

การสร้างโมเดลจัดประเภทราคาโทรศัพท์  
เสนอ

อาจารย์กุลวดี สมบูรณ์วิวัฒน์

จัดทำโดย

นายวิรเทพ รัตนจรัสกุล 6330300798

นางสาวณัฐนันท์ มุลทริพย์ 6330300305



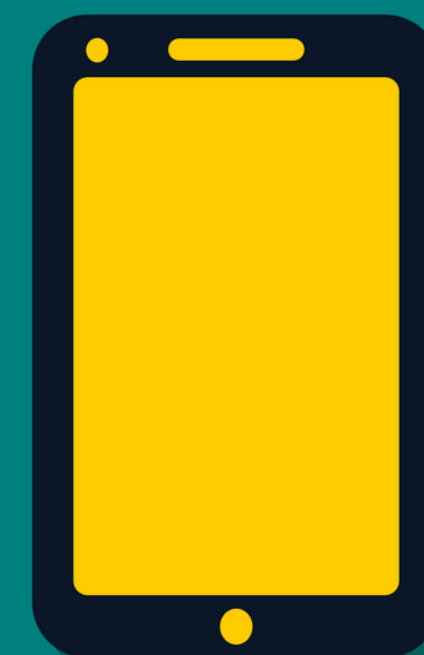
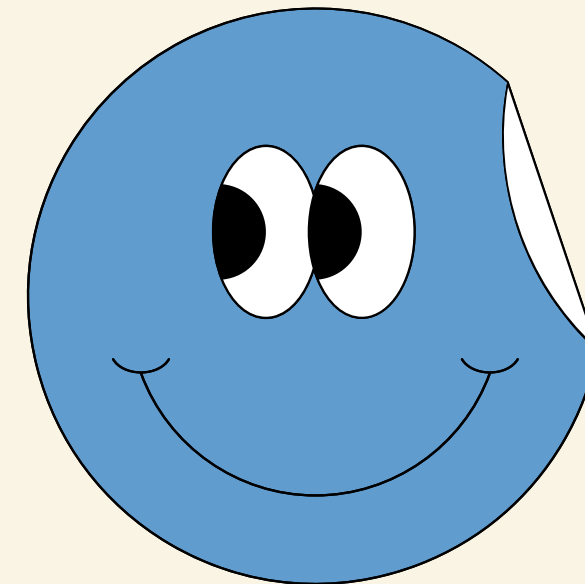
# เป้าหมาย

1. เพื่อเรียนรู้การสร้างโมเดลจัดราคาประเภทโทรศัพท์
2. ทำนายราคาโทรศัพท์ ช่วงราคาที่ระบุว่าราคาสูงเพียงใด

**DATASET : Mobile Price Classification**

**PRICE RANGE :**

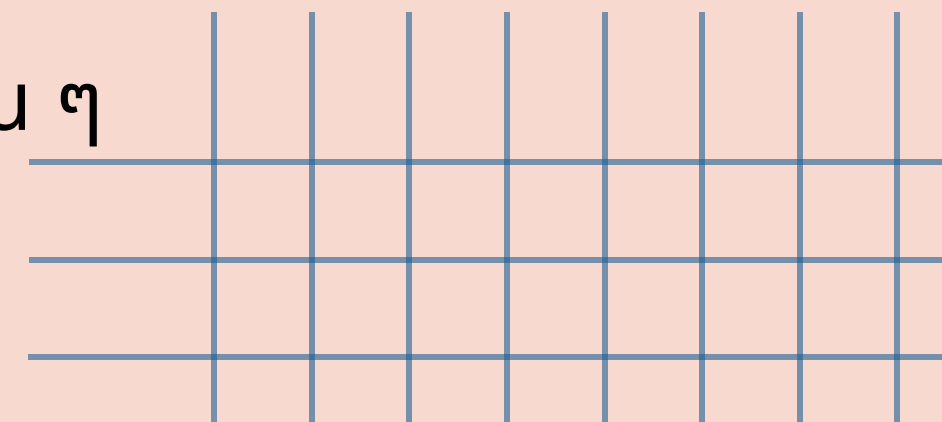
- 0 (low cost)
- 1 (medium cost)
- 2 (high cost)
- 3 (very high cost)



