DecisionTree

March 11, 2025

[1]:
```
# Load libraries
import pandas as pd
from sklearn.tree import DecisionTreeClassifier # Import Decision Tree␣
  ↪Classifier
from sklearn.model_selection import train_test_split # Import train_test_split␣
  ↪function
from sklearn import metrics #Import scikit-learn metrics module for accuracy␣
  ↪calculation
```

[2]:
```
col_names = ['pregnant', 'glucose', 'bp', 'skin', 'insulin', 'bmi', 'pedigree',␣
  ↪'age', 'label']
# load dataset
pima = pd.read_csv("diabetes.csv", header=None, names=col_names)
```

[3]:
```
pima.head()
```

[3]:
```
      pregnant  glucose             bp           skin  insulin   bmi  \
0  Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI
1            6      148             72             35        0  33.6
2            1       85             66             29        0  26.6
3            8      183             64              0        0  23.3
4            1       89             66             23       94  28.1

                   pedigree  age    label
0  DiabetesPedigreeFunction  Age  Outcome
1                     0.627   50        1
2                     0.351   31        0
3                     0.672   32        1
4                     0.167   21        0
```

[4]:
```
#split dataset in features and target variable
feature_cols = ['pregnant', 'insulin', 'bmi', 'age','glucose','bp','pedigree']
X = pima[feature_cols] # Features
y = pima.label # Target variable
```

[5]:
```
# Split dataset into training set and test set
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,␣
 ↪random_state=1) # 70% training and 30% test
```

[12]:
```
# Create Decision Tree classifier
clf = DecisionTreeClassifier()

# Train the model
clf.fit(X_train, y_train)
```

[12]:
```
DecisionTreeClassifier()
```

[13]:
```
# Split dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,␣
 ↪random_state=1) # 70% training and 30% test
```

[18]:
```
# 1. Import necessary libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score

# 2. Load dataset
df = pd.read_csv("diabetes.csv")  # Ensure this file exists
X = df.iloc[:, :-1]  # Features
y = df.iloc[:, -1]   # Target

# 3. Split dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

# 4. Train the Decision Tree Classifier
clf = DecisionTreeClassifier()
clf.fit(X_train, y_train)

# 5. Make Predictions
y_pred = clf.predict(X_test)  # Ensure this line runs

# 6. Calculate Accuracy
accuracy = accuracy_score(y_test, y_pred)  # Use accuracy_score from sklearn.
 ↪metrics
print(f"Model Accuracy: {accuracy:.2f}")
```

```
Model Accuracy: 0.74
```

[24]:
```
!pip install pydotplus
```

```
Collecting pydotplus
  Downloading pydotplus-2.0.2.tar.gz (278 kB)
     |                  | 278 kB 9.3 MB/s eta 0:00:01
Requirement already satisfied: pyparsing>=2.0.1 in
/home/exam/.local/lib/python3.8/site-packages (from pydotplus) (3.1.4)
Building wheels for collected packages: pydotplus
  Building wheel for pydotplus (setup.py) … done
  Created wheel for pydotplus: filename=pydotplus-2.0.2-py3-none-any.whl
size=24566
sha256=48e7c0372713a21be8a619ed861a3a575633db2fc79691702244d0f984d6f095
  Stored in directory: /home/exam/.cache/pip/wheels/fe/cd/78/a7e873cc049759194f8
271f780640cf96b35e5a48bef0e2f36
Successfully built pydotplus
Installing collected packages: pydotplus
Successfully installed pydotplus-2.0.2
```

[25]:

```
!pip install graphviz
```

```
Collecting graphviz
  Downloading graphviz-0.20.3-py3-none-any.whl (47 kB)
     |                  | 47 kB 4.4 MB/s eta 0:00:01
Installing collected packages: graphviz
Successfully installed graphviz-0.20.3
```

[28]:

```
print(f"Model was trained with {clf.n_features_in_} features.")
print(f"Feature columns provided: {feature_cols}")
print(f"Number of feature columns provided: {len(feature_cols)}")
```

```
Model was trained with 8 features.
Feature columns provided: ['pregnant', 'insulin', 'bmi', 'age', 'glucose', 'bp',
'pedigree']
Number of feature columns provided: 7
```

[29]:

```
feature_cols = X.columns.tolist()  # Get correct feature names
```

[33]:

```
pip install graphviz pydotplus
```

```
Requirement already satisfied: graphviz in /home/exam/.local/lib/python3.8/site-
packages (0.20.3)
Requirement already satisfied: pydotplus in
/home/exam/.local/lib/python3.8/site-packages (2.0.2)
Requirement already satisfied: pyparsing>=2.0.1 in
/home/exam/.local/lib/python3.8/site-packages (from pydotplus) (3.1.4)
Note: you may need to restart the kernel to use updated packages.
```

[34]:

```
from sklearn.tree import export_graphviz
from io import StringIO  #  FIX: Use `io.StringIO` instead of `sklearn.
 ↪externals.six`
```

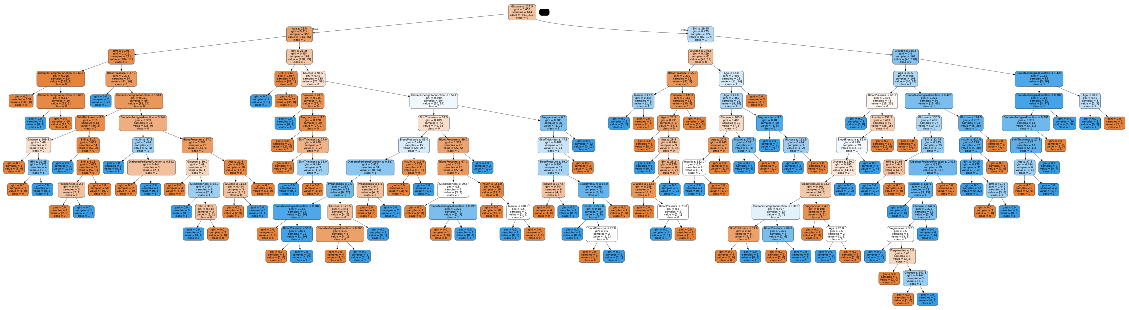
```
from IPython.display import Image
import pydotplus

# Ensure feature_cols matches the model's feature count
feature_cols = X.columns.tolist()

# Generate dot data
dot_data = StringIO()
export_graphviz(clf, out_file=dot_data,
                filled=True, rounded=True,
                special_characters=True, feature_names=feature_cols,␣
 ↪class_names=['0', '1'])

# Convert to graph and display
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png('diabetes.png')
Image(graph.create_png())  # Display image
```

[34]:



[35]:

```
# Create Decision Tree classifer object
clf = DecisionTreeClassifier(criterion="entropy", max_depth=3)

# Train Decision Tree Classifer
clf = clf.fit(X_train,y_train)

#Predict the response for test dataset
y_pred = clf.predict(X_test)

# Model Accuracy, how often is the classifier correct?
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.7662337662337663
```

[ ]: