ใบงานที่ 3 การจัดการและเตรียมข้อมูลสำหรับสร้างโมเดล AI

วัตถุประสงค์การเรียนรู้

1. เขียนคำสั่ง Python เพื่อเตรียมข้อมูล (Data Preprocessing) ซึ่งครอบคลุมการทำความสะอาด ปรับสเกล แปลงข้อมูลเชิงหมวดหมู่ และแบ่งชุดข้อมูลได้
2. อธิบายผลลัพธ์และค่าสถิติที่เกิดจากการเปลี่ยนแปลงของข้อมูลในแต่ละขั้นตอนของการเตรียมข้อมูลได้

เครื่องมือและอุปกรณ์

1. เครื่องคอมพิวเตอร์ที่ติดตั้งโปรแกรม PyCharm IDE
2. ไฟล์ข้อมูลดิบ lab1.csv
3. เอกสารใบความรู้หน่วยที่ 3
4. แบบฟอร์มบันทึกผลการปฏิบัติงาน (ท้ายใบงาน)

ลำดับขั้นตอนการปฏิบัติงาน

คำชี้แจง: ให้ผู้เรียนเขียนโค้ดลงในไฟล์ main.py ตามขั้นตอนต่อไปนี้

```
# ขั้นตอนที่ 1: การทำความสะอาดข้อมูล (Data Cleaning)
# 1. นำไฟล์ lab1.csv จากโฟลเดอร์ Data management มาเป็น Dataset สำหรับ AI ทำนายราคารถ
# 2. เขียนคำสั่ง นำเข้าไลบรารี Pandas และโหลดไฟล์ CSV เข้าสู่ DataFrame
import pandas as pd
df = pd.read_csv('lab1.csv')
# 3. เขียนคำสั่ง แปลงรูปแบบข้อมูล (Data Formatting)
# - แปลงคอลัมน์ราคา (Price) จากตัวอักษรให้เป็นตัวเลข
df['Price'] = df['Price'].astype(str).str.replace(',', '')
df['Price'] = pd.to_numeric(df['Price'], errors='coerce')
# - กำจัดช่องว่างและจัดรูปแบบตัวอักษรในคอลัมน์ชื่อยี่ห้อ (Brand)
df['Brand'] = df['Brand'].str.strip().str.title()
# 4. เขียนคำสั่ง จัดการข้อมูลสูญหาย (Missing Values)
```

```
# - ลบแถวข้อมูล หากค่าว่างเกิดในคอลัมน์เป้าหมาย (Target/Label)
df = df.dropna(subset=['Price'])
# - ใช้เทคนิคการแทนค่าข้อมูลที่เหมาะสมหากค่าว่างเกิดในคอลัมน์คุณลักษณะ (Feature)
import numpy as np
df['Brand'] = df.groupby('Model')['Brand'].transform(
    lambda x: x.fillna(
        x.mode()[0] if not x.mode().empty else np.nan
    )
)
# หากหาข้อมูลที่ถูกต้องมาแทนที่ไม่ได้ให้ลบทิ้ง
df = df.dropna(subset=['Brand'])
# 5. เขียนคำสั่ง จัดการข้อมูลผิดปกติ (Handling Outliers)
# - กรองข้อมูลรถที่มีปีผลิต (Year) ผิดปกติ
df = df[(df['Year'] >= 1900) & (df['Year'] <= 2026)]
# - กรองข้อมูลรถที่มีเลขไมล์ (Mileage) ผิดปกติ
df = df[df['Mileage'] >= 0]
# 6. เขียนคำสั่ง ลบข้อมูลที่ซ้ำซ้อน (Drop Duplicates)
df = df.drop_duplicates()

# ขั้นตอนที่ 2: การปรับสเกลข้อมูล (Feature Scaling)
# 1. นำเข้าไลบรารีที่จำเป็นสำหรับการสเกลข้อมูล
from sklearn.preprocessing import MinMaxScaler, StandardScaler
# 2. เขียนคำสั่ง เพื่อทำการปรับสเกลแบบ Normalization กับ Price
# - สร้างตัวแปรใหม่ Price_Norm เก็บคอลัมน์ Price ที่สเกลแล้ว
min_max_scaler = MinMaxScaler()
Price_Norm = min_max_scaler.fit_transform(df[['Price']])
df['Price'] = Price_Norm
