รานงานผลการทดลอง

การทำนายราคาขายรถใช้แล้วในอินเดียโดยใช้ต้นไม้ตัดสินใจ

จัดทำโดย

นายณัชพล แก้วตัน รหัสนิสิต 6610450927

เสนอ

รศ.ดร.นวลวรรณ สุนทรภิษัช

รายงานนี้เป็นส่วนหนึ่งของรายวิชาหลักพื้นฐานของปัญญาประดิษฐ์ (01418261)

ภาคเรียนที่ 2 ปีการศึกษา 2567

ข้อมูล (Dataset)

ชุดข้อมูลที่ได้รับ : carprice (ทำนายราคาขายรถใช้แล้วในอินเดีย)

ข้อมูลประกอบด้วย 8 columns คือ ยี่ห้อ, ปีที่ออกขายครั้งแรก, ราคาขายมือสอง, กิโลเมตรที่ขับไปแล้ว, ประเภทเชื้อเพลิง, ประเภทผู้ขายรถ, ประเภทเกียร์, จำนวนเจ้าของรถในอดีต

การเตรียมข้อมูล

1. การอ่านไฟล์จากชุดข้อมูล

```
import pandas as pd

# ----- plot tools -----
import matplotlib.pyplot as plt
import seaborn as sns

# ----- sklean -----
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error
from sklearn.model_selection import GridSearchCV, cross_val_score

import warnings
warnings.filterwarnings('ignore')
```

```
data = pd.read_excel("CarPrice.xlsx")
data.head(10)
```

	ยี่ห้อ	ปีที่ออกขายครั้งแรก	ราคาขายมือสอง	กิโลเมตรที่ขับไปแล้ว	ประเภทเชื้อเพลิง	ประเภทผู้ขายรถ	ประเภทเกียร์	จำนวนเจ้าของรถในอดีต
0	Suzuki	2007	60000	70000.0	เบนซิน	เจ้าของขายเอง	Manual	3.0
1	Suzuki	2007	135000	50000.0	เบนซิน	เจ้าของขายเอง	Manual	1.0
2	Hyundai Verna 1.6 SX	2012	600000	100000.0	ดีเซล	เจ้าของขายเอง	Manual	1.0
3	Mitsubishi	2017	250000	46000.0	เบนซิน	เจ้าของขายเอง	Manual	1.0
4	Honda	2014	450000	141000.0	ดีเซล	เจ้าของขายเอง	Manual	2.0
5	Suzuki	2007	140000	125000.0	เบนซิน	เจ้าของขายเอง	Manual	1.0
6	Hyundai Xcent 1.2 Kappa S	2016	550000	25000.0	เบนซิน	เจ้าของขายเอง	Manual	1.0
7	Tata	2014	240000	60000.0	เบนซิน	เจ้าของขายเอง	Manual	2.0
8	Hyundai Creta 1.6 VTVT S	2015	850000	25000.0	เบนซิน	เจ้าของขายเอง	Manual	1.0
9	Suzuki	2017	365000	78000.0	CNG	เจ้าของขายเอง	Manual	1.0

2. Handling missing values

```
data.dropna(inplace=True)
data.shape[0]
```

```
477
```

- เนื่องจาก 8 แถวด้านล่างสุดของชุดข้อมูลเป็นข้อมูลว่างจึงสามารถตัดออกได้โดยไม่กระทบข้อมูล

3. Handling duplicate values

```
data.drop_duplicates(inplace = True)
data.shape[0]
```

```
487
```

- ตัดข้อมูลที่ซ้ำกันออกเพื่อลด overfitting

4. เปลี่ยนชื่อ columns

```
data = data.rename(columns={"ยี่ห้อ" : "Brand",
                            "ประเภทเชื้อเพลิง" : "Fuel_Type",
                            "ประเภทผู้ขายรถ" : "Seller_Type",
                            "ประเภทเกียร์" : "Gear_Type",
                            "กิโลเมตรที่ขับไปแล้ว" : "Mileage",
                            "ปีที่ออกขายครั้งแรก" : "First_Sell_Year",
                            "ราคาขายมือสอง" : "Secondhand_Price",
                            "จำนวนเจ้าของรถในอดีต" : "Past_Owner"
                           })
data
```

