รายงาน

เรื่อง ความสัมพันธ์และการทำนายของการใช้งานเชื้อเพลิงของรถยนต์

จัดทำโดย

นายกฤษณพงษ์ เพ็งบุญ 6330300038

นายจิรเมธ สุทธาวาณิชย์ 6330300119

เสนอ

ผศ.ดร. กุลวดี สมบูรณ์วิวัฒน์

รายงานนี้เป็นส่วนหนึ่งของรายวิชา

03603452 การทำเหมืองข้อมูล หมู่เรียนบรรยาย 800

ภาคปลาย ปีการศึกษา 2565

มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตศรีราชา

คำนำ

รายงานเล่มนี้เป็นส่วนหนึ่งของรายวิชา 03603452 การทำเหมืองข้อมูลเพื่อใช้กระบวนการทำเหมืองข้อมูลในการค้นคว้าเกี่ยวกับความสัมพันธ์และการทำนายของการใช้งานเชื้อเพลิงของรถยนต์

ทางผู้จัดทำหวังว่า รายงานเล่มนี้จะเป็นประโยชน์กับผู้อื่นที่สนใจในเรื่อง การทำเหมืองข้อมูล หากมีข้อผิดพลาดประการใด ทางผู้จัดทำก็ขออภัยมา ณ ที่นี้ด้วย

ผู้จัดทำ

นายกฤษณพงษ์ เพ็งบุญ 6330300038

นายจิรเมธ สุทธาวาณิชย์ 6330300119

สารบัญ

คำนำ ..... ก

สารบัญ ..... ข

ปัญหา ความสำคัญ และเป้าหมายของรายงาน 1

ทฤษฎีพื้นฐาน 1

โมเดลที่เลือกใช้ 1

Classification: decision tree 1

Association analysis: FP-growth 2

เครื่องมือที่ใช้ในการทำเหมืองข้อมูล 2

Pyspark 2

หลักการอื่น ๆ ที่เกี่ยวข้อง 2

k-fold Cross-Validation 2

F1-score 2

การสำรวจข้อมูลเบื้องต้น 3

- **Dataset ที่ใช้** 3

- ข้อมูลทั้งหมด 3

- เชื้อเพลิงประเภท Gasoline 4

- เชื้อเพลิงประเภท Diesel 4

- เชื้อเพลิงประเภท Electro 5

data mining models 5

model 1 decision tree 5

Model 2 FP-Growth 7

ผลการรันโปรแกรม และซอร์สโค้ด 8

ผลการรันโปรแกรม 8

Model 1 decision tree 8

Model 2 FP-Growth 10

ซอร์สโค้ด 12

สรุปผลและวิจารณ์ 12

ปัญหา ความสำคัญ และเป้าหมายของรายงาน

หาชนิดของเชื้อเพลิงที่ใช้งานของรถยนต์ และหาความสัมพันธ์ขององค์ประกอบต่าง ๆ ของรถยนต์ในการใช้พลังงานชนิดนั้นเพื่อนำไปใช้ประโยชน์ต่าง ๆ เช่น หาเชื้อเพลิงที่ใช้ของรถที่ไม่มีข้อมูล และหาความสัมพันธ์ขององค์ประกอบของรถที่ใช้เชื้อเพลิงชนิดต่าง ๆ เพื่อนำไปวิเคราะและพัฒนาต่อ ๆ ไป

ทฤษฎีพื้นฐาน

โมเดลที่เลือกใช้

Classification: decision tree

Classification คือ เป็นจำแนกประเภทข้อมูล ของ Machine Learning แบบ Supervised Learningโดยมีตัวอย่างในชุดข้อมูลสอนคือ training set จะมีคุณลักษณะหนึ่งซึ่งบอกคำประเภทของตัวอย่างนั้น เราเรียกกันว่า Class Label

Supervised Learning คือการสอนให้คอมพิวเตอร์สามารถหาคำตอบได้ด้วยตัวเอง หลังจากเรียนรู้และฝึกหัดจากชุดข้อมูลตัวอย่างไปแล้วระยะหนึ่ง เปรียบเทียบก็คือการที่เรามีชุดข้อมูลที่มีคำตอบที่ถูกต้องแล้ว ป้อนให้กับโมเดลเพื่อให้มันเรียนรู้ระยะหนึ่ง แล้วค่อยทดสอบด้วยข้อมูลชุดอื่นเพื่อตรวจสอบว่าโมเดลมีความแม่นยำในการทำนายมากน้อยแค่ไหน

