Statistical Data Science
Kasetsart University (Bangkhen)

# Credit Card Fraud Detection

## Anonymized credit card transactions labeled as fraudulent or genuine

—

By Thanyarat Munkong 6214401206

# บทนำ

ที่มาของการทำการวิเคราะห์ธุรกรรมบัตรเครดิตนี้มาจากผู้เขียนมีความสงสัยเกี่ยวกับ **Fraud Detection** มาระยะหนึ่งจึงสนใจที่จะศึกษาหาความรู้ข้อมูลเพิ่มเติมจึงได้พบชุดข้อมูลที่น่าสนใจและเป็นข้อมูลขนาดใหญ่ได้มาจาก Machine Learning Group - ULB ซึ่งชุดข้อมูลนี้เป็นข้อมูลสาธรณะเปิดให้เข้าถึงได้จากใน https://www.kaggle.com/ ซึ่งเป็นชุดของมูลการทำธุรกรรมทางบัตรเครดิตที่มีทั้งเป็นการทำธุรกรรมแบบปกติและแบบไม่ปกติ (เกิดการฉ้อโกง) ซึ่งเมื่อนำชุดข้อมูลนี้มาวิเคราะห์ใน Machine Learning Algorithms แล้วจะสามารถสร้างโมเดลจากข้อมูลเชิงคุณภาพนี้มาตรวจสอบเฝ้าระวังพฤติกรรมในการใช้บัตรเครดิตที่ผิดปกติได้อย่างมีประสิทธิภาพ การนำโมเดลที่สร้างขึ้นมาใช้งานกับการตรวจสอบจริงครั้งนี้จะเป็นเครื่องมือในการคอยเฝ้าระวังให้กับผู้เป็นเจ้าของบัตรเครดิตไม่ให้เกิดการแอบอ้างการใช้งานบัตรในทางที่ผิดปกติได้ ชุดข้อมูลที่ได้มานี้มีทั้งหมด 31 ตัวแปร ซึ่งจะมีผลลัพธ์เป็นตัวแปร Class มีอยู่ด้วยกัน 2 Class คือ 0 และ 1 สำหรับ 0 คือ การทำธุรกรรมบัตรเครดิตที่เป็นปกติ 1 คือ การทำธุรกรรมบัตรเครดิตแบบไม่ปกติ(เกิดการฉ้อโกง) ส่วนอีก 30 ตัวแปรที่เหลือคือ Time Amount และ V1 - V28 เนื่องมาจากเหตุผลความเป็นส่วนตัว ตัวแปร V1 - V28 นี้ไม่สามารถบอกได้ว่าเป็นตัวแปรที่เกี่ยวข้องกับสิ่งใด แต่เป็นตัวแปรที่ผ่านกระบวณการ PCA Transformation มาแล้ว ข้อมูลที่ได้มาถูกปรับขนาดให้อยู่ในช่วงหนึ่งมาเรียบร้อยแล้ว ยกเว้นแต่ตัวแปร Time และ Amount ที่ยังไม่ถูกปรับ ข้อมูลชุดนี้ไม่พบข้อมูลที่ขาดหายไป ชุดข้อมูลนี้เป็นชุดข้อมูลขนาดใหญ่ที่มีจำนวนแถวของข้อมูลกว่าสามแสนแถว และเป็นข้อมูลที่มีปัญหาคาสไม่สมดุล

## เครื่องมือและวิธีการทางวิทยาการคอมพิวเตอร์

**เครื่องมือ**
Programing language: R-3.6.3
Text editor: Rstudio
Computer: ASUS ROG STRIX G531GU I7-9750H CPU@2.60GHz 64-bit GPU GeForce GTX1660Ti

