


데이터 베이스 시스템

건강 상태를 기반으로 적절한 약물 분류

- Decision Trees -

214683 ㅅ ㅅ ㅅ

목차

- 
1. 모델링 배경 및 목적
 2. 데이터 설명
 3. 모델링 과정
 4. 결과 및 성능

1. 모델링 배경 및 목적



최근 일교차가 심해지며 감기로 고생하는 사람이 많아졌습니다.

한 달 가까이 감기 증상에 감기약을 복용하였음에도 큰 효과를 느끼지 못하면서, “현재 나에게는 어떤 약이 더 효과적일까?” 하는 궁금증이 생겼습니다.

현재 데이터셋에는 약의 종류가 정해지진 않았지만 비슷한 성향을 분류한다는 내용에 대해 집중하였습니다.

약 이야기

#137 약을 약이 되도록 하는 기술 '약물전달시스템'

약 모양·형태와 투약방법
제각기 다른 이유있지요

류장훈 기자 jh@joongang.co.kr



2. 데이터 설명

```
df = pd.read_csv("drug200.csv")
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 200 entries, 0 to 199
```

```
Data columns (total 6 columns):
```

#	Column	Non-Null Count	Dtype
0	Age	200 non-null	int64
1	Sex	200 non-null	object
2	BP	200 non-null	object
3	Cholesterol	200 non-null	object
4	Na_to_K	200 non-null	float64
5	Drug	200 non-null	object

```
dtypes: float64(1), int64(1), object(4)
```

```
memory usage: 6.3+ KB
```

200개의 데이터셋 존재

5개의 독립변수, 1개의 종속 변수

종속 변수 : Drug

0. Age : 나이

1. Sex : 성별

2. BP : 혈압 상태

3. Cholesterol : 콜레스테롤 상태

4. Na_to_K 혈중 나트륨 대비 칼륨

3. 모델링 과정

```
from sklearn.preprocessing import LabelEncoder
```

```
# Label Encoding
```

```
le_sex = LabelEncoder()
```

```
le_bp = LabelEncoder()
```

```
le_chol = LabelEncoder()
```

```
le_drug = LabelEncoder()
```

```
df['Sex'] = le_sex.fit_transform(df['Sex']) # 'F', 'M' → 0, 1
```

```
df['BP'] = le_bp.fit_transform(df['BP']) # 'HIGH', 'NORMAL', 'LOW' → 0, 1, 2
```

```
df['Cholesterol'] = le_chol.fit_transform(df['Cholesterol'])
```

```
df['Drug'] = le_drug.fit_transform(df['Drug']) # 'A'~'Y' → 0~4
```

```
x = df.drop(columns='Drug')
```

```
y = df['Drug']
```

```
x_train, x_test, y_train, y_test = train_test_split(X, y, stratify=y, random_state=
```

데이터셋이 문자로 되어있는 관계로 숫자로 인식 될 수 있도록 인코딩 과정을 거침

인코딩 이후 데이터셋 x,y분리 및 훈련

4. 결과 및 성능

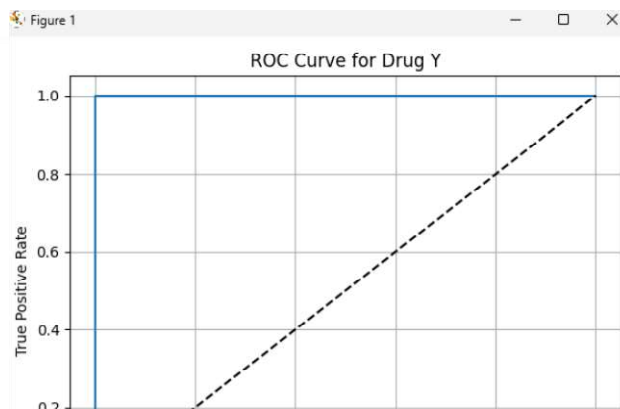
훈련 및 테스트 성능 검사

```
from sklearn.tree import DecisionTreeClassifier

dTreeAll = DecisionTreeClassifier(random_state=0)
dTreeAll.fit(x_train, y_train)

print("Train Score: {:.2f}".format(dTreeAll.score(x_train, y_train)))
print("Test Score : {:.2f}".format(dTreeAll.score(x_test, y_test)))
```