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

Association analysis: FP-growth

Association Rules เป็นกระบวนการหนึ่งในการทำ Data Mining ที่ได้รับความนิยมมาก ในการหาความสัมพันธ์ของข้อมูลสองชุดหรือมากกว่าสองชุดขึ้นไปภายในกลุ่มข้อมูลที่มีขนาดใหญ่

FP-Growth มีแนวคิดหลักในการใช้โครงสร้าง FP-tree (Frequent-Pattern tree) เพื่อย่อขนาดฐานข้อมูล เก็บเฉพาะข้อมูลที่จำเป็นต่อการค้นหา frequent item sets หลีกเลี่ยงการสแกนฐานข้อมูลซึ่งเป็นโอเปอเรชันที่ใช้เวลานานค้นหา frequent item sets จาก FP-tree ด้วยวิธีการ FP-growth divide-and-conquer ใช้วิธีค้นหา frequent item sets บนFP-treeที่มีขนาดเล็กลงเรื่อยๆ และหลีกเลี่ยงการสร้าง candidate item sets

เครื่องมือที่ใช้ในการทำเหมืองข้อมูล

Pyspark

Pyspark เป็นเครื่องมือหนึ่งที่เกิดจากการรวมตัวกันระหว่าง Apache Spark กับ Python ซึ่งทำให้เราสามารถเขียน Python ใน Spark ได้

Apache Spark เป็นเครื่องมือหนึ่งที่ถูกดีไซน์มาให้เราใช้งานแบบทำงานกลุ่มได้ โดยที่เชื่อมต่อระบบการทำงานของคอมพิวเตอร์เข้าด้วยกัน หรือเรียกว่า Cluster computing platform ซึ่งสามารถกระจายงานที่ต้องทำไปยังเครื่องอื่นๆภายในระบบ ทำให้เราสามารถประมวลผลข้อมูลขนาดใหญ่แบบเต็มประสิทธิภาพ หรือแบบ real-time ไปพร้อมๆกันได้

หลักการอื่น ๆ ที่เกี่ยวข้อง

k-fold Cross-Validation

k-fold Cross Validation คือเครื่องมือที่ช่วยให้เราตัดสินใจได้ว่าเราควรแบ่งข้อมูลส่วนไหนไปเป็น Training Data เป็นวิธีการประเมินประสิทธิภาพของโมเดล มีเป้าหมายเพื่อการใช้ประโยชน์จากข้อมูลที่มีอยู่อย่างมีประสิทธิผลมากที่สุด

F1-score

precision และ recall

- Precision : ค่าความแม่นยำ เกิดจากการนำ ค่า tp มาเทียบกับ fp
- Recall : ค่าความถูกต้อง เกิดจากการนำค่า tp มาเทียบกับ fn

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

tp คือ จำนวนครั้งที่ทำนายเป็นคลาสเป้าหมายแล้วทำนายคลาสนั้นได้ถูกต้อง

fp คือ จำนวนครั้งที่ทำนายไม่เป็นคลาสเป้าหมาย แต่คลาสจริงคือคลาสเป้าหมาย

fn คือ จำนวนครั้งที่ทำนายไม่เป็นคลาสเป้าหมายแล้วทำนายคลาสนั้นได้ถูกต้อง

F1-score คือ ค่าเฉลี่ยฮาร์มอนิกของ precision และ recall เป็นค่าที่ได้จากการเอาค่า precision และ recall มาคำนวณรวมกัน ดังนี้

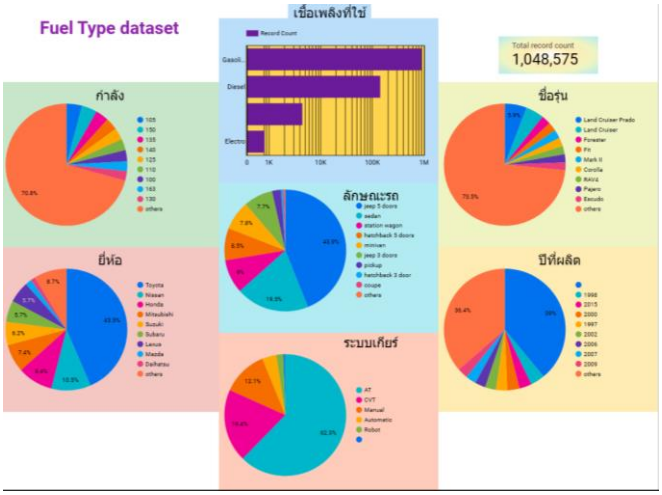
$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