	Brand	First_Sell_Year	Secondhand_Price	Mileage	Fuel_Type	Seller_Type	Gear_Type	Past_Owner
0	Suzuki	2007	60000	70000.0	เบนซิน	เจ้าของขายเอง	Manual	3.0
1	Suzuki	2007	135000	50000.0	เบนซิน	เจ้าของขายเอง	Manual	1.0
2	Hyundai Verna 1.6 SX	2012	600000	100000.0	ดีเซล	เจ้าของขายเอง	Manual	1.0
3	Mitsubishi	2017	250000	46000.0	เบนซิน	เจ้าของขายเอง	Manual	1.0
4	Honda	2014	450000	141000.0	ดีเซล	เจ้าของขายเอง	Manual	2.0

- เพื่อให้ง่ายต่อการสร้างรูปกราฟต่างๆและการเรียกใช้งาน

Feature engineering

1. การจัดลดความซับซ้อน
   - เพิ่ม feature Model เนื่องจากว่า feature Brand นี้มีความหลากหลายถึง 122 ตัว จึงตัดสินใจในการสร้าง feature นี้เพื่อให้โมเดลเรียนรู้จากทั้งยี่ห้อรถและรุ่นของรถนั้นๆ

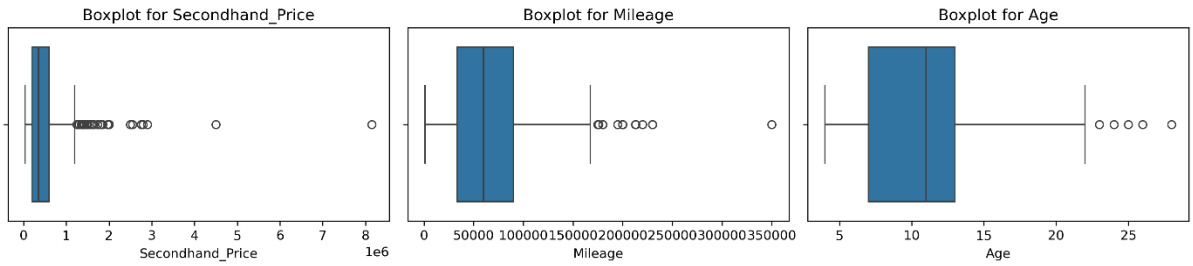
```
data["Model"] = data['Brand'].str.split().str[1]
data.head(5)
```

	Brand	First_Sell_Year	Secondhand_Price	Mileage	Fuel_Type	Seller_Type	Gear_Type	Past_Owner	Model
0	Suzuki	2007	60000	70000.0	เบนซิน	เจ้าของขายเอง	Manual	3.0	NaN
1	Suzuki	2007	135000	50000.0	เบนซิน	เจ้าของขายเอง	Manual	1.0	NaN
2	Hyundai Verna 1.6 SX	2012	600000	100000.0	ดีเซล	เจ้าของขายเอง	Manual	1.0	Verna
3	Mitsubishi	2017	250000	46000.0	เบนซิน	เจ้าของขายเอง	Manual	1.0	NaN
4	Honda	2014	450000	141000.0	ดีเซล	เจ้าของขายเอง	Manual	2.0	NaN

   - เพิ่ม feature Age เนื่องจากปีที่ออกขาย (First_Sell_Year) เป็นตัวเลขหลักพันและควรมองอายุของรถมากกว่า
   - ปรับจำนวนเจ้าของรถในอดีต (Past_Owner) ที่เป็นค่าทศนิยมเป็นจำนวนเต็ม

```
# add new feature : Age
data["Age"] = 2024 - data["First_Sell_Year"]

data.Past_Owner = data.Past_Owner.astype(int)
```

