การนำเข้าข้อมูลและการเตรียมข้อมูล

- โดยในชุดข้อมูลประกอบไปด้วย feature ดังนี้

features	Description
เพศ	ระบุเพศ
อายุ	ระบุอายุ
เคยมีแฟนมาแล้ว (คน)	จำนวนแฟนที่เคยมีมาแล้ว
จำนวนครั้งที่ไปออกกำลังกายต่อสัปดาห์	จำนวนครั้งความถี่ในการออกกาลังกายในหนึ่งสัปดาห์
ระยะเวลาการนอน	ช่วงเวลาการนอนหลับ
นิสัยการกินอาหาร	ประเภทอาหารที่ชอบรับประทาน
เคยคิดฆ่าตัวตาย	ระบุว่าเคยมีความคิดฆ่าตัวตายหรือไม่
จำนวนชั่วโมงที่เข้าห้องสมุดในหนึ่งสัปดาห์	ระบุเวลาที่ใช้ในห้องสมุดต่อสัปดาห์
ระดับความเครียดด้านการเงิน	ระดับความเครียดเกี่ยวกับการเงิน
เคยกินยานอนหลับ	ระบุว่าเคยใช้ยานอนหลับหรือไม่
ภาวะซึมเศร้า	สถานะภาวะซึมเศร้า

- แสดงชุดข้อมูลที่นำเข้ามา

	เพศ	อายุ	เคยมีแฟนมาแล้ว (คน)	จำนวนครั้งที่ไปออกกำลังกายต่อสัปดาห์	ระยะเวลาการนอน	นิสัยการกินอาหาร	เคยคิดฆ่าตัวตาย	จำนวนชั่วโมงที่เข้าห้องสมุดในหนึ่งสัปดาห์	ระดับความเครียดด้านการเงิน	เคยกินยานอนหลับ	ภาวะซึมเศร้า
0	Male	28	5	3	5-6 ชั่วโมง	อาหารสุขภาพ	Yes	8	3	Yes	Yes
1	Male	23	5	2	มากกว่า 8 ชั่วโมง	อาหารทั่วไป	No	10	4	No	Yes
2	Female	23	1	3	น้อยกว่า 5 ชั่วโมง	อาหารสุขภาพ	Yes	0	3	No	No
3	Female	20	5	5	มากกว่า 8 ชั่วโมง	Junkfood	Yes	2	5	No	Yes
4	Male	29	4	3	มากกว่า 8 ชั่วโมง	Junkfood	Yes	1	3	No	Yes

- การแสดงรายละเอียดของชุดข้อมูล

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 502 entries, 0 to 501
Data columns (total 11 columns):
 #   Column                                    Non-Null Count  Dtype
---  ------                                    --------------  -----
 0   เพศ                                       502 non-null    object
 1   อายุ                                      502 non-null    int64
 2   เคยมีแฟนมาแล้ว  (คน)                      502 non-null    object
 3   จำนวนครั้งที่ไปออกกำลังกายต่อสัปดาห์      502 non-null    object
 4   ระยะเวลาการนอน                            502 non-null    object
 5   นิสัยการกินอาหาร                          502 non-null    object
 6   เคยคิดฆ่าตัวตาย                           502 non-null    object
 7   จำนวนชั่วโมงที่เข้าห้องสมุดในหนึ่งสัปดาห์  502 non-null    object
 8   ระดับความเครียดด้านการเงิน                502 non-null    object
 9   เคยกินยานอนหลับ                           502 non-null    object
 10  ภาวะซึมเศร้า                              502 non-null    object
dtypes: int64(1), object(10)
memory usage: 43.3+ KB
```

- แสดงรายละเอียดของ features target

```
ภาวะซึมเศร้า
Yes    252
No     250
Name: count, dtype: int64
```

- แปลงข้อมูลที่ไม่เป็นตัวเลขให้เป็นค่าที่เหมาะสม โดย Label Encoding และตรวจสอบข้อมูลที่หายไปหรือไม่สมบูรณ์ แล้วแสดงชุดข้อมูลอีกครั้งหลังจากทำการแปลงข้อมูล

	เพศ	อายุ	เคยมีแฟนมาแล้ว (คน)	จำนวนครั้งที่ไปออกกำลังกายต่อสัปดาห์	ระยะเวลาการนอน	นิสัยการกินอาหาร	เคยคิดฆ่าตัวตาย	จำนวนชั่วโมงที่เข้าห้องสมุดในหนึ่งสัปดาห์	ระดับความเครียดด้านการเงิน	เคยกินยานอนหลับ	ภาวะซึมเศร้า
0	1	28	5.0	3.0	0	2	1	8.0	3.0	1	1
1	1	23	5.0	2.0	3	1	0	10.0	4.0	0	1
2	0	23	1.0	3.0	2	2	1	0.0	3.0	0	0
3	0	20	5.0	5.0	3	0	1	2.0	5.0	0	1
4	1	29	4.0	3.0	3	0	1	1.0	3.0	0	1

การทำนายผล

- การทำนายครั้งแรก