การสำรวจข้อมูลเบื้องต้น

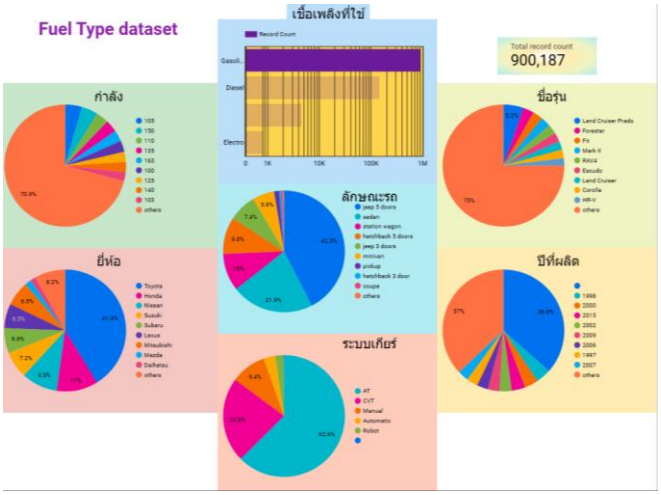
- **Dataset ที่ใช้**

CarFuelType | Kaggle

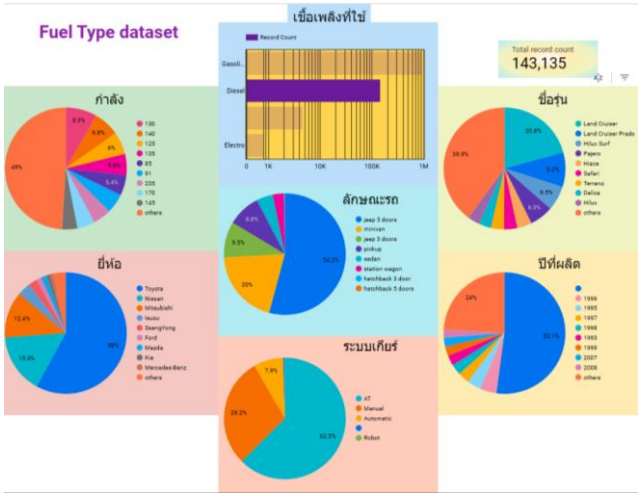
- ข้อมูลทั้งหมด



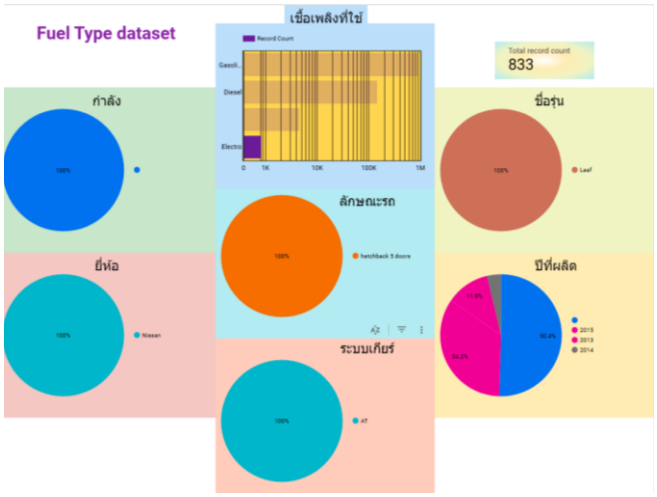
- เชื้อเพลิงประเภท Gasoline



- เชื้อเพลิงประเภท Diesel



- เชื้อเพลิงประเภท Electro



- https://lookerstudio.google.com/reporting/c3d3f4d5-f664-4683-b7c2-05f0c2c3671e

data mining models

model 1 decision tree

1. import spark library ที่ต้องใช้

```
import pyspark
from pyspark.sql import SparkSession
from pyspark.ml.classification import DecisionTreeClassifier
from pyspark.sql import functions
from pyspark.ml.feature import StringIndexer
from pyspark.ml.feature import VectorAssembler
from pyspark.ml import Pipeline
from pyspark.sql.functions import col
from pyspark.ml.tuning import ParamGridBuilder, CrossValidator
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
from pyspark.mllib.evaluation import MulticlassMetrics
```