# CLASSIFICATION

## Decision Tree

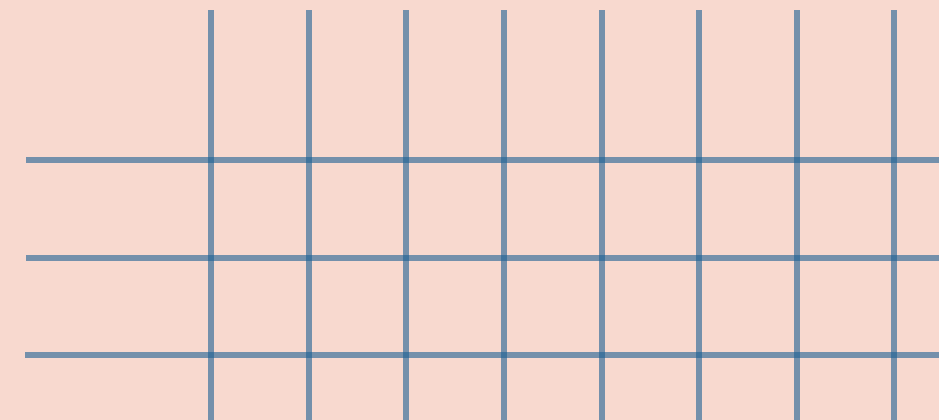
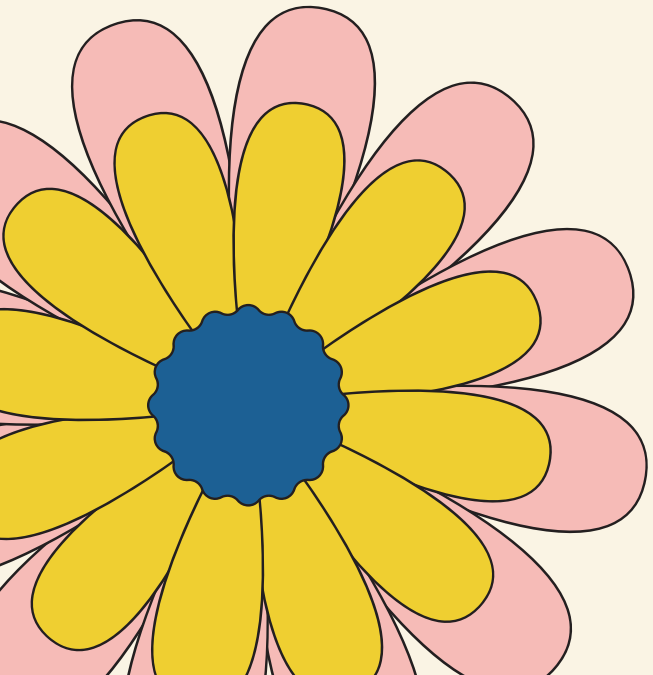
**Model Decision Tree** เป็น Rule-Based Model ที่จะสร้างเงื่อนไข If-else ขึ้นมาจากข้อมูลในตัวแปร เพื่อที่จะแบ่งข้อมูลออกเป็นกลุ่มใหม่ที่สามารถอธิบาย Target ได้ดีที่สุด โดยการสร้างเงื่อนไข If-else ในแต่ละตัวแปร จะถูกกำหนดด้วย Objective Function ซึ่ง Model Decision Tree มี Objective Function อยู่หลายตัว ตามประเภทของ Decision Tree นั้น ๆ



# CLUSTERING

## K-means

**K-means** คือ วิธีการหนึ่งใน Data mining อยู่  
ในกลุ่มของ Unsupervised Learning คือการ  
เรียนรู้แบบไม่ต้องสอน ไม่มีคำตอบตายตัว โดยหน้าที่  
หลักของ K-means คือการแบ่งกลุ่ม แบบ  
Clustering ซึ่งการแบ่งกลุ่มในลักษณะนี้จะใช้พื้น  
ฐานทางสถิติ ซึ่งหน้าที่ของ clustering คือการจับ  
กลุ่มของข้อมูลที่มีลักษณะใกล้เคียงกันเป็นกลุ่ม  
เดียวกัน