**การเก็บรวมรวมข้อมูล** เก็บรวบรววมข้อมูลมาจาก https://www.kaggle.com/mlg-ulb/creditcardfraud

**การสร้างเว็บไซต์** โดยใช้ package (shiny) เป็นส่วนหลัก และมีการตกแต่งหน้า User Interface โดยใช้ package (shinydashboard) ตกแต่ง carousel (ภาพส่วนหัว) โดยใช้ package (htmltools) และ (bsplus)
ตัวอย่าง
```
bs_carousel(id = "the_beatles", use_indicators = TRUE, use_controls=FALSE) %>%
bs_append(content = bs_carousel_image(src = "FruadDetection.jpg") )
```
bs_carousel(id = “id”, use_indicators = TRUE, use_control = FALSE) จากภาพหน้าปก
id คือ id ที่ระบุภาพส่วนหัว
use_indicators คือ จุดด้านล่างภาพ TRUE คือมีจุด FALSE คือ ไม่มี
use_control คือ ลูกศรเลื่อนซ้ายขวา TRUE คือมีลูกศร FALSE คือ ไม่มี
bs_append(content = bs_carousel_image(src = "image.jpg") )
content คือ เนื้อหาที่ต้องการจะใส่ไปใน carousel ว่าเป็นภาพหรือตัวหนังสือ ในที่นี้เป็นภาพ
src คือ ที่อยู่ที่ไปถึงภาพๆนั้น ในที่นี้ให้นำภาพไปไว้ในfolder www ตำแหน่งเดียวกับไฟล์ app.R แล้วจะสามารถเขียนชื่อภาพลงไปได้เลย *ข้อควรระวัง ถ้าไม่มี folder www จะไม่สามารถแสดงภาพได้เลย*
เงื่อนไขตรรกศาสตร์ที่ใช้ในการแสดงผล element ใช้ package (shinyjs) ยกตัวอย่าง
```
  observe({
        if (input$go == 0) { hide("underSamplingColumn") }
  })
```
โค้ดในที่นี้จะเป็นการเช็คว่าถ้า input ที่เราตั้งชื่อ id ว่า go มีค่าเท่ากับ 0 จะเลือกให้ซ่อน element id ที่ชื่อว่า underSamplingColumn

**การนำเข้าข้อมูล** ใช้ package (data.table) ในการอ่านข้อมูลในโปรแกรม R studio ตัวอย่าง
```
df <- fread('creditcard.csv', header=T)
```
fread(‘filename’, header) เป็นฟังก์ชั่นที่ใช้ในการอ่านไฟล์ csv
filename คือ ชื่อไฟล์ที่ต้องการอ่าน
header คือ แถวแรกของไฟล์คือหัวตารางหรือไม่ T คือ ใช่ F คือ ไม่ใช่

**การแก้ไขปัญหาคาสไม่สมดุล** ใช้การทำ Undersampling โดยใช้ package(unbalanced) ตัวอย่าง
```
df.ubUnder<-ubUnder(X=df[, -31], Y=df$Class, perc = 50,  method = "percPos")
```
ubUnder(X, Y, perc, method) เป็นฟังก์ชั่นที่ใช้ในการทำUndersampling โดยการลบข้อมูลจาก majority

คาสแบบสุ่มและเก็บรักษาข้อมูลของคาส minority ไว้เพื่อทำให้ข้อมูลของสองคาสมีปริมาณเท่ากัน
X คือ ตัวแปรที่เราจะนำมาทำให้คาสมีขนาดเท่ากัน ไม่รวมคาส
Y คือ ตัวแปรคาส
perc คือ เปอร์เซ้นต์ของการSampling
method คือ วิธีการดำเนินการUndersampling perPos คือ เปอร์เซ็นของคาสบวกที่ต้องการ perUnder คือ เปอร์เซ้นของคาสลบที่ต้องการ

**การแบ่งข้อมูล** ข้อมูลฝึก : ข้อมูลทดสอบ ขนาด 70 : 30 (เริ่มต้น) โดยใช้ package (caTools) ตัวอย่าง
train.id <- caTools::sample.split(newData$Class, SplitRatio = 0.70)
newData.train <- subset(newData, train.id)
newData.validate <- subset(newData, !train.id)
โค้ดบรรทัดบนเป็นการเลือกแบ่งข้อมูลเป็น 70% บรรทัดถัดลงมาเป็นการแบ่งข้อมูลเป็นสับเซตโดยข้อมูลฝึกเป็น 70% และข้อมูลทดสอบเป็น 30%

