สุรพล กุลเสวตร์ เลขทะเบียน 5709680085

กระบวนการ CRISP-DM ในการวิเคราะห์ข้อมูลเพื่อแก้ไขปัญหาในการพยากรณ์สถานะของบ่อน้ำ

1. ให้น.ศ.เข้าใจปัญหา (Business Understanding)
    a. Scenario
        i. ผู้คนในทวีปแอฟริกาขาดแคลดน้ำดื่ม รัฐบาลแก้ไขปัญหาด้วยการขุดเจาะบ่อน้ำเพื่อจ่ายน้ำ ดังนี้ผู้ใช้ต้องการระบบที่สามารถช่วยดูแลรักษาบ่อน้ำได้ในอยู่ในสภาพใช้งานได้
    b. Business objectives
        i. บ่อน้ำมีเวลาในการใช้งานเพื่อขึ้น
        ii. พยากรณ์ถึงสถานะของบ่อจากการใช้จ่ายน้ำในบ่อนั้น ๆ
    c. Business success criteria
        i. สามารถพยากรณ์ได้
2. ให้น.ศ.เข้าใจข้อมูล (Data understanding)
    a. ข้อมูลทั้งหมดใน train_data1.csv มีเป็นข้อมูลที่เกี่ยวข้องกับบ่อน้ำ มีทั้งหมด 5938 แถว มีหลักทั้งหมด 41 หลัก ประกอบไปด้วย
        i. ข้อมูล หมายเลขบ่อน้ำ ปริมาณน้ำที่มีในบ่อน้ำ วันที่บันทึกข้อมูล พิกัด ที่ตั้ง ฯลฯ ในไฟล์ data_description.pdf ดังภาพที่ 1

id	amnt_wat	date	sponsor	altitude	installer	longitude	latitude	name	no_privat	basin	village	region	region_id	district_id	lga	ward	demand	meeting	recorder
69572	6000	14/3/2011	Roman	1390	Roman	34.93809	-9.85632	none	0	Lake Nyas	Mnyusi B	Iringa	11	5	Ludewa	Mundindi	109	TRUE	GeoData C
41119	500	21/3/2011	Caritas	336	DWE	36.41593	-8.65746	Tulileli Tw	0	Rufiji	Minazini	Morogoro	5	4	Ulanga	Itete	186	TRUE	GeoData C
35572	0	25/3/2013	Governme	1624	DWE	37.51533	-3.26034	Kwa John	0	Pangani	Kotete	Kilimanjar	3	4	Moshi Rur	Marangu M	60	FALSE	GeoData C

ภาพที่ 1 ตัวอย่างข้อมูลของบ่อน้ำ

โดย target ของเราคือ status แถวสุดท้ายของข้อมูลนี้ที่สามารถบอกความสถานะของบ่อน้ำ

b. Data Preparation

i. ในการดูข้อมูล train_data1 มี attribute มีค่าซ้ำกันและไม่เกี่ยวข้องกับการพยากรณ์ของบ่อน้ำจึงทำการลบออกโดยใช้ภาษา R

```
data <- subset(data, select = -id)
data <- subset(data, select = -sponsor)
data <- subset(data, select = -no_private)
data <- subset(data, select = -date)
data <- subset(data, select = -recorder)
data <- subset(data, select = -e_type)
data <- subset(data, select = -e_type_grp)
data <- subset(data, select = -fee_type)
data <- subset(data, select = -source)
data <- subset(data, select = -quantity_grp)
data <- subset(data, select = -type_grp)
data <- subset(data, select = -permit)
data <- subset(data, select = -meeting)
```

ภาพที่ 2 ค่าซ้ำกันและไม่เกี่ยวข้องกับการพยากรณ์

Id ไม่จำเป็นต่อการพยากรณ์

Sponsor ไม่จำเป็นต่อการพยากรณ์

No_private ค่าที่ไม่ทราบ

Date มีค่าที่ผิดผลาด และ ไม่จำเป็นต่อการพยากรณ์

Recorder เป็นชื่อซ้ำกัน

E_type และ e_type_grp เป็นคำย่อ e_type_class

Fee_type เป็นคำย่องของ fee

Source เป็นค่าซ้ำของ source_type

quantity_grp เป็นค่าซ้ำของ w_quantity

type_grp เป็นค่าซ้ำของ type

permit ค่าซ้ำกัน

meeting ค่าซ้ำกัน

ii. ในไฟล์ train_data1 มีข้อมูลที่มีค่า 0 ค่าunknown ค่าที่เป็นพื้นที่ว่างจึงต้องทำการลบออกโดยใช้ภาษา R ดังโค็ดตัวอย่าง

```
data1 <- data

data1[data1=='unknown'] <- NA
data1[data1==''] <- NA
data1[data1==0] <- NA
data1 <- na.omit(data1)
```