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 21 columns):
#   Column                Non-Null Count  Dtype
---  -
0   battery_power         2000 non-null    int64
1   blue                  2000 non-null    int64
2   clock_speed           2000 non-null    float64
3   dual_sim              2000 non-null    int64
4   fc                    2000 non-null    int64
5   four_g                2000 non-null    int64
6   int_memory            2000 non-null    int64
7   m_dep                 2000 non-null    float64
8   mobile_wt             2000 non-null    int64
9   n_cores               2000 non-null    int64
10  pc                    2000 non-null    int64
11  px_height              2000 non-null    int64
12  px_width               2000 non-null    int64
13  ram                   2000 non-null    int64
14  sc_h                  2000 non-null    int64
15  sc_w                  2000 non-null    int64
16  talk_time              2000 non-null    int64
17  three_g                2000 non-null    int64
18  touch_screen           2000 non-null    int64
19  wifi                  2000 non-null    int64
20  price_range            2000 non-null    int64
dtypes: float64(2), int64(19)
memory usage: 328.2 KB
```

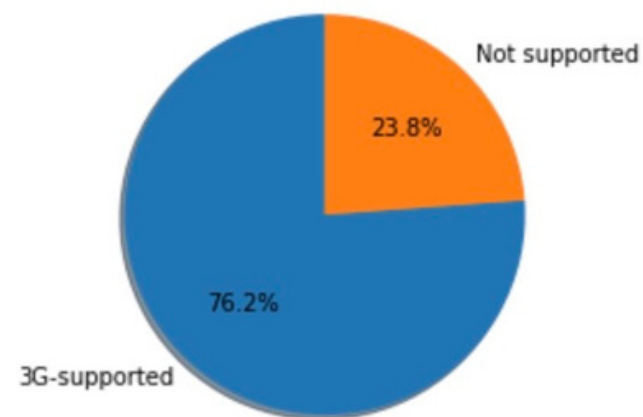
ในโครงการงานนี้ใช้ข้อมูล 21  
คอลัมน์ เช่น พลังงาน  
แบตเตอรี่,เปิดใช้งาน3G,WIFI,  
บลูทูธ, แรม ฯลฯ เรากำลังคาด  
การณ์ช่วงราคาของมือถือ



# Data Visualization & Analysis

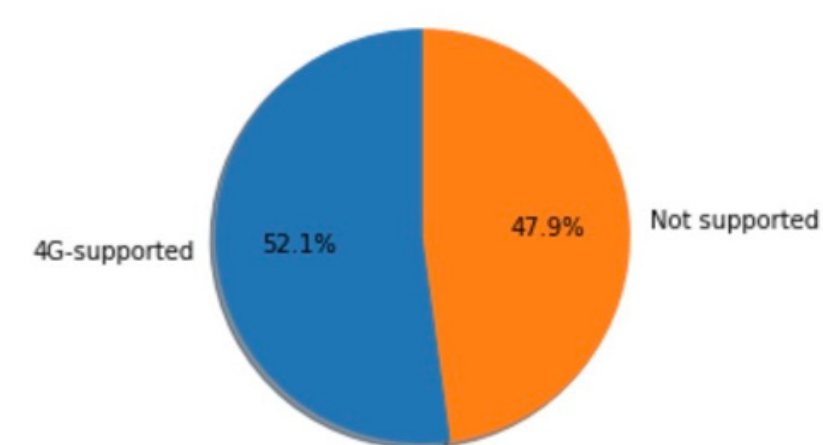
## % ของโทรศัพท์ที่รองรับ 3G

```
▶ labels = ["3G-supported", 'Not supported']  
values=dataset['three_g'].value_counts().values  
fig1, ax1 = plt.subplots()  
ax1.pie(values, labels=labels, autopct='%1.1f%%', shadow=True, startangle=90)  
plt.show()
```



## % ของโทรศัพท์ที่รองรับ 4G