**อัลกอริทึมที่ใช้ในการสร้างโมเดล** ใช้อัลกอรึทึมในการจำแนกคาสสองอัลกอริทึม คือ Logistic Regression และ K-nearest Neighbors Classification ซึ่งเป็น Supervised Learning
**Logistic Regression**
Fitting Generalized Linear Models (glm)
logistic.model = glm(formula = Class ~., family = binomial, data = newData.train)
Formula คือ สมการของโมเดล
family คือ การกระจายความผิดพลาดและลิงค์ฟังก์ชั่น ในที่นี้เป็นแบบ binomial และ ลิงค์ฟังก์ชั่นคือ logit
data คือ ชุดที่ข้อมูลฝึกที่เราจะนำไปสร้างเป็นโมเดล
Step AIC เป็นวิธีการคัดเลือกตัวแปรแบบ Akaike's Information Criteria เป้าหมายเพื่อใช้หาตัวแปรที่ให้ค่าพยากรณ์แม่นยำที่สุดโดยใช้การประมาณค่าความคลาดเคลื่อนในการพิจารณาโดยใช้ package (MASS) ในที่นี้โมเดลจะถูกสร้างมาจาก Step AIC อยู่ก่อนแล้วในR-shinyจะHardcodeไปเพื่อความรวดเร็วในการประมวลผล
step <- stepAIC(object = logistic.model, direction="both", trace=T)
object คือ โมเดลตั้งต้นที่เราจะทำการคัดเลือกตัวแปรในโมเดลนี้
direction คือ การค้นหาแบบขั้นตอน both คือ ค้นหาทั้งไปข้างหน้าและย้อนกลับ
trace คือ ถ้าเป็นบวกข้อมูลจะถูกแสดงผลในระหว่างการทำ Step AIC หรือไม่ ซึ่ง T คือใช่ F คือไม่ใช่
**K-nearest Neighbors**
เป็นอัลกอริทึมที่ใช้ในการจำแนกเพื่อนบ้านที่ใกล้ที่สุดของชุดข้อมูลทดสอบโดยได้โมเดลมาจากชุดข้อมูลฝึก ซึ่งจะใช้ package (class) ในการจำแนก
predicted <- knn(train = newData.train[, 1:30], test = newData.validate[, 1:30], cl = newData.train.class, k = k)
train คือ เซตของข้อมูลฝึกที่ไม่รวมตัวแปรคาสโดยชุดข้อมูลนี้ต้องเป็นชนิดเมทริกหรือดาต้าเฟรม
test คือ ชุดข้อมูลทดสอบที่ไม่รวมตัวแปรคาสโดยชุดข้อมูลนี้ต้องเป็นชนิดเมทริกหรือดาต้าเฟรม
cl คือ ตัวแปรที่ใช้จำแนก(คาส) เพียงอย่างเดียว
k คือ จำนวนเพื่อนบ้านที่ใกล้ที่สุดที่จะนำมาพิจารณา k ตัว

**การแปรผล**

**การพลอตกราฟ ใช้** package (ggplot2) **และ** (plotly) **ในการแสดงผล ยกตัวอย่าง**

```
pl <- ggplot(data=results)+geom_line(aes(x=k,y=Accuracy),size=1,color="red")+
geom_line(aes(x=k,y=Sensitivity),size=1,color="blue")+
geom_line(aes(x=k,y=Specificity),size=1,color="green")+
ylab('performance')+ theme_bw()
```