# - เขียนคำสั่งตรวจสอบผลลัพธ์ว่าค่าต่ำสุดเป็น 0.0 และสูงสุดเป็น 1.0 หรือไม่
print(f"Price -> Min: {df['Price'].min()}, Max: {df['Price'].max()}")
# 3. เขียนคำสั่ง เพื่อทำการปรับสเกลแบบ Standardization กับ Mileage
# - สร้างตัวแปรใหม่ Mileage_Std เก็บคอลัมน์ Mileage ที่สเกลแล้ว
std_scaler = StandardScaler()
Mileage_Std = std_scaler.fit_transform(df[['Mileage']])
```

```
df['Mileage'] = Mileage_Std
# เขียนคำสั่งตรวจสอบผลลัพธ์ (ค่าเฉลี่ยของคอลัมน์นี้ควรเข้าใกล้ 0.0)
print(f"Mileage -> Mean: {df['Mileage'].mean():.4f}")
# ขั้นตอนที่ 3: วิศวกรรมคุณลักษณะ (Feature Engineering)
# 1. เขียนคำสั่ง การคัดเลือกคุณลักษณะ (Feature Selection)
# - ให้ทำการลบคอลัมน์ที่ไม่เกี่ยวข้องกับการทำนายทิ้ง ได้แก่ Car_ID, Owner_Name และ
Lucky_Number
cols_to_drop = ['Car_ID', 'Owner_Name', 'Lucky_Number']
df = df.drop(columns=cols_to_drop)
# 2. เขียนคำสั่ง การแปลงข้อมูลหมวดหมู่ (Encoding Categorical Data)
# - ให้ใช้เทคนิค Label Encoding หรือ .map() กับคอลัมน์ Fuel_Type (ชนิดเชื้อเพลิง) เพื่อแปลงค่า
Petrol, Diesel, Hybrid ให้เป็นตัวเลข
from sklearn.preprocessing import LabelEncoder
le_encoder = LabelEncoder()
df['Fuel_Type'] = le_encoder.fit_transform(df['Fuel_Type'])
# - ให้ใช้เทคนิค Label Encoding กับคอลัมน์ Brand (ยี่ห้อรถ) เพื่อแปลงยี่ห้อรถให้เป็นตัวเลข
df['Brand'] = le_encoder.fit_transform(df['Brand'])
# - ให้ใช้เทคนิค Label Encoding กับคอลัมน์ Model (โมเดลรถ) เพื่อแปลงยี่ห้อรถให้เป็นตัวเลข
df['Model'] = le_encoder.fit_transform(df['Model'])
# (ในใบงานนี้เราจะใช้ Label Encoding เพื่อความง่ายในการเขียนโค้ดและการดูผลลัพธ์เบื้องต้น แม้ว่าทาง
ทฤษฎี Brand และ Model จะเป็นข้อมูล Nominal ที่ควรใช้ One-Hot Encoding ก็ตาม)

# ขั้นตอนที่ 4: การแบ่งชุดข้อมูล (Train-Test Split)
# 1. เขียนคำสั่ง แยกข้อมูลออกเป็น 2 ส่วน:
# - ตัวแปร X (Features): เก็บข้อมูลคุณลักษณะทั้งหมด
feature_cols = ['Year', 'Mileage', 'Brand', 'Fuel_Type']
X = df[feature_cols]
# - ตัวแปร y (Target): เก็บข้อมูลคำตอบที่ต้องการทำนาย
y = df['Price']
# 2. เขียนคำสั่ง นำเข้าฟังก์ชัน train_test_split จากไลบรารี sklearn
from sklearn.model_selection import train_test_split
# 3. เขียนคำสั่ง แบ่งข้อมูล X และ y ออกเป็นชุดสอน (Train Set) และชุดทดสอบ (Test Set) โดยกำหนด
ขนาด Test Set = 20%
```

```python
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
# 4. เขียนคำสั่ง ตรวจสอบจำนวนของข้อมูลทั้ง 4 ตัวแปรที่ได้ (X_train, X_test, y_train, y_test)
print(f"X_train len: {len(X_train)}")
print(f"X_test len: {len(X_test)}")
print(f"y_train len: {len(y_train)}")
print(f"y_test len: {len(y_test)}")
```