```
#แยก features และ target
col_names = patient.columns.tolist()
feature_cols = col_names[:-1]
X = patient[feature_cols]
y = patient['ภาวะซึมเศร้า']

# แบ่งข้อมูลออกเป็นชุดฝึก (train) และชุดทดสอบ (test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)

# ปรับมาตรฐาน
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# สร้างและฝึกโมเดล SVM
clf = svm.SVC(gamma='scale', kernel='rbf', C=1.0, random_state=42)
clf.fit(X_train, y_train)

# ทำนายผลและประเมินโมเดล
y_pred = clf.predict(X_test)

print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.9290780141843972
              precision    recall  f1-score   support

           0       0.92      0.94      0.93        70
           1       0.94      0.92      0.93        71

    accuracy                           0.93       141
   macro avg       0.93      0.93      0.93       141
weighted avg       0.93      0.93      0.93       141
```

- ลองใช้ GridSearchCV โดยจะค้นหาค่าที่ดีที่สุดสำหรับพารามิเตอร์โดยการลองค่าที่กำหนดในแต่ละตัวเลือก

```python
# กำหนดค่าพารามิเตอร์สำหรับ Grid Search
param_grid = {
    'C': [0.1, 1, 10, 100],          # ค่าพารามิเตอร์ C
    'gamma': [1, 0.1, 0.01, 0.001], # ค่าพารามิเตอร์ gamma
    'kernel': ['rbf', 'linear']      # ประเภท kernel
}

# สร้าง GridSearchCV โดยกำหนดจำนวน cross-validation folds
grid = GridSearchCV(
    estimator=SVC(),          # โมเดล SVM
    param_grid=param_grid,    # ค่าพารามิเตอร์ที่ต้องการค้นหา
    refit=True,               # เลือกโมเดลที่ดีที่สุด
    verbose=0,                # แสดง log ระหว่างการรัน
    cv=5                      # ใช้ 5-fold cross-validation
)