data **คือ ข้อมูลที่เราจะนำมาพลอต**

**จากเส้นกราฟน้ำเงินแสดงค่า** Sensitivity

x **คือ ค่า** k **ที่เราจะนำมาพลอต ในที่นี้** k =1 - 20

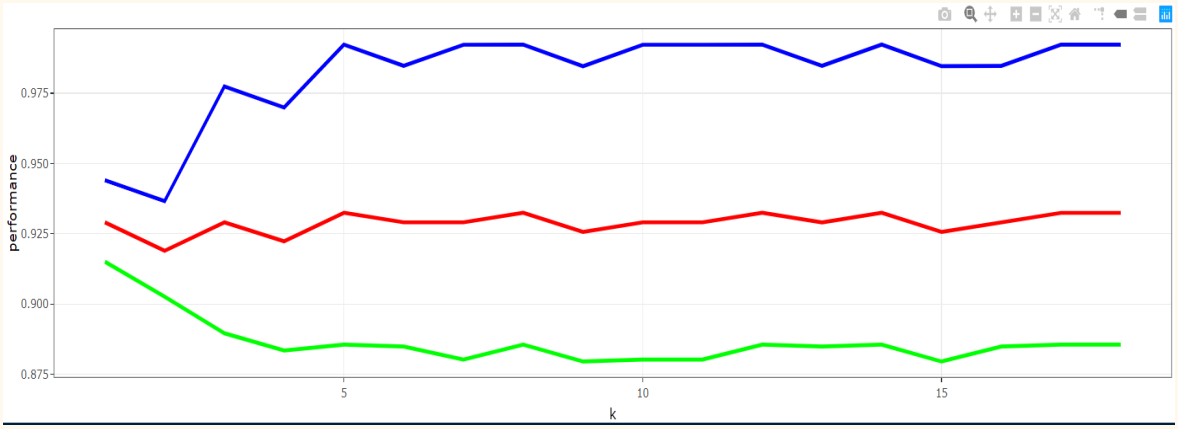
y **คือ ค่า** Accuracy **ที่เราจะนำมาพลอต**

size **คือ ขนาดเส้น**

color **คือ สีของเส้น**

ylab **คือ ชื่อแกน** Y

theme_bw() **คือ ธีม**




**ซึ่งจากกราฟนี้ผู้ใช้อาจแปรผลได้ว่า ที่** k = 3 **จะทำให้ได้ค่า** Accuracy **เท่ากับ** 0.92 Sensitivity **มีค่า** 0.97 **และ** Specificity **มีค่า** 0.89 **ซึ่งผู้ใช้อาจนำโมเดลที่ใช้ค่า** k = 3 **ไปตรวจสอบการใช้งานบัตรเครดิตที่ผิดประเภทได้**

**อื่นๆ** package (dplyr) **ใช้สำหรับการสเกลค่า เช่น**

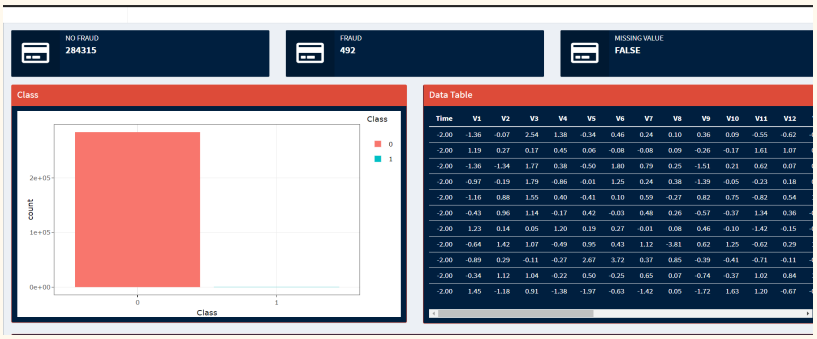
```
df$Amount<- df$Amount  %>% scale() %>% as.data.frame()
```

**เป็นการใช้เครื่องหมายไปป์** %>% **ในการสเกลตัวแปร** Amount **ให้มีค่าอยู่ในช่วงหนึ่งๆ โดยหลังจากสเกลแล้วจะแปลง**Amount**นั้นให้เป็นชนิด** dataframe