2. สร้าง spark session

```
spark = SparkSession.builder \
    .master('local[*]') \
    .config("spark.driver.memory", "15g") \
    .config("spark.logConf", "true") \
    .appName('my-app') \
    .getOrCreate()
```

3. โหลดไฟล์ และ ทำการจัดการข้อมูลที่เป็นค่า null

```
df = spark.read.csv('fueltype.csv',header=True)
df = df.dropna(subset='fuelType')
df = df.fillna('0')


df.show(5)
df.printSchema()
df = df.fillna('0')
```

4. ทำการแปลงข้อมูลแต่ละแอตทริบิวต์ให้เป็นตัวเลข

```
indexers = [StringIndexer(inputCol=column, outputCol=column+"_n").fit(df) for column in list(
    set(df.columns)-set(['_c0']))]#

pipeline = Pipeline(stages=indexers)
df_r = pipeline.fit(df).transform(df)

df_r = df_r.select(*(col(c).cast("int").alias(c) for c in df_r.columns))
df_fea = df_r.drop('_c0','brand','name','bodyType','year','transmission','power','fuelType')

df_fea.show(5)
df_fea.printSchema()
```

5. ทำการแปลงข้อมูลให้เหลือเพียง features และ target โดยการรวม features ทั้งหมดมาเก็บในรูป vector

```
assembler = VectorAssembler(inputCols=['brand_n', 'name_n','bodyType_n','year_n',
                                       'transmission_n','power_n'], outputCol='features')
output = assembler.transform(df_fea)

final_data = output.select('features', 'fuelType_n')

final_data.show(5)

train_data,test_data = final_data.randomSplit([0.7,0.3])
```

6. สร้าง model, กำหนดความลึกของ model (7,10,13,15,20), กำหนดตัววัดประสิทธิภาพเป็น F1-score และใช้การเลิก model จากการใช้วิธี 5-fold Cross-Validation

```
model = DecisionTreeClassifier(labelCol='fuelType_n',featuresCol='features', maxDepth=10)

# Create ParamGrid for Cross Validation
dtparamGrid = (ParamGridBuilder()
             .addGrid(model.maxDepth, [7, 10, 13, 15, 20])
             .addGrid(model.maxBins, [20, 40, 60, 80, 100])
             .build())

# Evaluate model
dtevaluator = MulticlassClassificationEvaluator(labelCol= 'fuelType_n',metricName='f1')#

# Create 5-fold CrossValidator
dtcv = CrossValidator(estimator = model,
                      estimatorParamMaps = dtparamGrid,
                      evaluator = dtevaluator,
                      numFolds = 5)
```

7. ทำการ train model จาก train_data และ ทำมาทดสอบและแสดงผลประสิทธิภาพของ model

```
dtc_model = dtcv.fit(train_data)

dtc_preds = dtc_model.transform(test_data)

pred_acc = dtevaluator.evaluate(dtc_preds)

train_acc = dtevaluator.evaluate(dtc_model.transform(train_data))

pred_table = dtc_preds.select("features", "prediction",'probability', "fuelType_n")
pred_table.show(10)
pred_table.sample(False, 0.1, seed=0).limit(10).show()
print('')
print('=========================================')
print('')
print('')
print('F1-score train : ',train_acc )
print('')
print('F1-score test : ',pred_acc )
print('')
print('')
print('=========================================')
print('')
```

Model 2 FP-Growth

1. import spark library ที่ต้องใช้

```
import pyspark
from pyspark.sql import SparkSession
from pyspark.sql import functions
from pyspark.sql.functions import col
from pyspark.ml.fpm import FPGrowth
from pyspark.sql.functions import *
```

2. สร้าง spark session

```
spark = SparkSession.builder \
    .master('local[*]') \
    .config("spark.driver.memory", "15g") \
    .config("spark.logConf", "true") \
    .appName('my-app') \
    .getOrCreate()
```