```
▶ labels4g = ["4G-supported", 'Not supported']  
values4g = dataset['four_g'].value_counts().values  
fig1, ax1 = plt.subplots()  
ax1.pie(values4g, labels=labels4g, autopct='%1.1f%%', shadow=True, startangle=90)  
plt.show()
```

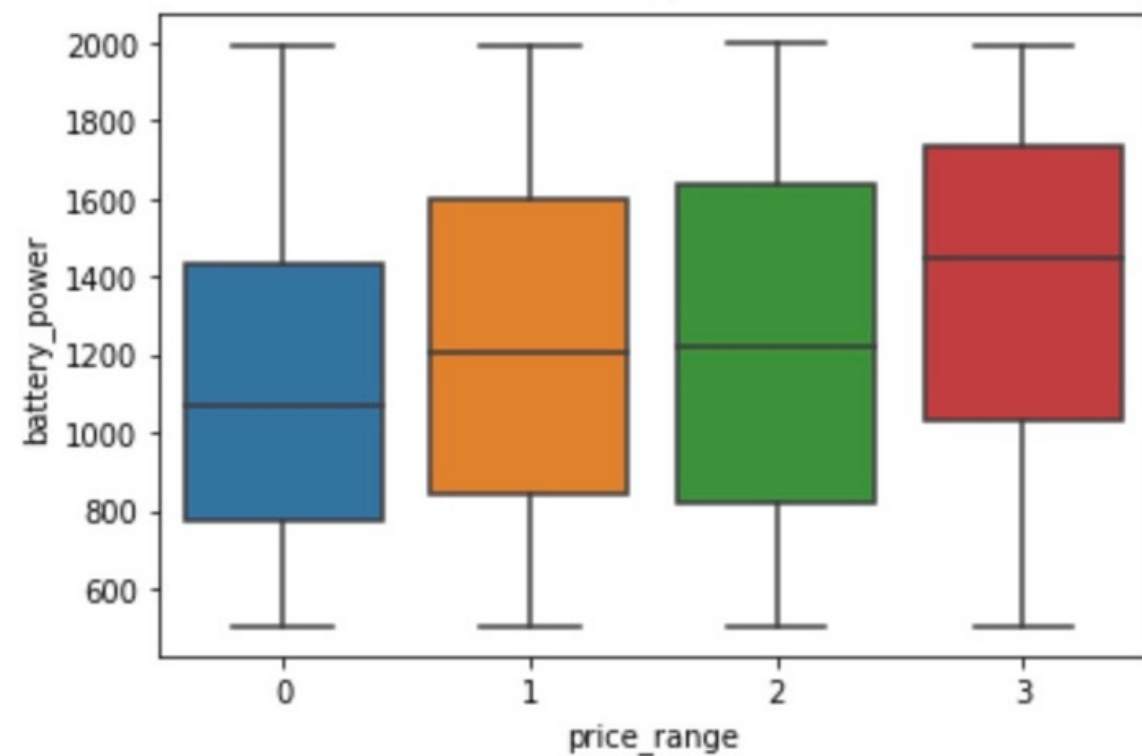


# Data Visualization & Analysis

ความสัมพันธ์ระหว่างพลังงาน  
แบตเตอรี่เทียบกับช่วงราคา

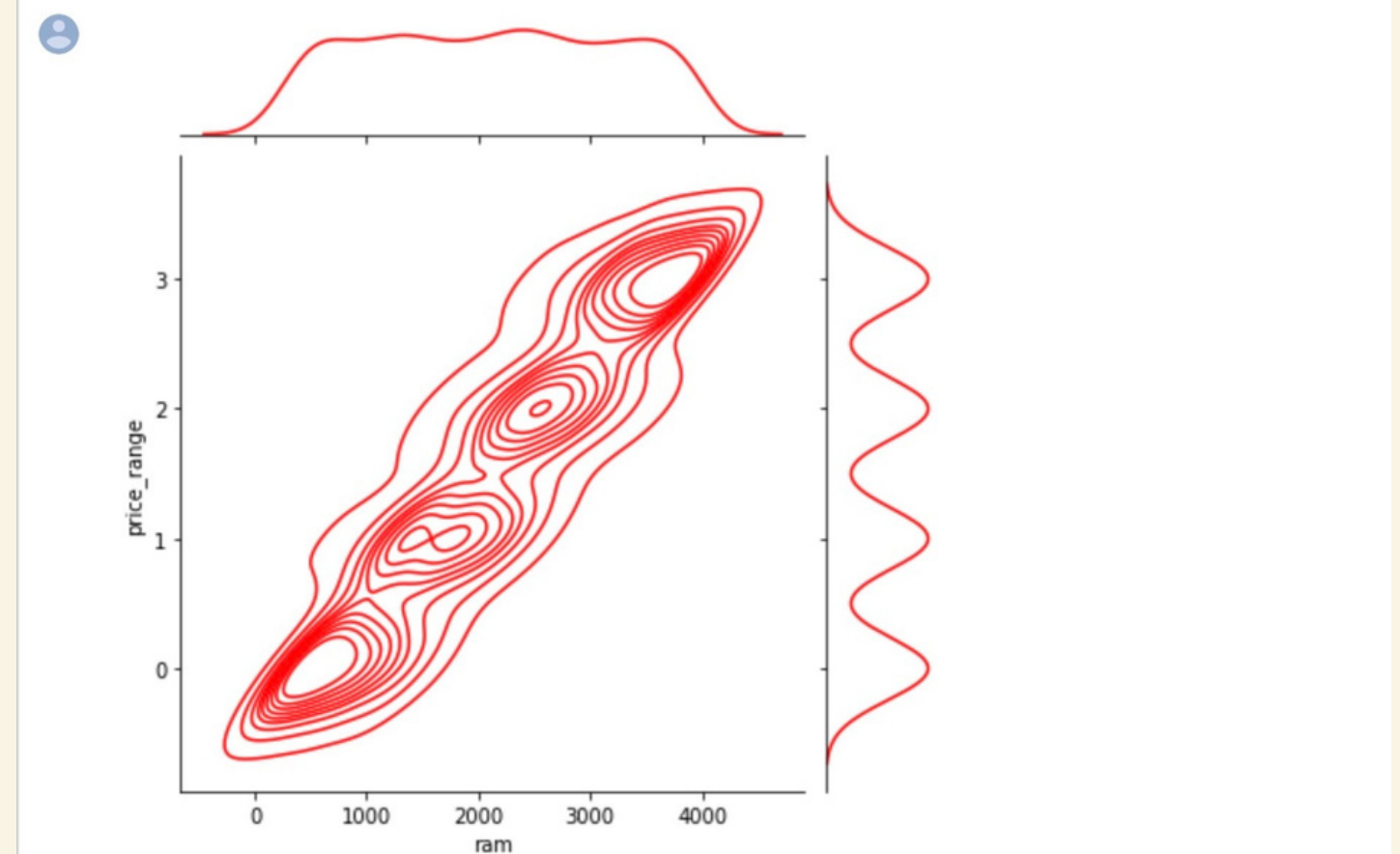
```
sns.boxplot(x="price_range", y="battery_power", data=dataset)
```