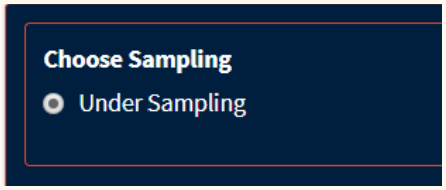
## วิธีการใช้งานเว็บไซต์

1. ไปที่ https://rosenx.shinyapps.io/creditCardFraudDetection/ หรือสแกน QR CODE

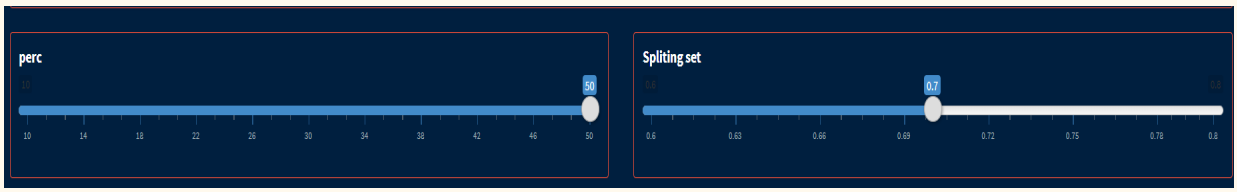
2. เช็คคาสว่าสมดุลไหมระหว่าง No Fraud คือการใช้งานบัตรเครดิตแบบปกติ และ Fraud คือการใช้งานบัตรเครดิตไม่ปกติ(เกิดการฉ้อโกง) สำรวจว่ามีข้อมูลขาดหายไหม ซึ่งจะโชว์อยู่ในกล่องสีน้ำเงินด้านขวา และสำรวจข้อมูลตารางฝั่งขวา ส่วนฝั่งซ้ายเป็นกราฟของคาสที่ไม่สมดุลกัน



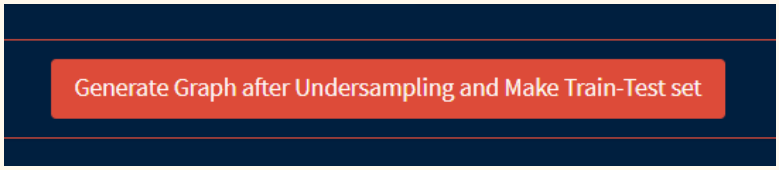

3. จัดการกับปัญหาคาส imbalance โดยการเลือกวิธีการ Sampling ข้อมูลในที่นี้จะเลือก Undersampling ให้อยู่แล้ว ไม่แนะนำการทำ Oversampling หรือ SMOTE เหตุผลเนื่องมาจาก Server ที่ใช้มีทรัพยากรไม่เพียงพอและการทำทั้งสองวิธีนี้จะใช้เวลาคำนวณนานและอาจเกิดปัญหา overfitting ได้




4. เลือกเปอร์เซ็นต์ของการ Sampling ในที่นี้จะเลือก Sampling ข้อมูลคาส No Fraud ว่าจะเลือกมากี่เปอร์เซ็นต์ ในที่นี้กำหนดมา 50% และเลือกเปอร์เซ็นต์ของการSplit ข้อมูลฝึก ในที่นี้เป็น 70%