3.โหลดไฟล์, ทำการจัดการข้อมูลที่เป็นค่า null และแปลงข้อมูลให้เป็น transaction

```
df = spark.read.csv('fueltype.csv',header=True)
df = df.dropna(subset='fuelType')
df_g = df.select('*').where(df.fuelType == "Gasoline")
df_g.show(5)

df_d = df.select('*').where(df.fuelType == "Diesel")
df_d.show(5)

df_e = df.select('*').where(df.fuelType == "Electro")
df_e.show(5)

df_g_merged = df_g.select(array('brand','name','bodyType','year',
                                'transmission','power','fuelType').alias('item'))
df_g_merged = df_g_merged.withColumn('items', functions.expr('filter(item, x -> x is not null)'))

df_d_merged = df_d.select(array('brand','name','bodyType','year',
                                'transmission','power','fuelType').alias('item'))
df_d_merged = df_d_merged.withColumn('items', functions.expr('filter(item, x -> x is not null)'))

df_e_merged = df_e.select(array('brand','name','bodyType','year',
                                'transmission','power','fuelType').alias('item'))
df_e_merged = df_e_merged.withColumn('items', functions.expr('filter(item, x -> x is not null)'))


df_g_merged.show(5)
df_d_merged.show(5)
df_e_merged.show(5)
```

4.สร้าง model, กำหนดค่า support และ confident แล้วนำข้อมูล transction ไปประมวลผล

```
fp = FPGrowth(itemsCol="items", minSupport=0.2, minConfidence=0.3)
fpmodel1 = fp.fit(df_g_merged)
fpmodel2 = fp.fit(df_d_merged)
fpmodel3 = fp.fit(df_e_merged)
```

5. แสดงผลลัพธ์

```
a1 = fpmodel1.associationRules.filter(array_contains(col("consequent"), "Gasoline"))
a1.select("antecedent","consequent","support","confidence").orderBy(
    col("support").desc(),col("confidence").desc()).show()
a2 = fpmodel2.associationRules.filter(array_contains(col("consequent"), "Diesel"))
a2.select("antecedent","consequent","support","confidence").orderBy(
    col("support").desc(),col("confidence").desc()).show()
a3 = fpmodel3.associationRules.filter(array_contains(col("consequent"), "Electro"))
a3.select("antecedent","consequent","support","confidence").orderBy(
    col("support").desc(),col("confidence").desc()).show()
```

ผลการรันโปรแกรม และซอร์สโค้ด

ผลการรันโปรแกรม

Model 1 decision tree

- ข้อมูลดิบ

```
+---+------+------------------+-----------------+----+------------+-----+--------+
| id| brand|              name|         bodyType|year|transmission|power|fuelType|
+---+------+------------------+-----------------+----+------------+-----+--------+
|  0|Toyota|Land Cruiser Prado|     jeep 5 doors|1995|          AT|  130|  Diesel|
|  1|Toyota|      Land Cruiser|     jeep 5 doors|   0|   Automatic|  286|  Diesel|
|  2|Toyota|              Vitz|hatchback 5 doors|2019|         CVT|   95|Gasoline|
|  3|Toyota|           Mark II|            sedan|2002|          AT|  160|Gasoline|
|  4|Toyota|              RAV4|     jeep 5 doors|2010|          AT|  170|Gasoline|
+---+------+------------------+-----------------+----+------------+-----+--------+
only showing top 5 rows

root
 |-- id: string (nullable = false)
 |-- brand: string (nullable = false)
 |-- name: string (nullable = false)
 |-- bodyType: string (nullable = false)
 |-- year: string (nullable = false)
 |-- transmission: string (nullable = false)
 |-- power: string (nullable = false)
 |-- fuelType: string (nullable = false)
```

- ข้อมูลหลังแปลงเป็นเลข

```
[Stage 8830:>                                                   (0 + 0) / 1]+---+
+----------+------+------+-------+-------+----------+
| id|transmission_n|  id_n|fuelType_n|year_n|name_n|power_n|brand_n|bodyType_n|
+---+--------------+------+----------+------+------+-------+-------+----------+
|  0|             0|     0|         1|    24|     0|      8|      0|         0|
|  1|             3|     1|         1|     0|     1|    106|      0|         0|
|  2|             1|158861|         0|    28|    18|     34|      0|         3|
|  3|             0|269363|         0|     5|     4|     10|      0|         1|
|  4|             0|379878|         0|    10|     6|      9|      0|         0|
+---+--------------+------+----------+------+------+-------+-------+----------+
only showing top 5 rows

root
 |-- id: integer (nullable = true)
 |-- transmission_n: integer (nullable = true)
 |-- id_n: integer (nullable = true)
 |-- fuelType_n: integer (nullable = true)
 |-- year_n: integer (nullable = true)
 |-- name_n: integer (nullable = true)
 |-- power_n: integer (nullable = true)
 |-- brand_n: integer (nullable = true)
 |-- bodyType_n: integer (nullable = true)
```

