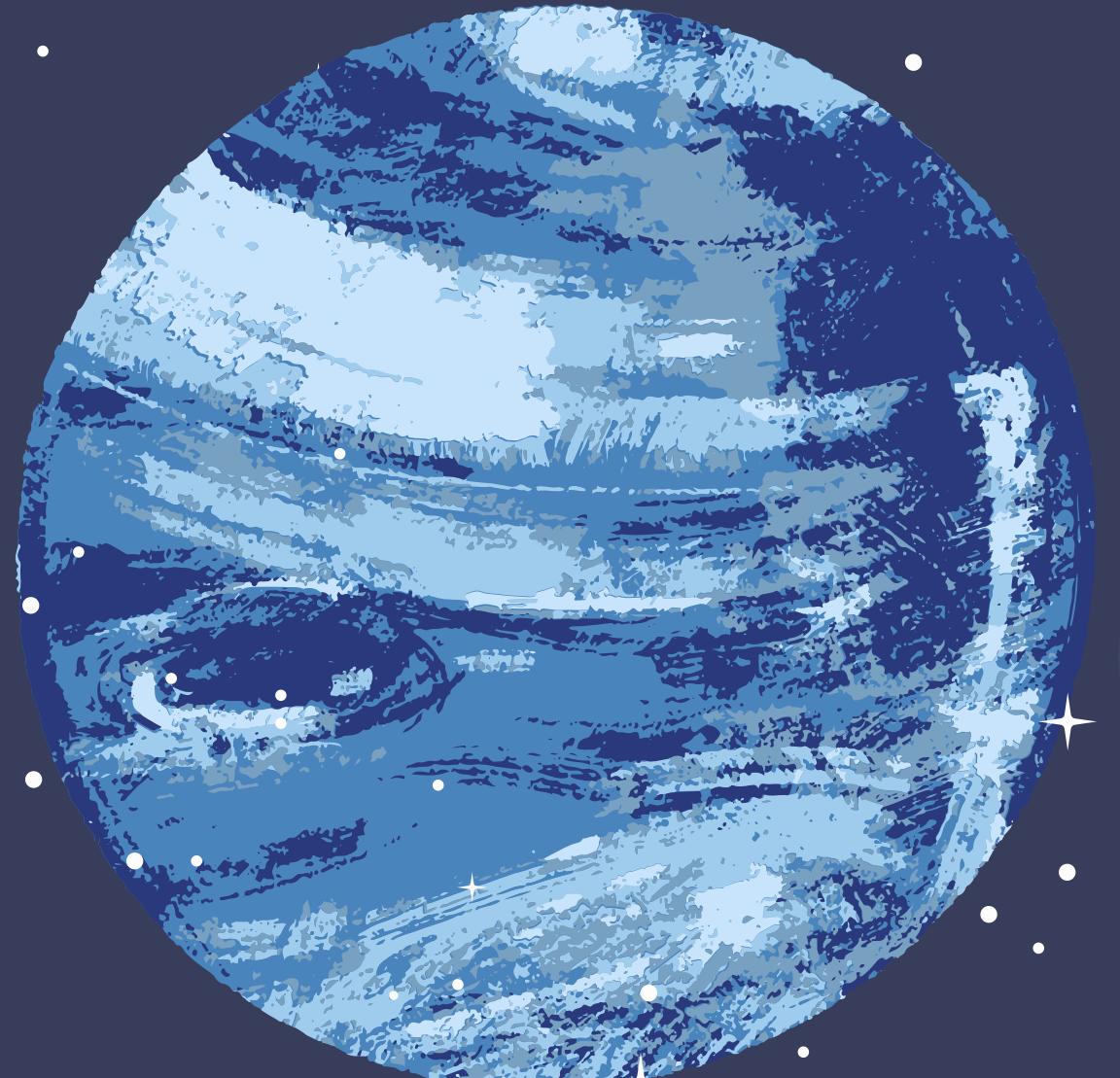
Star Type - ประเภทดาว (ดาว凸率等, ดาว凸率等, ดาว凸率等, ดาว凸率等, แบบจำดับหลัก,
ดาวฤกษ์ยักษ์ใหญ่, ดาวฤกษ์ยักษ์ใหญ่)

Import Data

	Temperature (K)	Luminosity(L/Lo)	Radius(R/Ro)	Absolute magnitude(Mv)	Star type	Star color	Spectral Class
0	3068	0.002400	0.1700	16.12	0	Red	M
1	3042	0.000500	0.1542	16.60	0	Red	M
2	2600	0.000300	0.1020	18.70	0	Red	M
3	2800	0.000200	0.1600	16.65	0	Red	M
4	1939	0.000138	0.1030	20.06	0	Red	M
...
235	38940	374830.000000	1356.0000	-9.93	5	Blue	O
236	30839	834042.000000	1194.0000	-10.63	5	Blue	O
237	8829	537493.000000	1423.0000	-10.73	5	White	A
238	9235	404940.000000	1112.0000	-11.23	5	White	A
239	37882	294903.000000	1783.0000	-7.80	5	Blue	O

240 rows × 7 columns

Question ?