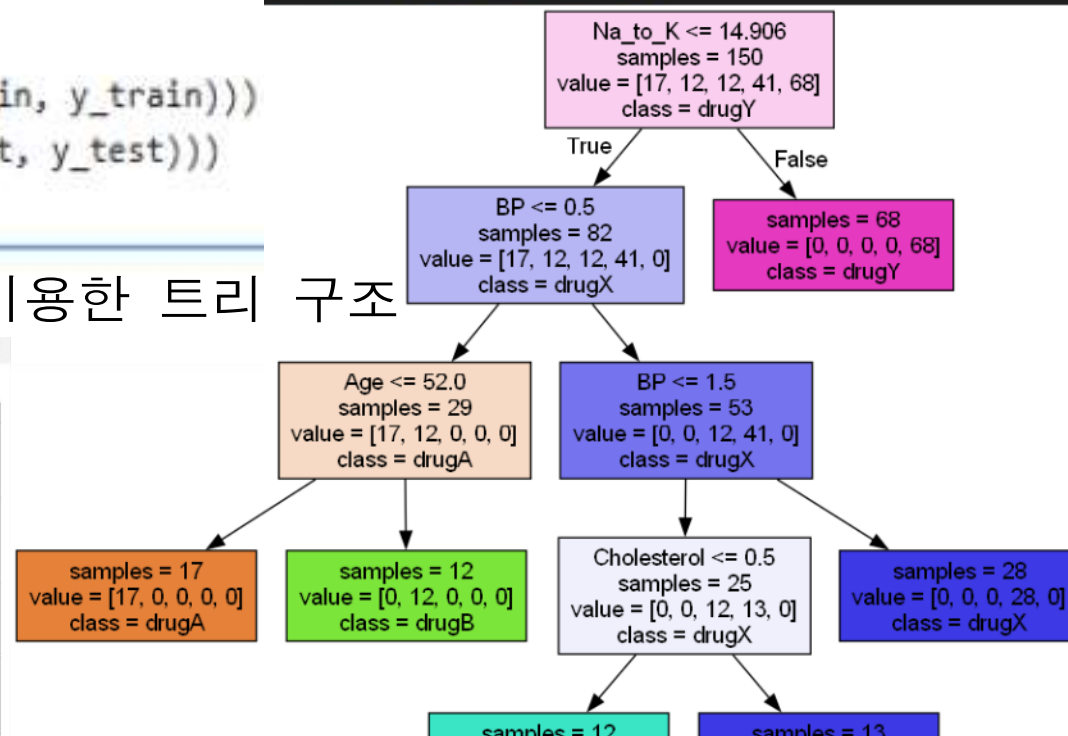
Train Score: 1.00
Test Score : 0.98



Pydot을 이용한 트리 구조

```
export_graphviz(
    dTreeAll,
    out_file="decisionTree0.dot",
    class_names=le_drug.classes_,
    feature_names=x.columns,
    impurity=False,
    filled=True
)

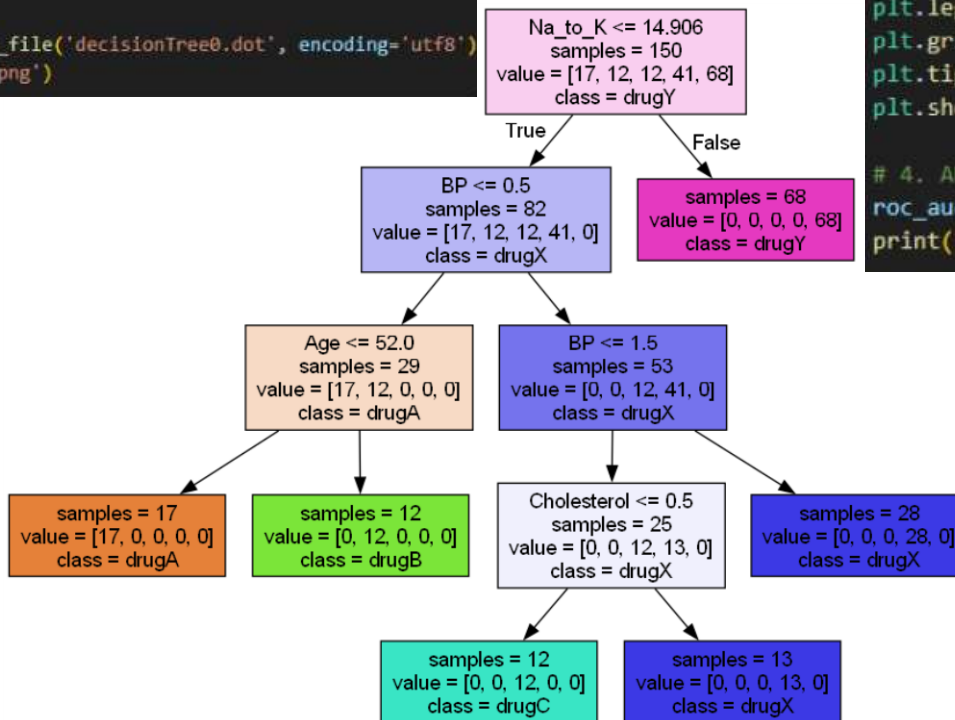
(graph,) = pydot.graph_from_dot_file('decisionTree0.dot', encoding='utf8')
graph.write_png('decisionTree0.png')
```



4. 결과 및 성능

Pydot을 이용한 트리 구조 및 Curve

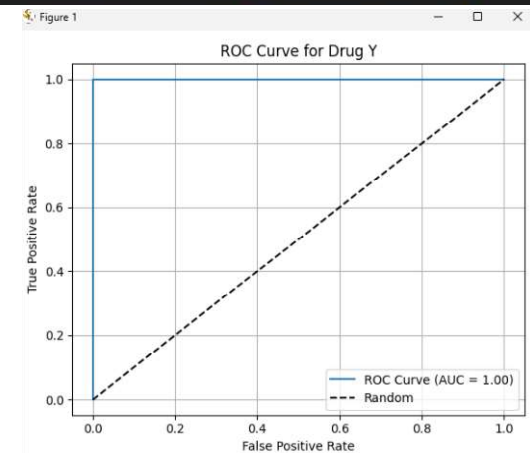
```
export_graphviz(  
    dTreeAll,  
    out_file="decisionTree0.dot",  
    class_names=le_drug.classes_,  
    feature_names=x.columns,  
    impurity=False,  
    filled=True  
)  
  
(graph,) = pydot.graph_from_dot_file('decisionTree0.dot', encoding='utf8')  
graph.write_png('decisionTree0.png')
```



```
# 3. PR Curve
precision, recall, _ = precision_recall_curve(y_test_binary, y_proba)

plt.figure(figsize=(6, 5))
plt.plot(recall, precision, label="PR Curve")
plt.xlabel("Recall")
plt.ylabel("Precision")
plt.title("Precision-Recall Curve for Drug Y")
plt.legend()
plt.grid()
plt.tight_layout()
plt.show()

# 4. AUC 값 출력
roc_auc_val = roc_auc_score(y_test_binary, y_proba)
print("ROC AUC Score (Drug Y vs others):", round(roc_auc_val, 3))
```



출처

데이터 세트 :

<https://www.kaggle.com/datasets/pablomgomez21/drugs-a-b-c-x-y-for-decision-trees>