```
<AxesSubplot:xlabel='price_range', ylabel='battery_power'>
```



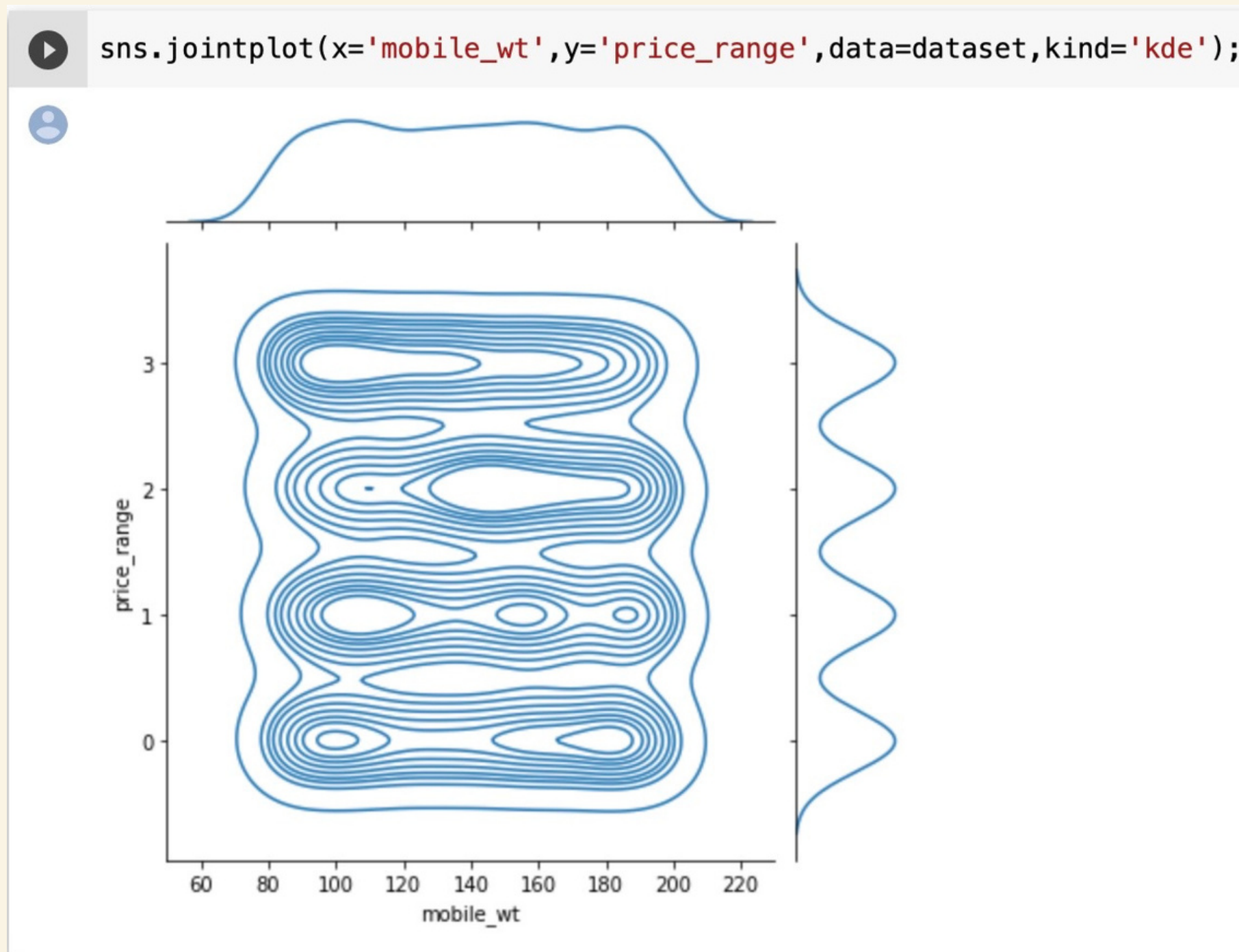
ความสัมพันธ์ระหว่างแรมเทียบกับช่วงราคา

