

การนำเสนอและอธิบายการทำงานของโค้ด

1. โหลดและตรวจสอบข้อมูล

เริ่มต้นด้วยการโหลดข้อมูลจากไฟล์ CSV (เลือกจะทำเรื่อง

การทำนาย

โรคเบาหวาน) และตรวจสอบข้อมูลเบื้องต้นเพื่อตรวจสอบว่าข้อมูล
มีลักษณะอย่างไร และมีค่าว่างหรือไม่

2. ตรวจสอบข้อมูลที่ขาดหาย

ตรวจสอบว่ามีค่าว่างในข้อมูลหรือไม่

หากมีค่าว่างเราจะทำการจัดการค่าว่างนั้น

3. การจัดการกับค่าว่าง

แทนที่ค่า 0 ในคอลัมน์ที่ควรมีค่าที่ไม่เป็น 0

ด้วยค่ามัธยฐานของแต่ละคอลัมน์เพื่อป้องกันปัญหาที่อาจเกิดขึ้นจ
ากข้อมูลที่ขาดหาย

4. การบันทึกข้อมูลที่ preprocessing

เสร็จแล้วเป็นไฟล์ใหม่

บันทึกข้อมูลที่ผ่านการ preprocessing

แล้วลงในไฟล์ใหม่เพื่อใช้ในการสร้างโมเดล

```
import pandas as pd

# 1. โหลดและตรวจสอบข้อมูล
file_path = r'C:\Users\Windows 11\Desktop\ML\MLactivity03-03\diabetes.csv'
df = pd.read_csv(file_path)
print("ข้อมูลเบื้องต้น:")
print(df.head())

# 2. ตรวจสอบข้อมูลที่ขาดหาย
print("\nการตรวจสอบค่าว่าง:")
print(df.isnull().sum())

# 3. การจัดการกับค่าว่าง (ถ้ามี)
# แทนที่ค่า 0 ในคอลัมน์ที่ควรมีค่าที่ไม่เป็น 0
df[['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI']] = df[['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI']].replace(0, pd.NA)

# แทนที่ค่า NA ด้วยค่ามัธยฐานของแต่ละคอลัมน์
df.fillna(df.median(), inplace=True)

print("\nข้อมูลหลังจากการจัดการค่าว่าง:")
print(df.head())

# 4. การบันทึกข้อมูลที่ preprocessing เสร็จแล้วเป็นไฟล์ใหม่
output_file_path = r'C:\Users\Windows 11\Desktop\ML\MLactivity03-03\diabetes_preprocessed.csv'
df.to_csv(output_file_path, index=False)
print(f"บันทึกไฟล์ที่ preprocessing เสร็จแล้วเป็นไฟล์ใหม่ที่ {output_file_path}")
```

ข้อมูล เบื้องต้น:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

การตรวจสอบค่าว่าง :

```
Pregnancies      0
Glucose           0
BloodPressure     0
SkinThickness     0
Insulin           0
BMI               0
DiabetesPedigreeFunction  0
Age               0
Outcome           0
dtype: int64
```

ข้อมูลหลังจากการจัดการค่าว่าง :

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148.0	72.0	35.0	125.0	33.6	0.627	50	1
0	6	148.0	72.0	35.0	125.0	33.6	0.627	50	1
1	1	85.0	66.0	29.0	125.0	26.6	0.351	31	0
2	8	183.0	64.0	29.0	125.0	23.3	0.672	32	1
1	1	85.0	66.0	29.0	125.0	26.6	0.351	31	0
2	8	183.0	64.0	29.0	125.0	23.3	0.672	32	1
3	1	89.0	66.0	23.0	94.0	28.1	0.167	21	0
2	8	183.0	64.0	29.0	125.0	23.3	0.672	32	1
3	1	89.0	66.0	23.0	94.0	28.1	0.167	21	0
3	1	89.0	66.0	23.0	94.0	28.1	0.167	21	0
4	0	137.0	40.0	35.0	168.0	43.1	2.288	33	1
4	0	137.0	40.0	35.0	168.0	43.1	2.288	33	1

บันทึกไฟล์ที่ preprocessing เสร็จแล้วเป็นไฟล์ใหม่ที่ C:\Users\Windows 11\Desktop\ML\MLactivity03-03\diabetes_preprocessed.csv
บันทึกไฟล์ที่ preprocessing เสร็จแล้วเป็นไฟล์ใหม่ที่ C:\Users\Windows 11\Desktop\ML\MLactivity03-03\diabetes_preprocessed.csv

