

Support Vector Machine for Cancer Classification

1. Introduction

ในไฟล์ cancer.csv จะเป็นข้อมูลผู้ป่วยที่รักษาโรคมะเร็งตับ โดยจะแบ่งออกเป็น 4 ระยะ (CA Level) ซึ่งจะมีข้อมูลแต่ละคอลัมน์ดังนี้ Sex, Ascites, Hepatomegaly, Spiders, Edema, Bilirubin, Cholesterol, Albumin, Copper, Alk_Phos, SGOT, Tryglicerides, Platelets, Prothrombin, Ca level ในไฟล์ cancer.csv จะเป็นข้อมูลผู้ป่วยที่รักษาโรคมะเร็งตับ โดยจะแบ่งออกเป็น 4 ระยะ (CA Level) ซึ่งจะมีข้อมูลแต่ละคอลัมน์ดังนี้ Sex, Ascites, Hepatomegaly, Spiders, Edema, Bilirubin, Cholesterol, Albumin, Copper, Alk_Phos, SGOT, Tryglicerides, Platelets, Prothrombin, Ca level

2. Preprocessing

Sex	Ascites	Hepatomegaly	Spiders	Edema
F	Y	Y	Y	Y
F	N	Y	Y	N
M	N	N	N	S
F	N	Y	Y	S
F	N	Y	Y	N
F	N	Y	N	N

คอลัมน์เหล่านี้มีค่าข้อมูลเป็น String ซึ่งเราจะทำการเปลี่ยนให้อยู่ในรูปของตัวเลขก่อนทำการเทรน

```
df['Sex'] = df['Sex'].map({'M': 1, 'F': 0})
df['Ascites'] = df['Ascites'].map({'Y': 1, 'N': 0, 'S': 2})
df['Hepatomegaly'] = df['Hepatomegaly'].map({'Y': 1, 'N': 0, 'S': 2})
df['Spiders'] = df['Spiders'].map({'Y': 1, 'N': 0, 'S': 2})
df['Edema'] = df['Edema'].map({'Y': 1, 'N': 0, 'S': 2})

X = df.drop(columns=['CA level']) # feature
y = df['CA level'] # class
```

✓ 0.0s

Python

3. Training Model

3.1 train test split โดยกำหนด ข้อมูลที่ใช้ train กับ test อัตราส่วนคือ 7:3 และกำหนด random state = 3

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=3, shuffle=True) # แบ่งข้อมูลเป็น train และ test
```

✓ 0.0s

Python

3.2 ใช้ scaler เพื่อปรับข้อมูลในช่วย 0-1 และทำให้คำนวณเร็วขึ้น

```
scaler = StandardScaler() # ปรับข้อมูลให้อยู่ในช่วง 0-1
X_train = scaler.fit_transform(X_train) # ปรับข้อมูล train
X_test = scaler.transform(X_test) # ปรับข้อมูล test
```

✓ 0.0s

Python

3.3 สร้าง model และ เทรน

```
grid = {
    'C': [0.01, 0.1, 1, 10],
    'kernel': ['linear', 'poly', 'rbf', 'sigmoid'],
    'degree': [1, 3, 5, 7],
    'gamma': [0.01, 1]
}

svm_cv = GridSearchCV(SVC(), grid, cv=5)
svm_cv.fit(X_train, y_train)
print("Best Parameters:", svm_cv.best_params_)
print("Train Score:", svm_cv.best_score_)
print("Test Score:", svm_cv.score(X_test, y_test))
```

✓ 4.2s Python

Best Parameters: {'C': 10, 'degree': 1, 'gamma': 0.01, 'kernel': 'sigmoid'}
Train Score: 0.5072938689217759
Test Score: 0.5483870967741935

3.4 สร้าง model และ เทรนด้วย OneVsOneClassifier และ LinearSVC

```
# สร้าง model และ train ด้วย OneVsOneClassifier และ LinearSVC
clf = OneVsOneClassifier(LinearSVC(dual="auto", random_state=0)).fit(X_train, y_train)

y_pred = clf.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
```

✓ 0.0s Python

4. Result

จาก model ที่เทรนด้วยวิธีที่นำเสนอมีค่าความแม่นยำประมาณ 55%

```
Accuracy: 0.5483870967741935
=====
Confusion Matrix:
[[ 0  2  0  0]
 [ 0  5 10  2]
 [ 0  4 23  7]
 [ 1  0 16 23]]
=====
Classification Report:

```

	precision	recall	f1-score	support
1	0.00	0.00	0.00	2
2	0.45	0.29	0.36	17
3	0.47	0.68	0.55	34
4	0.72	0.57	0.64	40
accuracy			0.55	93
macro avg	0.41	0.39	0.39	93
weighted avg	0.56	0.55	0.54	93