Association Rule

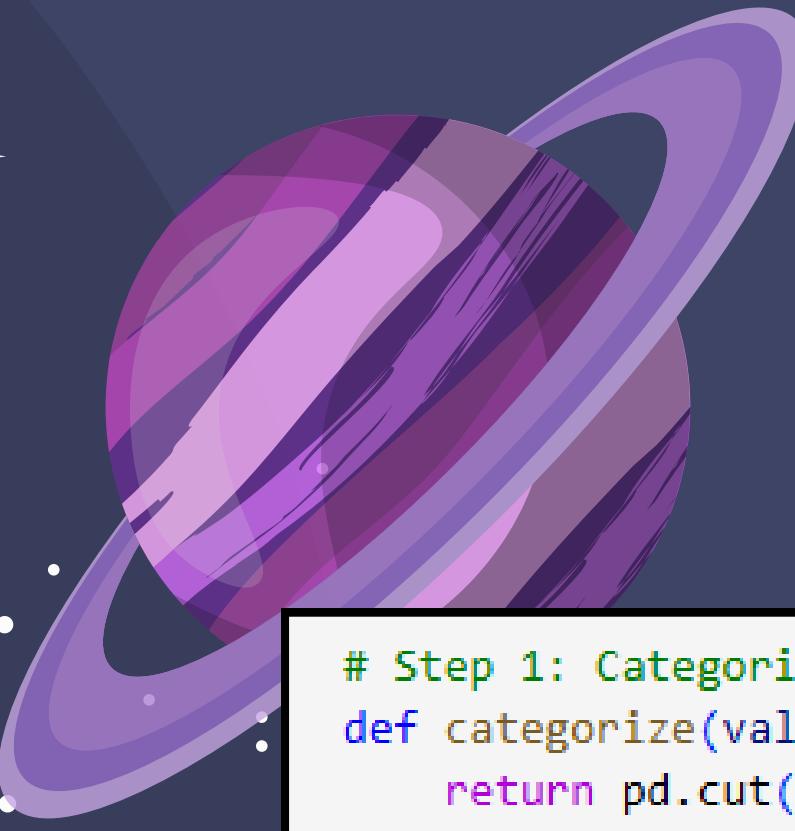
วิเคราะห์ความสัมพันธ์ระหว่าง categories ต่าง ๆ ที่เกิดขึ้นร่วมกันในข้อมูลดาวฤกษ์ซึ่งบอกให้เห็นถึงความสัมพันธ์ของ Luminosity Category, Magnitude Category, และ Radius Category ในการจำแนกดาว

Model

กำหนด Spectral Class ของดาวฤกษ์



Association Rule



Association Rule

```
# Step 1: Categorize continuous data (Temperature, Luminosity, Radius)
def categorize(value, bins, labels):
    return pd.cut([value], bins=bins, labels=labels)[0]

# Binning temperature and luminosity into categories
temp_bins = [0, 4000, 7000, 10000, 20000, 30000, 40000]
temp_labels = ['Very Cold', 'Cold', 'Medium', 'Hot', 'Very Hot', 'Extremely Hot']

luminosity_bins = [0, 0.1, 1, 10, 100, 1000, 10000]
luminosity_labels = ['Very Low', 'Low', 'Medium', 'High', 'Very High', 'Extremely High']

radius_bins = [0, 1, 2, 3, 4, 5]
radius_labels = ['Very Small', 'Small', 'Medium', 'Large', 'Very Large']

star['Temperature Category'] = star['Temperature (K)'].apply(lambda x: categorize(x, temp_bins, temp_labels))
star['Luminosity Category'] = star['Luminosity(L/Lo)'].apply(lambda x: categorize(x, luminosity_bins, luminosity_labels))
star['Radius Category'] = star['Radius(R/Ro)'].apply(lambda x: categorize(x, radius_bins, radius_labels))

# Categorizing Absolute Magnitude based on its value
star['Magnitude Category'] = star['Absolute magnitude(Mv)'].apply(lambda x: 'Bright' if x < 0 else 'Dim')
```