ภาพที่ 3 ข้อมูลที่เป็น missing value จึงต้องทำการลบ

1. ทำการย้ายข้อมูลไปที่ data1 ข้อมูลในdata1 ทั้งหมดที่มี 0 , unknow ,ค่าว่าง แทนด้วย NA จากนั้นนำการลบข้อมูลที่มี NA ทั้งหมด
2. ข้อมูลที่เหลือทั้งหมดจะเหลือ 1088 แถว มีหลักทั้งหมด 28 หลัก

iii. ทำการแยกข้อมูล data1 เป็น trainset กับ test โดยแบ่งอัตราส่วน 70:30

1.
```
index <- 1:nrow(data1)
set.seed(333)
dt <- sample(index,trunc(0.3*nrow(data1)))
trainset <- na.omit(data1[-dt,])
test <-  na.omit(data1[dt,])
```

ภาพที่ 4 แบ่งอัตราส่วน 70:30

a. ตัวแปร trainset มีค่าทั้งหมด 762
b. ตัวแปร test มีค่าทั้งหมด 326

3. สร้างแบบจำลอง (Modeling)

    i. เนื่องจากเรามี target ชัดเจนซึ่งได้แก่ functional , broken , non-functional จึงควรใช้การจำลองในการพยากรณ์แบบ Classification โดยใช้ Support Vector Machine โดยใช้ caret

        1. ทำการจำลองโดยใช้ cross-validation 10 ชั้น เป็นตัวช่วย test performance
        2. ใช้ Support Vector Machine แบบ Radial Kernel

```
TrainingP <- trainControl(method = "cv" , number = 10)
set.seed(3233)
svm_model1 <- train(status ~ . , data = trainset , method = "svmRadial" , trControl = TrainingP)
```

ภาพที่ 5 จำลองข้อมูล

จากการทดสอบโมเดลได้ผลลัพท์ดังนี้

```
Support Vector Machines with Radial Basis Function Kernel

762 samples
 27 predictor
  3 classes: 'broken', 'functional', 'non-functional'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 686, 685, 687, 685, 687, 686, ...
Resampling results across tuning parameters:

  C     Accuracy   Kappa
  0.25  0.6876992  0.000000000
  0.50  0.6876992  0.000000000
  1.00  0.6863313  0.005730449

Tuning parameter 'sigma' was held constant at a value of 4.490228e-06
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were sigma = 4.490228e-06 and C = 0.25.
```

ภาพที่ 6 ผลจากการสร้าง model

3. การทดสอบ model กับ test data ที่เหลือ 30

```
test_pred <- predict(svm_model1, newdata = test)

confusionMatrix(test_pred, test$status )
```

ภาพที่ 7 คำสั่ง predict

จากการทำงานคำสั่ง predict และ สร้าง confusion matrix

```
Confusion Matrix and Statistics

                Reference
Prediction       broken functional non-functional
  broken              0          0              0
  functional         20        227             79
  non-functional      0          0              0

Overall Statistics

               Accuracy : 0.6963
                 95% CI : (0.6432, 0.7458)
    No Information Rate : 0.6963
    P-Value [Acc > NIR] : 0.5271

                  Kappa : 0
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: broken Class: functional Class: non-functional
Sensitivity                0.00000            1.0000                0.0000
Specificity                1.00000            0.0000                1.0000
Pos Pred Value                 NaN            0.6963                   NaN
Neg Pred Value             0.93865               NaN                0.7577
Prevalence                 0.06135            0.6963                0.2423
Detection Rate             0.00000            0.6963                0.0000
Detection Prevalence       0.00000            1.0000                0.0000
Balanced Accuracy          0.50000            0.5000                0.5000
```

ภาพที่ 8 ผลลัพธ์ confusion matrix

จากการทดสอบโมเดลกับtest set ได้ค่า accuracy อยู่ที่ 69% ซึ่งถือว่าค่าน้อย

4.เพิ่มข้อมูล train_data2 โดยใช้คำสั่ง

```
data <- read.csv("train_data1.csv")
data2 <- read.csv("train_data2.csv")

data3 <- rbind(data,data2)
```

ภาพที่ 9 การเพิ่มข้อมูล

หลังจากการเพิ่มข้อมูลและทำการ Data Preparation เหลือข้อมูลทั้งหมด 2190 row และหลังจากทำการแบ่งข้อมูล 70:30 เป็น train และ test

Train set มีจำนวนทั้งหมด 1533 row และ test set มีทั้งหมด 657 row

ทำการโมเดลได้ผลลัพท์ดังนี้

```
Support Vector Machines with Radial Basis Function Kernel

1533 samples
  27 predictor
   3 classes: 'broken', 'functional', 'non-functional'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 1379, 1380, 1379, 1380, 1380, 1380, ...
Resampling results across tuning parameters:

  C     Accuracy   Kappa
  0.25  0.7018929  0.000000000
  0.50  0.7018929  0.003179829
  1.00  0.7018886  0.009806749

Tuning parameter 'sigma' was held constant at a value of 2.961754e-06
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were sigma = 2.961754e-06 and C = 0.25.
```

ภาพที่ 10 โมเดล

ทำการทดสอบกับ test ได้ผลลัพท์ดังนี้

```
Confusion Matrix and Statistics

                Reference
Prediction       broken functional non-functional
  broken              0          0              0
  functional         60        463            134
  non-functional      0          0              0

Overall Statistics

               Accuracy : 0.7047
                 95% CI : (0.6682, 0.7394)
    No Information Rate : 0.7047
    P-Value [Acc > NIR] : 0.5194

                  Kappa : 0
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: broken Class: functional Class: non-functional
Sensitivity                0.00000            1.0000                 0.000
Specificity                1.00000            0.0000                 1.000
Pos Pred Value                 NaN            0.7047                   NaN
Neg Pred Value             0.90868               NaN                 0.796
Prevalence                 0.09132            0.7047                 0.204
Detection Rate             0.00000            0.7047                 0.000
Detection Prevalence       0.00000            1.0000                 0.000
Balanced Accuracy          0.50000            0.5000                 0.500
```

ภาพที่ 11 ผลลัพธ์ confusion matrix

เวลาในการพยากรณ์จะนานขึ้นเนื่องจากข้อมูลที่เพิ่มขึ้น จากการเติม data set เข้าไป ทำให้ค่า accuracy เพิ่มขึ้นเป็น 70 % ถือว่าค่าน้อยอยู่

5. จากการจำลอง ในข้อ 2- 4

   a. ข้อมูลทั้งหมด 5940 แถวมี มี 41 หลัก ข้อมูลส่วนใหญ่เป็น 0 ช่องว่าง เว้นวรรก โดยใช้จากคำสั่ง summary

   b. ทำการตัดข้อมูลที่ไม่จำเป็นทั้งหมดทั้งหมดเหลือ 2190 แถว 28 หลัก โดยใช้คำสั่ง subset

   c. ทำการจำลองแบบที่สร้างขึ้นมีลักษณะเป็น Support Vector Machine แบบ Radial Kernel

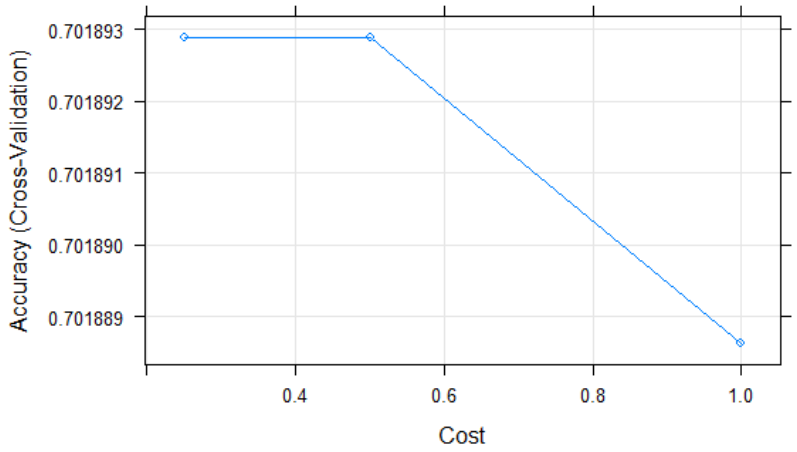
   d. มีโครงสร้าง

```
TrainingP <- trainControl(method = "cv" , number = 10)
set.seed(3233)
svm_model <- train(status ~ . , data = trainset , method = "svmRadial" , trControl = TrainingP)
```

TrainingP เป็นตัวแปรที่เก็บค่า cross-validation มีพารามิเตอร์ method คือ ฟังก์ชันนี้จะคำนวณข้อผิดพลาดในการคาด พารามิเตอร์ number คือ รอบของ cross-validation

Svm_model เป็นตัวแปรที่เก็บค่าโมเดล มีพารามิเตอร์ status ¬ . คือค่า target และค่าจาก attribute ที่ต้องการในที่นี้ใช่ . เพื่อเอาทุก attribute พารามิเตอร์ method คือ โมเดลที่ต้องการ พารามิเตอร์ trControl คือตัวแปรควบคุม cross-validation

   e.



ภาพที่ 12 cross-validation

   i. จากกราฟด้านบนแสดงให้เห็นว่า C ในช่วง 0.25 – 0.5 มีค่า accuracy สูงสุดคือ 70.1%

f. ในการนำไปใช้จริงโมเดลนี้ยังไม่สามารถทำได้ดี อยู่ในระดับพอใช้ได้ เนื่องจากค่า accuracy น้อย อยู่ที่ 70.1%

6.สรุป

จากข้อมูลผู้คนในทวีปแอฟริกาขาดแคลดน้ำดื่ม รัฐบาลแก้ไขปัญหาด้วยการขุดเจาะบ่อน้ำเพื่อจ่ายน้ำ ระบบที่สามารถช่วยดูแลรักษาบ่อน้ำได้ในอยู่ในสภาพใช้งานได้ ในการพยากรณ์ข้อมูลโดยใช้โมเดล Support Vector Machine แบบ Radial Kernel จากข้อมูลทั้งหมด 5940 รวม train_data1 และ train_data2 และได้ทำการ Data Preparation จนเหลือ 2190 ได้ผลความแม่นยำ 70 % จากผลความแม่นยำผมคิดว่าข้อมูลน้อยไปทำให้ความแม่นยำน้อยลง