- ข้อมูลหลังแปลงให้เป็น features และ target

```
+--------------------+----------+
|            features|fuelType_n|
+--------------------+----------+
|(6,[3,5],[24.0,8.0])|         1|
|[0.0,1.0,0.0,0.0,...|         1|
|[0.0,18.0,3.0,28....|         0|
|[0.0,4.0,1.0,5.0,...|         0|
|[0.0,6.0,0.0,10.0...|         0|
+--------------------+----------+
```

- ผลลัพธ์การทำนาย (เนื่องจากตอน spilt เป็น train และ test set จะมีการ sort ข้อมูลอยู่ด้วยจึงทำให้ผลลัพธ์ที่ออกมามี features เหมือนกัน)

```
|            features|prediction|  probability|fuelType_n|
+-------------------+---------+-------------+---------+
|(6,[1,2],[5.0,1.0])|      0.0|[1.0,0.0,0.0]|        0|
|(6,[1,2],[5.0,1.0])|      0.0|[1.0,0.0,0.0]|        0|
|(6,[1,2],[5.0,1.0])|      0.0|[1.0,0.0,0.0]|        0|
|(6,[1,2],[5.0,1.0])|      0.0|[1.0,0.0,0.0]|        0|
|(6,[1,2],[5.0,1.0])|      0.0|[1.0,0.0,0.0]|        0|
|(6,[1,2],[5.0,1.0])|      0.0|[1.0,0.0,0.0]|        0|
|(6,[1,2],[5.0,1.0])|      0.0|[1.0,0.0,0.0]|        0|
|(6,[1,2],[5.0,1.0])|      0.0|[1.0,0.0,0.0]|        0|
|(6,[1,2],[5.0,1.0])|      0.0|[1.0,0.0,0.0]|        0|
|(6,[1,2],[5.0,1.0])|      0.0|[1.0,0.0,0.0]|        0|
+-------------------+---------+-------------+---------+
only showing top 10 rows

+--------------------+----------+-------------+----------+
|            features|prediction|  probability|fuelType_n|
+--------------------+----------+-------------+----------+
| (6,[1,2],[5.0,1.0])|       0.0|[1.0,0.0,0.0]|         0|
| (6,[1,2],[5.0,1.0])|       0.0|[1.0,0.0,0.0]|         0|
| (6,[1,2],[5.0,1.0])|       0.0|[1.0,0.0,0.0]|         0|
| (6,[1,2],[5.0,1.0])|       0.0|[1.0,0.0,0.0]|         0|
| (6,[1,2],[5.0,1.0])|       0.0|[1.0,0.0,0.0]|         0|
|(6,[1,2],[24.0,1.0])|       0.0|[1.0,0.0,0.0]|         0|
|(6,[1,2],[24.0,1.0])|       0.0|[1.0,0.0,0.0]|         0|
|(6,[1,2],[24.0,1.0])|       0.0|[1.0,0.0,0.0]|         0|
|(6,[1,2],[76.0,1.0])|       0.0|[1.0,0.0,0.0]|         0|
|(6,[1,2],[81.0,2.0])|       0.0|[1.0,0.0,0.0]|         0|
+--------------------+----------+-------------+----------+
```

- ความแม่นยำของ model ที่ได้

```
==========================================


F1-score train :  0.9986538668995799

F1-score test :  0.9987718846598841


==========================================
```