# ฝึกโมเดลโดยใช้ GridSearchCV
grid.fit(X_train, y_train)

# แสดงพารามิเตอร์ที่ดีที่สุด
print("Best Parameters:", grid.best_params_)

# ใช้โมเดลที่ดีที่สุดเพื่อทำนาย
y_pred = grid.best_estimator_.predict(X_test)

# แสดงผลการวิเคราะห์
print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

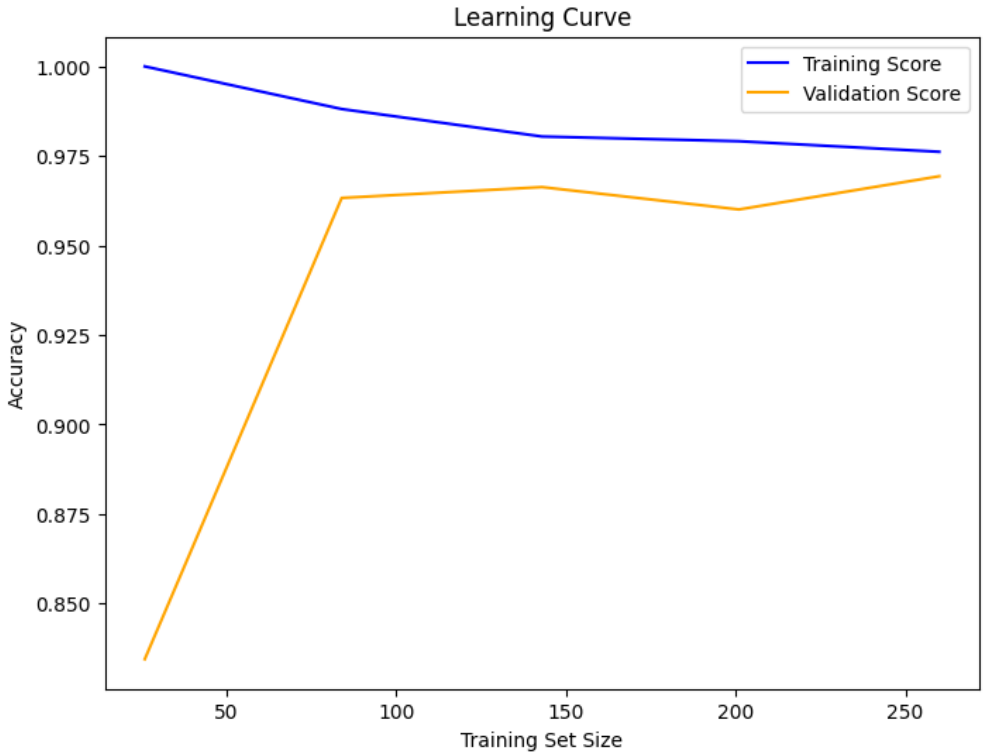
```
Best Parameters: {'C': 1, 'gamma': 1, 'kernel': 'linear'}
Accuracy: 0.9574468085106383
              precision    recall  f1-score   support

           0       0.94      0.97      0.96        70
           1       0.97      0.94      0.96        71

    accuracy                           0.96       141
   macro avg       0.96      0.96      0.96       141
weighted avg       0.96      0.96      0.96       141
```

และมีการแสดงผลลัพธ์ทั้งหมดที่ได้จากการค้นหา

	param_C	param_gamma	param_kernel	mean_test_score	rank_test_score
0	0.1	1.000	rbf	0.512261	31
1	0.1	1.000	linear	0.960093	7
2	0.1	0.100	rbf	0.923310	25
3	0.1	0.100	linear	0.960093	7
4	0.1	0.010	rbf	0.717995	29
5	0.1	0.010	linear	0.960093	7
6	0.1	0.001	rbf	0.512261	31
7	0.1	0.001	linear	0.960093	7
8	1.0	1.000	rbf	0.693380	30
9	1.0	1.000	linear	0.969324	1
10	1.0	0.100	rbf	0.938695	20
11	1.0	0.100	linear	0.969324	1
12	1.0	0.010	rbf	0.935618	22
13	1.0	0.010	linear	0.969324	1
14	1.0	0.001	rbf	0.865082	26
15	1.0	0.001	linear	0.969324	1
16	10.0	1.000	rbf	0.726993	27
17	10.0	1.000	linear	0.960047	12
18	10.0	0.100	rbf	0.935478	23
19	10.0	0.100	linear	0.960047	12
20	10.0	0.010	rbf	0.963124	6
21	10.0	0.010	linear	0.960047	12
22	10.0	0.001	rbf	0.938648	21
23	10.0	0.001	linear	0.960047	12
...					
28	100.0	0.010	rbf	0.960093	7
29	100.0	0.010	linear	0.957063	16
30	100.0	0.001	rbf	0.966200	5
31	100.0	0.001	linear	0.957063	16