Association Rule

ลักษณะของดาว

Luminosity Category = Very Low (ความสว่างต่ำมาก)

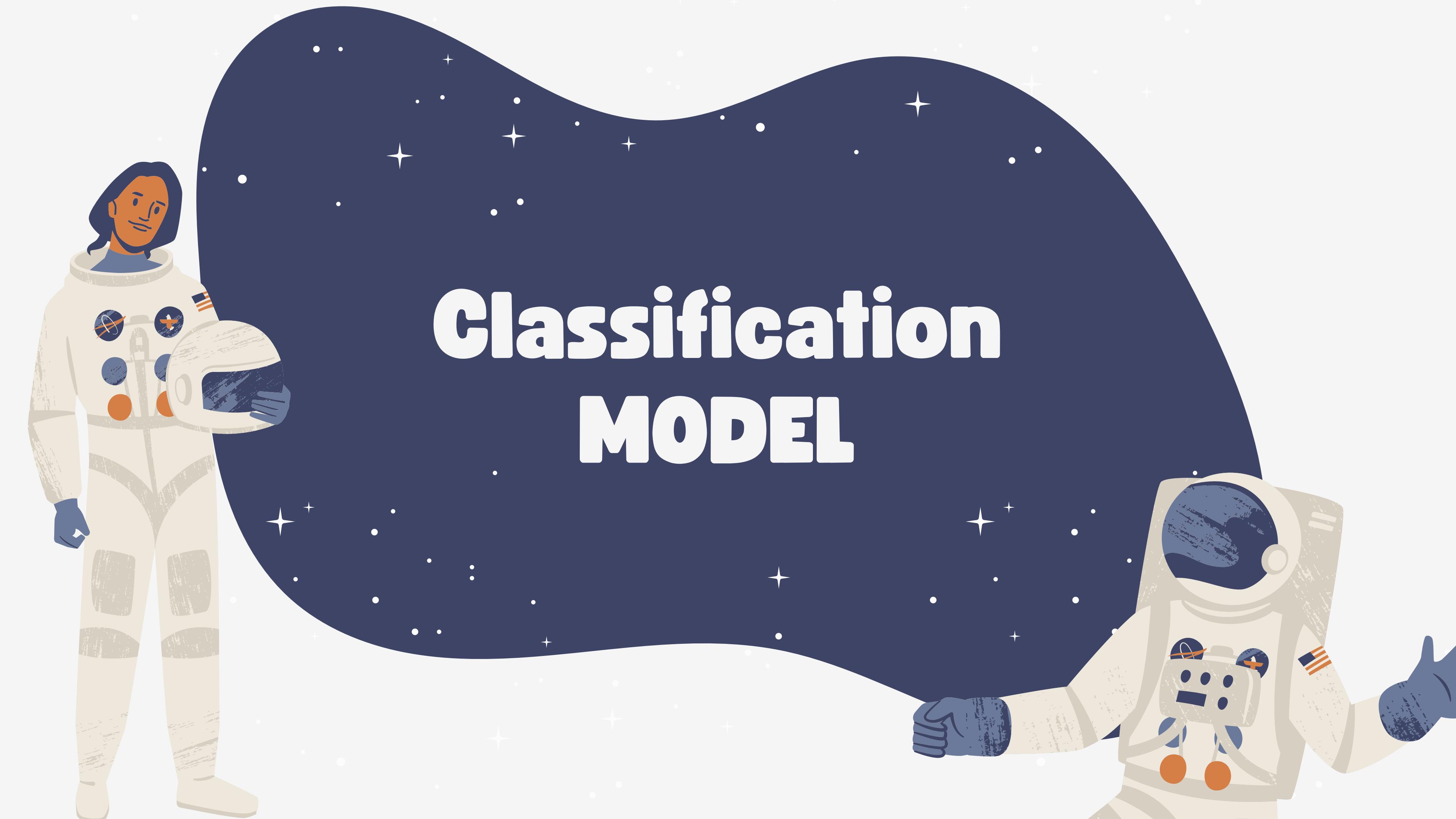
Magnitude Category = Dim (มีความมืดมาก)

Radius Category = Very Small (มีรัศมีเล็กมาก)

Association Rule

Item1	Item2	Item3	Support	Lift
Luminosity Category = Very Low	Magnitude Category = Dim	Radius Category = Very Small	50.42%	1.89 เท่า