5. การสร้างและฝึกโมเดลทั้ง 6 แบบ

แบ่งข้อมูลเป็น Training และ Test set

จากนั้นสร้างและฝึกโมเดล Machine Learning ทั้ง 6
แบบ ประกอบด้วย Logistic Regression, Naive
Bayes, Decision Tree, Support Vector
Machine (SVM), K-Nearest Neighbors
(KNN), และ Artificial Neural Network
(ANN)

6. สรุปผลการประเมินโมเดล

แสดงความแม่นยำของแต่ละโมเดลเพื่อเปรียบเทียบประสิทธิภาพ
และเลือกโมเดลที่ดีที่สุด

```

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from tensorflow.keras.models import Sequential      Import "tensorflow.keras.models" could not be resolved
from tensorflow.keras.layers import Dense          Import "tensorflow.keras.layers" could not be resolved

# โหลดข้อมูลที่ preprocessing แล้ว
df = pd.read_csv(r'C:\Users\Windows 11\Desktop\ML\MLactivity03-03\diabetes_preprocessed.csv')

# แบ่งข้อมูลเป็น Training และ Test set
X = df.drop('Outcome', axis=1)
y = df['Outcome']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

```

5. การสร้างและฝึกโมเดลทั้ง 6 แบบ

5.1 Logistic Regression

```

log_reg = LogisticRegression()
log_reg.fit(X_train, y_train)
y_pred_log = log_reg.predict(X_test)
accuracy_log = accuracy_score(y_test, y_pred_log)
print('Logistic Regression Accuracy:', accuracy_log)

```

5.2 Naive Bayes

```

nb = GaussianNB()
nb.fit(X_train, y_train)
y_pred_nb = nb.predict(X_test)
accuracy_nb = accuracy_score(y_test, y_pred_nb)
print('Naive Bayes Accuracy:', accuracy_nb)

```

5.3 Decision Tree

```

tree = DecisionTreeClassifier()
tree.fit(X_train, y_train)
y_pred_tree = tree.predict(X_test)
accuracy_tree = accuracy_score(y_test, y_pred_tree)
print('Decision Tree Accuracy:', accuracy_tree)

```

```
# 5.4 Support Vector Machine (SVM)
svm = SVC()
svm.fit(X_train, y_train)
y_pred_svm = svm.predict(X_test)
accuracy_svm = accuracy_score(y_test, y_pred_svm)
print('SVM Accuracy:', accuracy_svm)

# 5.5 K-Nearest Neighbors (KNN)
knn = KNeighborsClassifier()
knn.fit(X_train, y_train)
y_pred_knn = knn.predict(X_test)
accuracy_knn = accuracy_score(y_test, y_pred_knn)
print('KNN Accuracy:', accuracy_knn)
```

```
# 5.6 Artificial Neural Network (ANN)
ann = Sequential()
ann.add(Dense(12, input_dim=X_train.shape[1], activation='relu'))
ann.add(Dense(8, activation='relu'))
ann.add(Dense(16, activation='relu'))
ann.add(Dense(32, activation='relu'))
ann.add(Dense(64, activation='relu'))
ann.add(Dense(128, activation='relu'))
ann.add(Dense(256, activation='relu'))
ann.add(Dense(1, activation='sigmoid'))

ann.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
ann.fit(X_train, y_train, epochs=100, batch_size=20, verbose=0)
y_pred_ann = (ann.predict(X_test) > 0.5).astype(int)
accuracy_ann = accuracy_score(y_test, y_pred_ann)
print('ANN Accuracy:', accuracy_ann)

# สรุปผลการประเมินโมเดล
print("\nสรุปผลการประเมินโมเดล:")
print(f'Logistic Regression Accuracy: {accuracy_log}')
print(f'Naive Bayes Accuracy: {accuracy_nb}')
print(f'Decision Tree Accuracy: {accuracy_tree}')
print(f'SVM Accuracy: {accuracy_svm}')
print(f'KNN Accuracy: {accuracy_knn}')
print(f'ANN Accuracy: {accuracy_ann}')
```

สรุปผลการประเมินโมเดล:

Logistic Regression Accuracy: 0.7727272727272727

Naive Bayes Accuracy: 0.7532467532467533

Decision Tree Accuracy: 0.7142857142857143

SVM Accuracy: 0.7662337662337663

KNN Accuracy: 0.6753246753246753

ANN Accuracy: 0.7142857142857143

ผลโดยสรุปจากการทำนายโรคเบาหวานโมเดลแบบ

Logistic Regression มีความแม่นยำสูงที่สุดเหมาะ

สำหรับการทำนายของชุดข้อมูลนี้