Model 2 FP-Growth

- ข้อมูลดิบ

```
+---+------+------------+-----------------+----+------------+-----+--------+
| id| brand|        name|         bodyType|year|transmission|power|fuelType|
+---+------+------------+-----------------+----+------------+-----+--------+
|  2|Toyota|        Vitz|hatchback 5 doors|2019|         CVT|   95|Gasoline|
|  3|Toyota|     Mark II|            sedan|2002|          AT|  160|Gasoline|
|  4|Toyota|        RAV4|     jeep 5 doors|2010|          AT|  170|Gasoline|
|  5|Toyota|      Cresta|            sedan|null|          AT|  180|Gasoline|
|  7|Toyota|Corolla Axio|            sedan|null|         CVT|  105|Gasoline|
+---+------+------------+-----------------+----+------------+-----+--------+
only showing top 5 rows
```

```
+---+------+-----------------+------------+----+------------+-----+--------+
| id| brand|             name|    bodyType|year|transmission|power|fuelType|
+---+------+-----------------+------------+----+------------+-----+--------+
|  0|Toyota|Land Cruiser Prado|jeep 5 doors|1995|          AT|  130|  Diesel|
|  1|Toyota|      Land Cruiser|jeep 5 doors|null|   Automatic|  286|  Diesel|
|  6|Toyota|      Estima Emina|     minivan|null|          AT|  100|  Diesel|
| 25|Nissan|        Pathfinder|jeep 5 doors|2009|          AT|  190|  Diesel|
| 35|Toyota|      Land Cruiser|jeep 5 doors|1992|          AT|  165|  Diesel|
+---+------+-----------------+------------+----+------------+-----+--------+
only showing top 5 rows

+----+------+----+-----------------+----+------------+-----+--------+
|  id| brand|name|         bodyType|year|transmission|power|fuelType|
+----+------+----+-----------------+----+------------+-----+--------+
|1399|Nissan|Leaf|hatchback 5 doors|2013|          AT| null| Electro|
|3401|Nissan|Leaf|hatchback 5 doors|2013|          AT| null| Electro|
|5402|Nissan|Leaf|hatchback 5 doors|2013|          AT| null| Electro|
|7402|Nissan|Leaf|hatchback 5 doors|2013|          AT| null| Electro|
|9402|Nissan|Leaf|hatchback 5 doors|2013|          AT| null| Electro|
+----+------+----+-----------------+----+------------+-----+--------+
only showing top 5 rows
```

- ข้อมูลที่แปลงเป็น transaction (แถว item เป็นแถวที่มีค่า null อยู่ ส่วน แถว items จะไม่มีค่า null)

```
+--------------------+--------------------+ +--------------------+--------------------+
|                item|               items| |                item|               items|
+--------------------+--------------------+ +--------------------+--------------------+
|[Toyota, Vitz, ha...|[Toyota, Vitz, ha...| |[Toyota, Land Cru...|[Toyota, Land Cru...|
|[Toyota, Mark II,...|[Toyota, Mark II,...| |[Toyota, Land Cru...|[Toyota, Land Cru...|
|[Toyota, RAV4, je...|[Toyota, RAV4, je...| |[Toyota, Estima E...|[Toyota, Estima E...|
|[Toyota, Cresta, ...|[Toyota, Cresta, ...| |[Nissan, Pathfind...|[Nissan, Pathfind...|
|[Toyota, Corolla ...|[Toyota, Corolla ...| |[Toyota, Land Cru...|[Toyota, Land Cru...|
+--------------------+--------------------+ +--------------------+--------------------+
only showing top 5 rows                     only showing top 5 rows
```

```
+--------------------+--------------------+
|                item|               items|
+--------------------+--------------------+
|[Nissan, Leaf, ha...|[Nissan, Leaf, ha...|
|[Nissan, Leaf, ha...|[Nissan, Leaf, ha...|
|[Nissan, Leaf, ha...|[Nissan, Leaf, ha...|
|[Nissan, Leaf, ha...|[Nissan, Leaf, ha...|
|[Nissan, Leaf, ha...|[Nissan, Leaf, ha...|
+--------------------+--------------------+
only showing top 5 rows
```