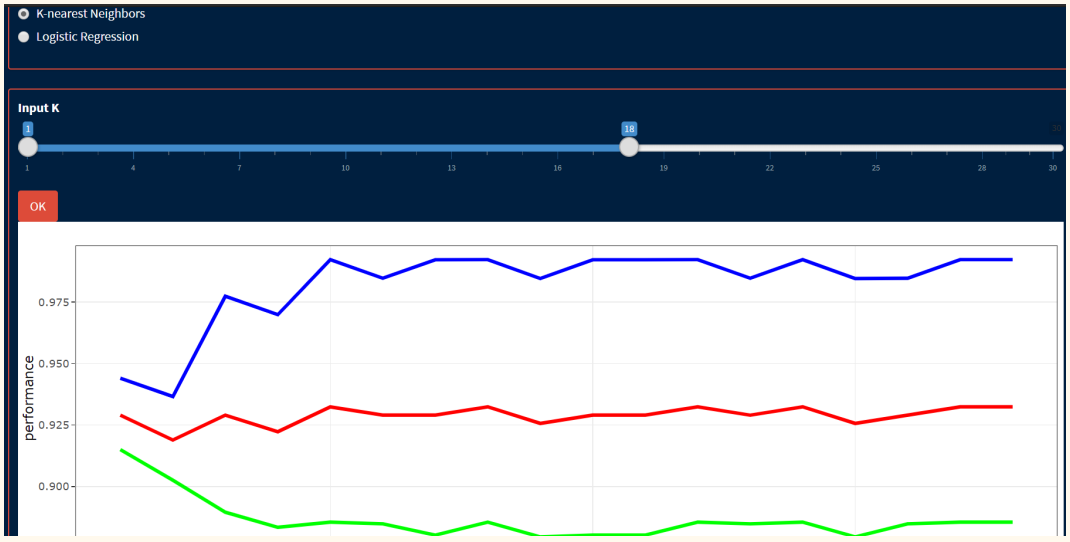


5. กดปุ่มเพื่อดูกราฟว่าคาสที่ทำUndersampling มาเท่ากันหรือยังพร้อมกับการแบ่งข้อมูลฝึกกับข้อมูลทดสอบในปุ่มเดียว

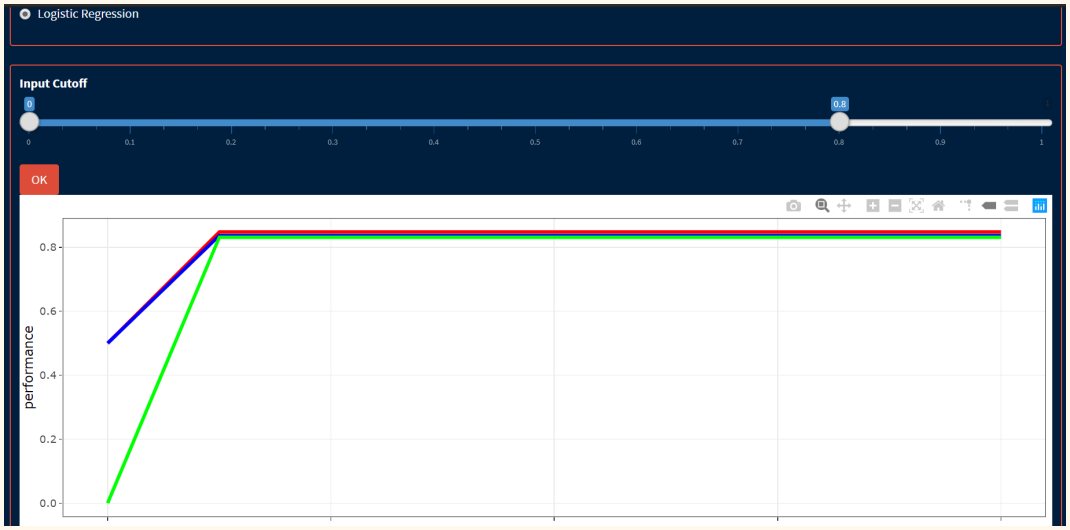



6. หลังจากนั้นให้เลือก Algorithm ที่จะใช้ทดสอบซึ่งมีสองอันคือ K-nearest Neighbors และ Logistic Regression

**สำหรับ K-nearest Neighbor**s ให้เลือกช่วงค่า k ที่ต้องการดูผลลัพธ์ในแบบของกราฟที่แสดง Accuracy Specitivity และ Sensitivity




**สำหรับ Logistic Regression** ให้เลือกจุด cutoff แทนจะได้กราฟแสดงผลค่า Accuracy Specitivity และ Sensitivity ดังภาพ



## ผู้สนับสนุน

1. **ชุดข้อมูล** creditcard.csv **จาก** https://www.kaggle.com/mlg-ulb/creditcardfraud
2. **ผู้สนับสนุนการสร้างเว็บไซต์ จาก** https://www.shinyapps.io/

## ขอขอบคุณ

1. **ความรู้ในวิชา** Statistical Data Science **จาก ดร. ธรรมกร แซ่ตั้ง**
2. **ความรู้ในวิชา** Machine Learning **จาก รศ. นวลวรรณ สุนทรภิษัช**