Decision Trees && K-Nearest Neighbors

Decision Trees

Dicision trees คือ เป็น Machine learing อย่างหนึ่งที่จะสามารถอธิบายได้ว่า ทำไมต้องใช้คลาสนี้ และจะอธิบายออกมาในรูปของ Tree หรือ แบบรากต้นไม้ จะมี Node ใหญ่จะเป็นตัวตั้งคำถามว่า ใช่ หรือ ไม่ใช่ Node ตัวถัดไปจะถามต่อไปว่าใช่หรือไม่จนครบ ยิ่งมีจำนวนชั้นของ Tree มากขึ้นเท่าไหร่จะยิ่งมีความแม่นยำมากขึ้นเท่านั้น

- **Parameters:** max_depth(int defalt=none)

  เป็นการระบุความลึกของ tree สามารถใช้ป้องการเกิด over-fit

K-Nearest Neighbors (KNN)

ขั้นตอนวิธีการเพื่อนบ้านใกล้ที่สุด (K-Nearest Neighbour Algorithm) เป็นวิธีที่ใช้ในการจัดแบ่งคลาส โดยเทคนิคนี้จะตัดสินใจว่า คลาสใดที่จะแทนเงื่อนไขหรือกรณีใหม่ๆ ได้บ้าง โดยการตรวจสอบจำนวนบางจำนวน ในขั้นตอนวิธีการเพื่อนบ้านใกล้ที่สุด ของกรณีหรือเงื่อนไขที่เหมือนกันหรือใกล้เคียงกันมากที่สุด โดยจะหาผลรวม (Count Up) ของจำนวนเงื่อนไข หรือกรณีต่างๆ สำหรับแต่ละคลาส และกำหนดเงื่อนไขใหม่ๆ ให้คลาสที่เหมือนกันกับคลาสที่ใกล้เคียงกันมากที่สุด

- **Parameters:** n_neighbors(int defalt=5)

  จำนวนสมาชิกใกล้เคียงที่นำมาเปรียบเทียบ

การนำมาใช้ใน scikit

*ใช้ dataset ของลักษณะต่าง ๆ จำนวน 30 ชนิดเพื่อนำมาจำแนกระดับบองมะเร็งเต้านมโดย มี 2 ระดับ คือ รุนแรงและไม่รุนแรง dataset นี้มาจาก scikit-learn

- Import dataset

```
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_breast_cancer
from sklearn import tree,neighbors
breast_cancer = load_breast_cancer()
```

- แบ่ง train กับ test

```
X,xtest,y,ytest= train_test_split(breast_cancer.data, breast_cancer.target,random_state=20)
tr = tree.DecisionTreeClassifier(max_depth=30)
```

- นำข้อมูลในส่วนของ train มาสร้าง model

```
tr = tree.DecisionTreeClassifier(max_depth=30)
tr = tr.fit(X, y)
```
Decision Trees

```
knear = neighbors.KNeighborsClassifier()
knear = knear.fit(X, y)
```
KNN

- นำ model ไปทดสอบหาอัตราความแม่นยำกับข้อมูลในส่วนของ test

```
print("train tree acc " + str(tr.score(X,y)))
print("test tree acc " + str(tr.score(xtest,ytest)))
```

```
train tree acc 1.0
test tree acc 0.8881118881118881
```
Decision Trees

```
print("train K-near acc " + str(knear.score(X,y)))
print("test K-near acc " + str(knear.score(xtest,ytest)))
```

```
train K-near acc 0.9483568075117371
test K-near acc 0.9300699300699301
```
KNN

logistic regression

เป็นการวิเคราะห์ dataset โดยการสร้างสมการ zigmoid $\left(\frac{1}{1+e^{-(\beta_0+\beta_1 X)}}\right)$ มีเป้าหมายในทำนายข้อมูลว่ามีโอกาสที่กลุ่มนั้นๆหรือไม่

SVM

เป็นการวิเคราะห์ dataset โดยหาเส้นแบ่งที่มีระยะห่างระหว่างตัวที่ใกล้ที่สุดของแต่ละกลุ่มมากที่สุด

ภาพ iris จากข้อมูลที่มากที่สุดในแต่ละ Attribute