- ผลลัพธ์

```
+----------+----------+-------------------+----------+
|antecedent|consequent|            support|confidence|
+----------+----------+-------------------+----------+
|      [AT]|[Gasoline]| 0.6256888846428575|       1.0|
|[jeep 5 ...|[Gasoline]|  0.424729528420206|       1.0|
|[Toyota]|[Gasoline]| 0.4126942513055621|       1.0|
|[Toyota, AT]|[Gasoline]|  0.285425139443249|       1.0|
|[jeep 5 ...|[Gasoline]| 0.2832433705441203|       1.0|
|     [CVT]|[Gasoline]|0.22601081775231147|       1.0|
|  [sedan]|[Gasoline]|0.21868345132733533|       1.0|
+----------+----------+-------------------+----------+
```

```
+----------+----------+-------------------+----------+
|antecedent|consequent|            support|confidence|
+----------+----------+-------------------+----------+
|      [AT]|  [Diesel]| 0.6248227198099696|       1.0|
|[Toyota]|  [Diesel]| 0.5795298145107766|       1.0|
|[jeep 5 ...|  [Diesel]| 0.5415586683899815|       1.0|
|[jeep 5 ...|  [Diesel]|0.39103643413560624|       1.0|
|[Toyota, AT]|  [Diesel]|0.38493031054598803|       1.0|
|[jeep 5 ...|  [Diesel]| 0.3408320816012855|       1.0|
|  [Manual]|  [Diesel]|0.29167569078143013|       1.0|
|[jeep 5 ...|  [Diesel]| 0.2634994934851713|       1.0|
|[Land Cruiser]|  [Diesel]|0.20765710692702694|       1.0|
|[Land Cruiser, To...|  [Diesel]|0.20765710692702694|       1.0|
|[Land Cruiser, je...|  [Diesel]|0.20159290180598735|       1.0|
|[Land Cruiser, je...|  [Diesel]|0.20159290180598735|       1.0|
+----------+----------+-------------------+----------+
```

```
+--------------------+----------+-------------------+----------+
|          antecedent|consequent|            support|confidence|
+--------------------+----------+-------------------+----------+
|           [Nissan]| [Electro]|                1.0|       1.0|
|[AT, Nissan, hatc...| [Electro]|                1.0|       1.0|
|          [AT, Leaf]| [Electro]|                1.0|       1.0|
| [hatchback 5 doors]| [Electro]|                1.0|       1.0|
|[AT, Leaf, Nissan...| [Electro]|                1.0|       1.0|
|  [AT, Leaf, Nissan]| [Electro]|                1.0|       1.0|
|[Leaf, hatchback ...| [Electro]|                1.0|       1.0|
|        [AT, Nissan]| [Electro]|                1.0|       1.0|
|[AT, Leaf, hatchb...| [Electro]|                1.0|       1.0|
|[Leaf, Nissan, ha...| [Electro]|                1.0|       1.0|
|[Nissan, hatchbac...| [Electro]|                1.0|       1.0|
|              [Leaf]| [Electro]|                1.0|       1.0|
|                [AT]| [Electro]|                1.0|       1.0|
|[AT, hatchback 5 ...| [Electro]|                1.0|       1.0|
|      [Leaf, Nissan]| [Electro]|                1.0|       1.0|
|  [2015, AT, Nissan]| [Electro]|0.34213685474189676|       1.0|
|[2015, AT, Leaf, ...| [Electro]|0.34213685474189676|       1.0|
|      [2015, Nissan]| [Electro]|0.34213685474189676|       1.0|
|[2015, AT, Nissan...| [Electro]|0.34213685474189676|       1.0|
|[2015, Nissan, ha...| [Electro]|0.34213685474189676|       1.0|
+--------------------+----------+-------------------+----------+
only showing top 20 rows
```

ซอร์สโค้ด

- Kit6330300038/datamining (github.com)

สรุปผลและวิจารณ์

จากโมเดล decision tree มีประสิทธิภาพถึง 99.87 % และจากโมเดล FP-Growth พบว่ารถที่ใช้เชื้อเพลิงประเภท Gasoline และ Diesel ส่วนใหญ่จะมีระบบเกียร์แบบ AT ,รถที่ใช้เชื้อเพลิงประเภท Electro ทั้งหมดจะมียี่ห้อ Nissan รุ่น Leaf ระบบเกียร์แบบ AT ลักษณะรถแบบ hatchback 5 doors โดยข้อมูลที่นำมาใช้ยังขาดความหลากหลายโดยเฉพาะรถประเภทที่ใช้เชื้อเพลิงประเภท Electro แทบจะเป็นข้อมูลเดียวกันในทุก record และบางแอตทริบิวต์มีผลลัพธ์ส่วนใหญ่หรือทั้งหมดเป็นเชื้อเพลิงประเภทเดียวเช่น รุ่นของรถ