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

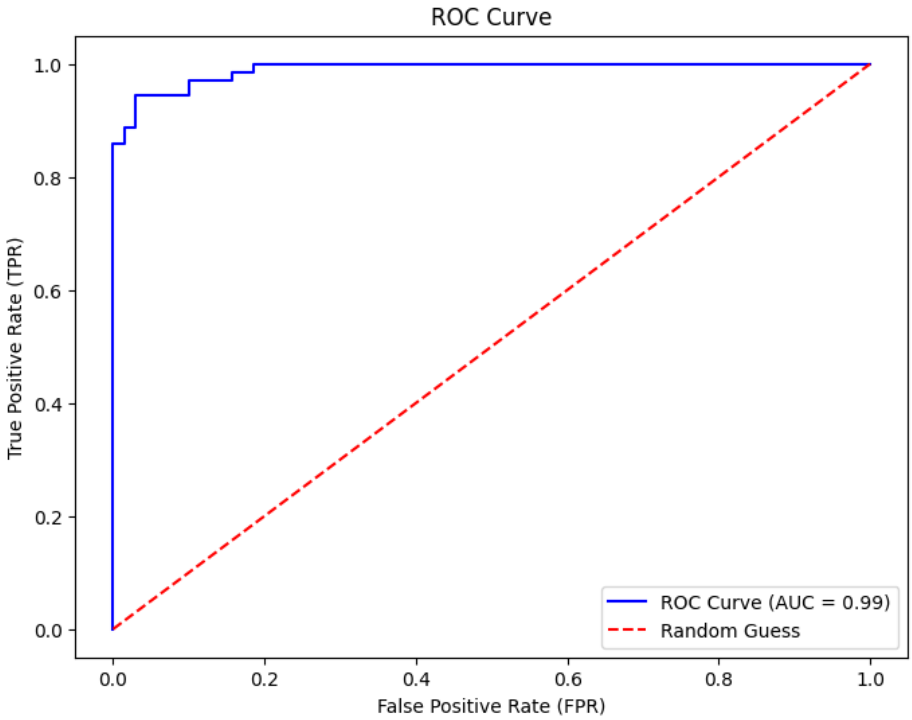
- แสดง Learning Curve



1. **แกน X** แสดงจำนวนข้อมูลที่ใช้ในการฝึก (Training Set) ซึ่งเพิ่มขึ้นเรื่อย ๆ จากซ้ายไปขวา
    - เริ่มจากใช้ข้อมูลจำนวนน้อย (เช่น 50 ตัวอย่าง) ไปจนถึงข้อมูลทั้งหมดในชุดฝึก (เช่น 250 ตัวอย่าง)
2. **แกน Y** แสดงค่าความแม่นยำ (Accuracy) ของโมเดล ซึ่งมีค่าระหว่าง 0 ถึง 1.0
    - ค่าใกล้ 1.0 หมายถึงโมเดลมีความแม่นยำสูง
    - ค่าใกล้ 0 หมายถึงโมเดลมีความแม่นยำต่ำ
3. **เส้นสีน้ำเงิน (Training Score)** แสดงค่าความแม่นยำของโมเดลบนชุดข้อมูลฝึก (Training Set)
    - ในช่วงเริ่มมีค่าใกล้ 1.0 เพราะโมเดลสามารถจดจำข้อมูลฝึกทั้งหมดได้แต่เริ่มลดลงเล็กน้อย เพราะข้อมูลฝึกเพิ่มขึ้น ทำให้โมเดลต้องปรับตัวให้เหมาะสมมากขึ้น ซึ่งยังคงอยู่ในระดับสูง (>0.95) หมายความว่าโมเดลสามารถเรียนรู้ข้อมูลฝึกได้ดี
4. **เส้นสีส้ม (Validation Score)** แสดงค่าความแม่นยำของโมเดลบนชุดข้อมูล (Validation Set)
    - ค่าอยู่ในระดับต่ำ (<0.9) เพราะโมเดลไม่สามารถ generalize กับข้อมูลในชุดตรวจสอบเนื่องจากข้อมูลฝึกน้อยเกินไป จากนั้นเริ่มเพิ่มสูงขึ้น เพราะโมเดลเรียนรู้จากข้อมูลที่หลากหลายขึ้น และใกล้กับเส้น Training Score แสดงว่าโมเดล generalize ได้ดีในชุดตรวจสอบ

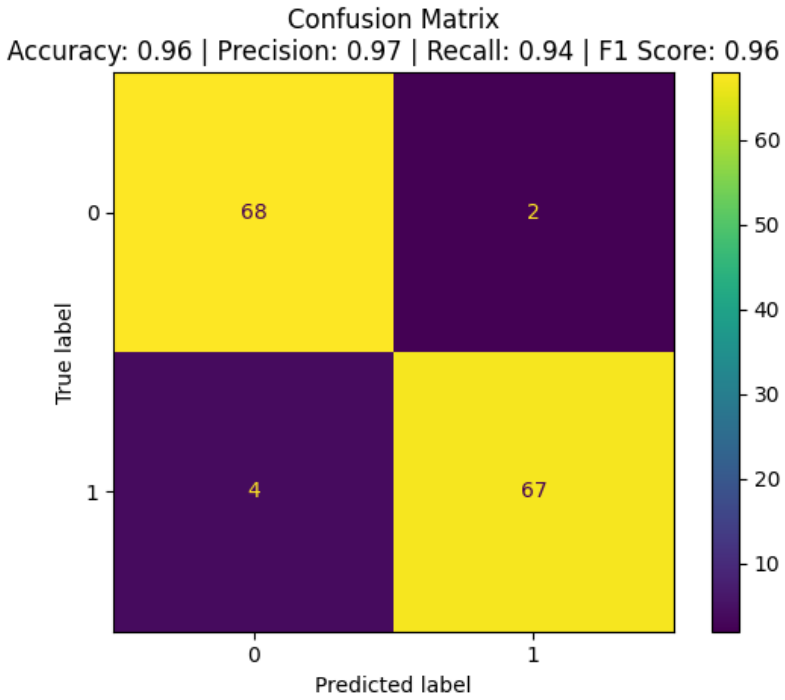
โดยจากกราฟ ช่องว่างระหว่างสองเส้นลดลง แสดงว่าโมเดลไม่ได้ Overfit หรือ Underfit