ผลการวิเคราะห์พบว่า Luminosity Category = Very Low, Magnitude Category = Dim และ Radius Category = Very Small เกิดร่วมกันบ่อยครั้ง โดยมี Support = 50.42% และ Confidence = 100% หมายความว่าเมื่อพบ Luminosity ต่ำมาก จะพบ Magnitude ต่ำและ Radius เล็กเสมอ ค่าของ Lift = 1.8898 แสดงถึงความสัมพันธ์ที่แข็งแกร่งกว่าการเกิดขึ้นแบบสุ่มถึง 1.89 เท่า ซึ่งบ่งชี้ว่าดาวในกลุ่มนี้มักเป็น ดาว凸 ดาวแคระแดง หรือดาว凸 ดาวแคระขาว และสามารถใช้ในการจำแนกประเภทดาวใน HR-Diagram ได้อย่างมีประสิทธิภาพ



Classification MODEL

Classification MODEL

```
# ลบช่องว่างที่ไม่จำเป็นออกจากชื่อคอลัมน์ทั้งหมด
for_model.columns = for_model.columns.str.strip()

for_model.columns

Index(['Temperature (K)', 'Luminosity(L/Lo)', 'Radius(R/Ro)',
       'Absolute magnitude(Mv)', 'Star type', 'Star color', 'Spectral Class'],
      dtype='object')

# ลบช่องว่างทั้งหมดภายในชื่อคอลัมน์
for_model.columns = for_model.columns.str.replace(' ', '', regex=True)

for_model.columns

Index(['Temperature(K)', 'Luminosity(L/Lo)', 'Radius(R/Ro)',
       'Absolutemagnitude(Mv)', 'Startype', 'Starcolor', 'SpectralClass'],
      dtype='object')
```

1. ลบช่องว่างที่อยู่รอบๆ ชื่อคอลัมน์ หน้า-หลัง
2. ลบช่องว่างทั้งหมดที่อยู่ภายในชื่อคอลัมน์
เพื่อให้เขียนได้ดีง่ายขึ้น

Classification MODEL

กำหนดการของดาว

เลือก Feature ที่เราจะใช้เป็นตัวแปรอิสระ (X) ได้แก่ อุณหภูมิ, ความสว่าง, รัศมี และค่าความสว่างส้มบูรณ์ ส่วนตัวแปรเป้าหมาย (y) คือ เป็นระดับสเปกตรัม แต่เนื่องจากค่าของระดับสเปกตรัม เป็นข้อมูลประเภทตัวอักษร เราจึงต้องแปลงให้เป็นค่าตัวเลขโดยใช้ LabelEncoder() เพื่อให้สามารถใช้กับโมเดลได้ และใช้ Stratified K-Fold Cross Validation โดยกำหนดค่า K เท่ากับ 5 หมายความว่า ข้อมูลจะถูกแบ่งออกเป็น 5 ส่วน เพื่อให้แต่ละคลาสมีสัดส่วนที่สมดุลและช่วยให้ผลลัพธ้มีความแม่นยำมากขึ้น

Classification MODEL

ใช้ 4 โมเดลในการจำแนกประเภทของดาว

- SVM
- Random Forest
- Decision Tree
- Naïve Bayes

คำนวณค่า KPI (Accuracy, Precision, Recall, F1 Score) ด้วย Cross-Validation และสร้าง Confusion

Matrix เพื่อวิเคราะห์ผลการจำแนกประเภท

Classification MODEL

Model	Accuracy	Precision	Recall	F1 Score
SVM	0.504167	0.314626	0.504167	0.384287
Random Forest	0.895833	0.898594	0.895833	0.887469
Decision Tree	0.900000	0.918877	0.900000	0.899762
Naive Bayes	0.404167	0.740868	0.404167	0.419982

Classification MODEL

สรุปผล

- SVM มี Accuracy ต่ำสุด (0.5042) และ F1 Score ต่ำ (0.3843) ซึ่งบ่งบอกว่าอาจไม่เหมาะสมกับชุดข้อมูลนี้
- Naive Bayes มี Accuracy (0.4042) และ F1 Score (0.4200) ต่ำมาก แสดงว่าไม่สามารถจำแนกข้อมูลได้ดี
- Random Forest มี Accuracy สูงถึง 0.8958 และ F1 Score 0.8875 ซึ่งเป็นผลลัพธ์ที่ดีมาก
- Decision Tree มี Accuracy สูงสุด (0.9000) และ F1 Score 0.8998 ซึ่งดีกว่า Random Forest เล็กน้อย

สรุปได้ว่า Decision Tree เป็นโมเดลที่มีประสิทธิภาพดีที่สุด

Member

653020205-1 นางสาวณัชชา ประทุมชาติ

653020206-9 นางสาวณัฐร์กฤตา ไชยโภคภิ

653020567-7 นางสาวกัญร์ตัน อินกร์สำราญ



THANK YOU