```
sns.jointplot(x='ram', y='price_range', data=dataset, color='red', kind='kde');
```

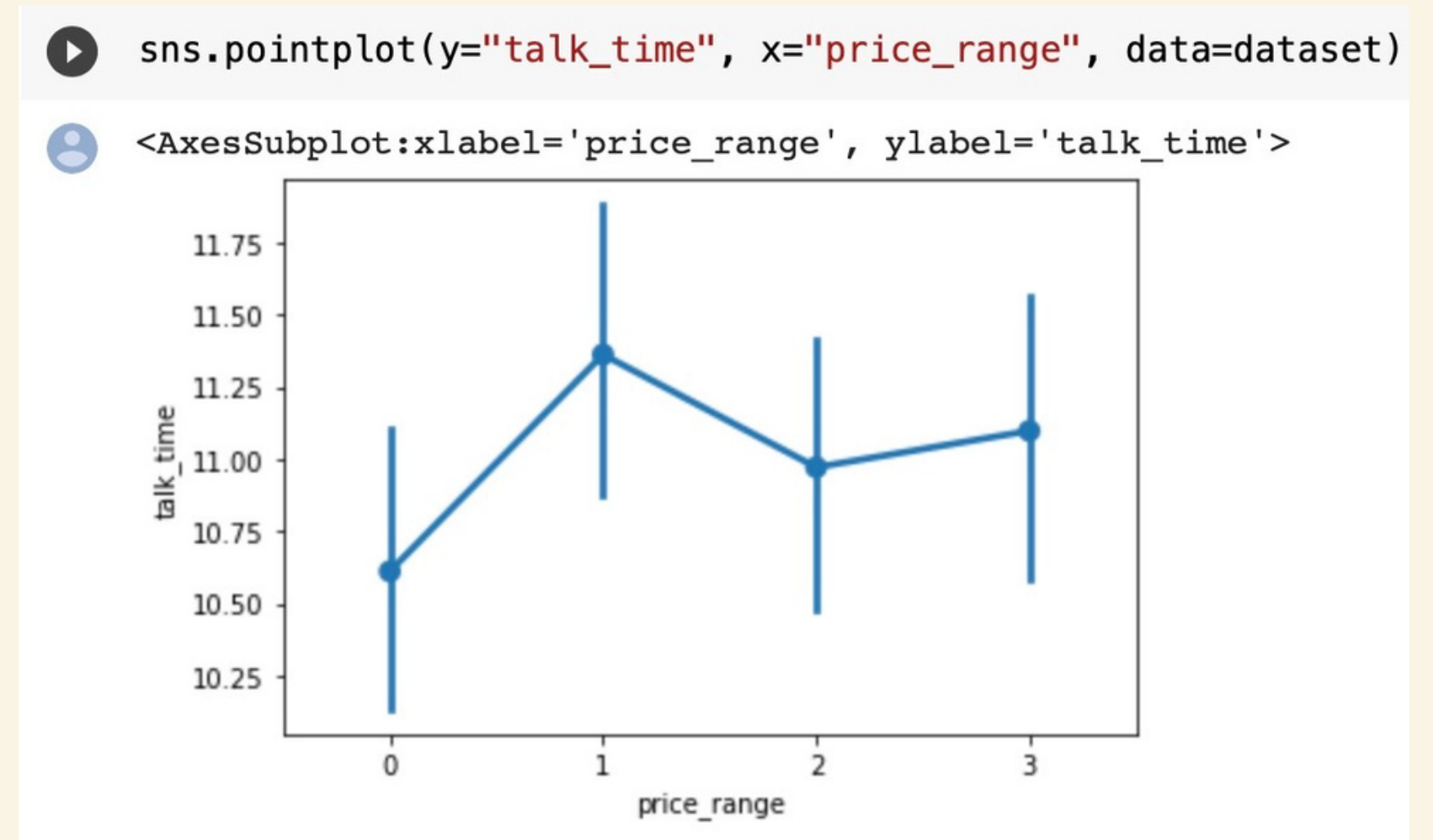


# Data Visualization & Analysis

ความสัมพันธ์ระหว่างน้ำหนักมือถือเทียบกับช่วงราคา



ความสัมพันธ์ระหว่างเวลาสนทนาเทียบกับช่วงราคา





# Result

- **Decision Tree**

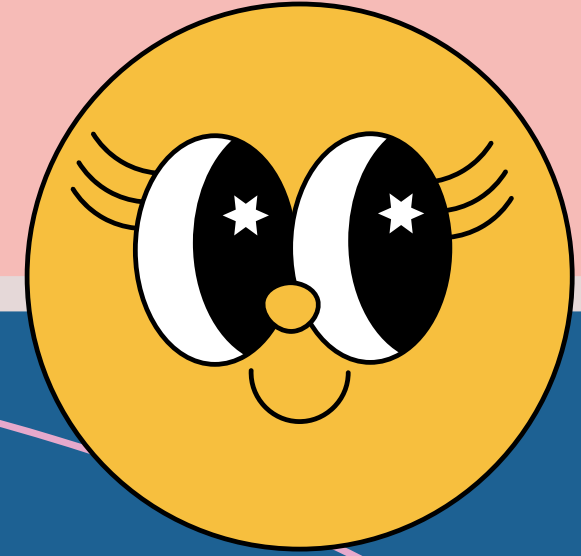
```
[ ] from pyspark.ml.evaluation import MulticlassClassificationEvaluator
    evaluator = MulticlassClassificationEvaluator(labelCol="labelIndex", predictionCol="prediction", metricName="accuracy")
    accuracy = evaluator.evaluate(predictions)
    print("Test Error = %g " % (1.0 - accuracy))
    print("Accuracy = %g " % accuracy)
```

```
Test Error = 0.200965
Accuracy = 0.799035
```

- **K-means**

```
[ ] eval = ClusteringEvaluator()
    silhouette = eval.evaluate(predictions)
    print(f'Silhouette with euclidean distance: {silhouette}')
```

```
Silhouette with euclidean distance: 0.3540456155121568
```



# CONCLUSION

- การคาดคะเนแบบนี้จะช่วยให้บริษัทต่างๆ ประเมินราคาโทรศัพท์มือถือเพื่อให้แข่งขันกับผู้ผลิตมือถือรายอื่นได้
- นอกจากนี้ จะเป็นประโยชน์สำหรับผู้บริโภคในการยืนยันว่าพวกเขากำลังจ่ายในราคาที่ดีที่สุดสำหรับมือถือ