- แสดง ROC Curve (Receiver Operating Characteristic Curve) และ AUC (Area Under Curve) ซึ่งเป็นเครื่องมือที่ใช้สำหรับการประเมินประสิทธิภาพของโมเดล Classification



1. **แกน X** แสดงอัตราการทำนายผิด (False Positive Rate) โดยค่าอยู่ระหว่าง 0 ถึง 1
   - คำนวณจาก $FRP = \frac{\text{จำนวน } \boldsymbol{False\ Positive}}{\text{จำนวน } \boldsymbol{True\ Negative + Flase\ Positives}}$
2. **แกน Y** แสดงอัตราการทำนายถูก (True Positive Rate) หรือ Recall โดยค่าอยู่ระหว่าง 0 ถึง 1
   - คำนวณจาก $FRP = \frac{\text{จำนวน } \boldsymbol{True\ Positive}}{\text{จำนวน } \boldsymbol{True\ Positive + Flase\ Negatives}}$
3. **เส้นสีน้ำเงิน** เป็นเส้น ROC Curve แสดงความสัมพันธ์ระหว่าง FPR และ TPR ของโมเดล โดยกราฟโค้งสูงขึ้นและเข้าใกล้มุมบนซ้ายของกราฟอย่างชัดเจน แสดงว่าโมเดลสามารถทำนายกลุ่ม Positive ได้ถูกต้องเกือบทั้งหมด ในขณะที่ยังคงรักษา FPR ให้ต่ำมาก
4. **เส้นประสีแดง** เป็นเส้นมาตรฐานสำหรับการทำนายแบบสุ่ม (Random Guess) ซึ่ง กราฟ ROC Curve สีน้ำเงินห่างจากเส้นทแยงมุมสีแดงอย่างชัดเจน แสดงว่าโมเดลมีความแม่นยำดีกว่าการเดาสุ่ม
5. **ค่า AUC (Area Under the Curve)** พื้นที่ใต้กราฟ ROC ซึ่งบอกถึงประสิทธิภาพโดยรวมของโมเดลซึ่งโมเดลนี้มีค่า AUC = 0.99 หมายถึงว่าโมเดลนี้มีประสิทธิภาพสูงมากสามารถแยกระหว่างกลุ่ม Positive และ Negative ได้ดีมาก

- แสดงผลการทำนายในรูปแบบ Confusion Matrix



แกน X (Predicted) แสดงค่าผลลัพธ์ที่โมเดลทำนาย (0 หรือ 1)
แกน Y (Actual) แสดงค่าจริงที่เกิดขึ้นในชุดข้อมูล (0 หรือ 1)
โดยค่าตัวเลขในแต่ละช่องแสดงจำนวนตัวอย่างที่โมเดลจัดให้อยู่ในกลุ่มนั้น
ช่อง (True Negative) – 68 ตัวอย่างที่โมเดลทำนายว่าเป็นคลาส 0 และค่าจริงคือคลาส 0
ช่อง (False Positive) – 2 ตัวอย่างที่โมเดลทำนายว่าเป็นคลาส 1 และค่าจริงคือคลาส 0
ช่อง (False Negative) – 4 ตัวอย่างที่โมเดลทำนายว่าเป็นคลาส 0 และค่าจริงคือคลาส 1
ช่อง (True Positives) – 67 ตัวอย่างที่โมเดลทำนายว่าเป็นคลาส 1 และค่าจริงคือคลาส 1

Accuracy (ความแม่นยำของโมเดล): 0.96
Precision (ความถูกต้องของการทำนายคลาส 1): 0.97
Recall (ความครอบคลุมของการทำนายคลาส 1): 0.94
F1 Score (คะแนนเฉลี่ยถ่วงน้ำหนักของ Precision และ Recall): 0.83