แบบบันทึกผลการปฏิบัติงาน

ส่วนที่ 1: ผลการทำความสะอาดข้อมูล (Data Cleaning Results)

ตรงกับวัตถุประสงค์การเรียนรู้ข้อที่ 2: อธิบายผลลัพธ์และค่าสถิติที่เกิดจากการเปลี่ยนแปลงของข้อมูลในแต่ละขั้นตอนของการเตรียมข้อมูลได้

คำชี้แจง: ให้ดูผลลัพธ์จากหน้าจอ Terminal หลังรันคำสั่ง แล้วบันทึกค่าที่ได้ลงในช่องว่าง

1. จำนวนข้อมูล (Rows):
   - จำนวนแถวก่อนทำความสะอาด (Original Data): 100 แถว
   - จำนวนแถวหลังทำความสะอาด (Cleaned Data): 96 แถว
2. การจัดการข้อมูลผิดปกติ (Outliers):
   - ปีผลิต (Year): ค่าต่ำสุด (Min) ที่เหลืออยู่คือปี ค.ศ. 2015 (ต้องไม่ต่ำกว่า 1900)
   - เลขไมล์ (Mileage): ค่าต่ำสุด (Min) ที่เหลืออยู่คือ 10000 (ต้องไม่ติดลบ)
3. การแปลงข้อมูล (Formatting):
   - ชนิดข้อมูล (Dtype) ของคอลัมน์ Price ปัจจุบันคือ: float64

ส่วนที่ 2: ผลการปรับสเกลข้อมูล (Feature Scaling Results)

ตรงกับวัตถุประสงค์การเรียนรู้ข้อที่ 2: อธิบายผลลัพธ์และค่าสถิติที่เกิดจากการเปลี่ยนแปลงของข้อมูลในแต่ละขั้นตอนของการเตรียมข้อมูลได้

1. Normalization (Min-Max Scaling) บนคอลัมน์ Price:
   - ค่าต่ำสุด (Min) ของ Price: 0.0
   - ค่าสูงสุด (Max) ของ Price: 1.0
2. Standardization (Z-Score Scaling) บนคอลัมน์ Mileage:
   - ค่าเฉลี่ย (Mean) ของ Mileage: 0.0000

ส่วนที่ 3: ผลการทำวิศวกรรมคุณลักษณะ (Feature Engineering)

ตรงกับวัตถุประสงค์การเรียนรู้ข้อที่ 1 และ 2: (ข้อ 1-2 วัดการเลือกใช้คำสั่งเตรียมข้อมูล ส่วนข้อ 3 วัดความเข้าใจผลลัพธ์การแปลงข้อมูล

1. รายชื่อคอลัมน์ที่ไม่เกี่ยวข้องและถูกตัดออก (Drop) คือ?

Car_ID, Owner_Name, Lucky_Number

2. รายชื่อคอลัมน์ที่ เหลืออยู่ สำหรับนำไปใช้ทำนายผล คือ?

Year, Mileage, Brand, Fuel_Type

3. จงบันทึกค่าตัวเลขที่ได้จากการแปลง (Mapping/Label Encoding) ของข้อมูลตัวอย่าง 3 แถวแรก

รายการที่	ค่าเดิมของ Brand (เช่น Toyota)	ค่าตัวเลขใหม่ของ Brand (เช่น 0)	ค่าเดิมของ Fuel_Type (เช่น Petrol)	ค่าตัวเลขใหม่ของ Fuel_Type (เช่น 1)
1	Toyota	9	Petrol	3
2	Honda	1	Petrol	3
3	Ford	0	Diesel	1

ส่วนที่ 4: ผลการแบ่งชุดข้อมูล (Data Splitting Results)

ตรงกับวัตถุประสงค์การเรียนรู้ข้อที่ 1 และ 2

คำชี้แจง: บันทึกขนาด (Shape) ของตัวแปรที่ได้จากการแบ่ง Train/Test

1. สัดส่วนการแบ่งข้อมูล: 80 : 20 (Train : Test)
2. จำนวนข้อมูลในชุดสอน (Train Set):
   - X_train มีจำนวน 76 แถว
   - y_train มีจำนวน 76 แถว
3. จำนวนข้อมูลในชุดทดสอบ (Test Set):
   - X_test มีจำนวน 20 แถว
   - y_test มีจำนวน 20 แถว