2. Handling Outliers
   - ข้อมูล จะถูกกรอง ซึ่งช่วยในการลบ outliers ที่มีค่าเกินไปออกไปจากข้อมูล



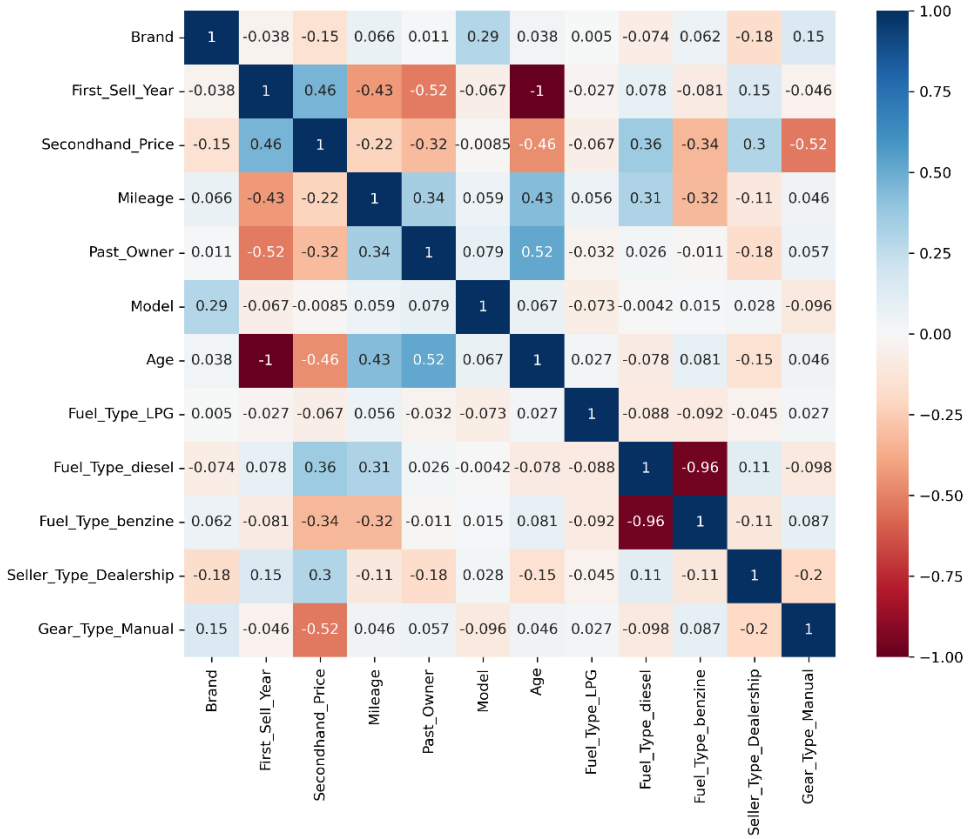

```
data = data[(data["Secondhand_Price"] < data["Secondhand_Price"].quantile(0.99)) &
            (data["Mileage"] < data["Mileage"].quantile(0.99))]
```

3. Encoding
   - ใช้ LabelEncoder กับ feature Brand Model
   - ใช้ get_dummies กับ feature Fuel_Type, Seller_Type , Gear_Type

```
le = LabelEncoder()
data["Brand"] = le.fit_transform(data["Brand"])
data["Model"] = le.fit_transform(data["Model"])

# Type feature (categorical columns)
data = pd.get_dummies(data, columns=['Fuel_Type', 'Seller_Type', 'Gear_Type'], drop_first=True)
columns_to_int = ['Fuel_Type_LPG', 'Fuel_Type_ดีเซล', 'Fuel_Type_เบนซิน',
                  'Gear_Type_Manual', 'Seller_Type_เต็นท์รถ']

data[columns_to_int] = data[columns_to_int].astype(int)

data = data.rename(columns={"Fuel_Type_ดีเซล" : "Fuel_Type_diesel",
                            "Fuel_Type_เบนซิน" : "Fuel_Type_benzine",
                            "Seller_Type_เต็นท์รถ" : "Seller_Type_Dealership",
                           })
data.head(5)
```

	Brand	First_Sell_Year	Secondhand_Price	Mileage	Past_Owner	Model	Age	Fuel_Type_LPG	Fuel_Type_diesel	Fuel_Type_benzine	Seller_Type_Dealership	Gear_Type_M
0	12	2007	60000	70000.0	3	25	17	0	0	1	0	
1	12	2007	135000	50000.0	1	25	17	0	0	1	0	
2	6	2012	600000	100000.0	1	18	12	0	1	0	0	
3	9	2017	250000	46000.0	1	25	7	0	0	1	0	
4	5	2014	450000	141000.0	2	25	10	0	1	0	0	

4. ความสัมพันธ์ระหว่าง feature



การประเมินโมเดล

โมเดลตัวที่ 00

Feature : Brand, Age, Mileage, Fuel_Type_LPG, Fuel_Type_diesel, Fuel_Type_benzine, Past_Owner, Model, Seller_Type_Dealership, Gear_Type_Manual

hyperparameter : random_state = 12

การแบ่งชุดข้อมูล : ชุดฝึก (Train) 80% และชุดทดสอบ (Test) 20%

ผลการทดสอบ :

Metric	Value
R^2	0.563
Mean Squared Error	61125178515.658
Mean Absolute Error	161484.369

การแบ่งชุดข้อมูล : ชุดฝึก (Train) 60% และชุดทดสอบ (Test) 40%

ผลการทดสอบ :

Metric	Value
R^2	0.581
Mean Squared Error	78634873654.518
Mean Absolute Error	178722.539

โมเดลตัวที่ 01

Feature : Brand, Mileage, Fuel_Type_LPG, Fuel_Type_diesel, Fuel_Type_benzine,

Gear_Type_Manual, Age, Seller_Type_Dealership

การแบ่งชุดข้อมูล : ชุดฝึก (Train) 60% และชุดทดสอบ (Test) 40%

Hyperparameter : random_state = 12

ผลการทดสอบ :

Metric	Value
R^2	0.642
Mean Squared Error	67178494335.141
Mean Absolute Error	176808.900

โมเดลตัวที่ 02

Feature : Brand, Mileage, Fuel_Type_LPG, Fuel_Type_diesel, Fuel_Type_benzine,

Gear_Type_Manual, Age, Seller_Type_Dealership

การแบ่งชุดข้อมูล : ชุดฝึก (Train) 60% และชุดทดสอบ (Test) 40%

Hyperparameter : max_depth=5, random_state=14

ผลการทดสอบ :

Metric	Value
R^2	0.731
Mean Squared Error	50461965967.113
Mean Absolute Error	150597.884

Hyperparameter Tuning

```
param_grid = {
    'max_depth': [3,4,5,6,7],
    'min_samples_split': [2, 3, 4 , 5, 6, 10],
}

grid_search = GridSearchCV(DecisionTreeRegressor(), param_grid, cv=3, scoring='r2')
grid_search.fit(X_train, y_train)

print("Best Parameters:", grid_search.best_params_)
bestmodel = grid_search.best_estimator_
```

ค่า Hyperparameter ที่ได้ : max_depth: 5, min_samples_split: 10

ผลการทดสอบ :

Metric	Value
R^2	0.568
Mean Squared Error	50461965967.113
Mean Absolute Error	150597.884

สรุปผลการทดลอง

โมเดล	Test size	R^2	MSE	MAE
ตัวที่ 00	20%	0.563	61125178515.658	161484.369
ตัวที่ 00	40%	0.581	78634873654.518	178722.539
ตัวที่ 01	40%	0.642	67178494335.141	176808.900
ตัวที่ 02	40%	0.731	50461965967.113	150597.884
Tuning	40%	0.568	50461965967.113	150597.884

จากการทดลองจะพบว่าในโมเดลตัวที่ 00 เมื่อเพิ่มขนาด Test size ค่า R^2 มากขึ้น เป็นเพราะโมเดลลด overfitting ลง และข้อมูลทดสอบมีความหลากหลายมากขึ้นครอบคลุมมากได้ดียิ่งขึ้น ซึ่งช่วยให้สามารถวัดประสิทธิภาพในการทำนายได้แม่นยำขึ้นเมื่อพบกับข้อมูลใหม่ที่ไม่เคยเห็นมาก่อน

โมเดลตัวที่ 01 และ 02 มีการลด feature พบว่าค่า R^2 มากขึ้น และ ค่า MSE, MAE ลดลง ซึ่งหมายความว่าการลด features ที่ไม่เกี่ยวข้อง เช่น Past_Owner และ Model ช่วยให้โมเดลมีความแม่นยำในการทำนายมากขึ้นทำให้โมเดลสามารถอธิบายความแปรผันของข้อมูลได้ดีขึ้นและลดข้อผิดพลาดในการทำนาย

โดยโมเดลตัวที่ 02 มีการ Pruning โดยพบว่าค่า max_depth = 5 ส่งผลให้โครงสร้างของโมเดลมีความเรียบง่ายขึ้น ช่วยลดความซับซ้อน (Complexity) และป้องกันปัญหา Overfitting ข้อมูลใน Training set ในขณะเดียวกันยังสามารถรักษาความแม่นยำใน Test set ได้ดี ส่งผลให้ค่า R^2 เพิ่มขึ้น และค่าความผิดพลาด MSE และ MAE ลดลงมากยิ่งขึ้นเมื่อเปรียบเทียบกับโมเดลตัวที่ 01

การ Hyperparameter Tuning พบว่าได้ค่า ค่า R^2 ลดลง แต่ค่า MSE, MAE เท่าเดิม เป็นเพราะโมเดลเกิด overfitting เนื่องจากค่า min_samples_split = 10 ดังนั้นค่าของ Hyperparameter จึงไม่เหมาะสมต่อ Test set นั่นเอง