Abhishek Murthy
21BDS0064
Fall Sem 2024-2025
DA - 3
Data Mining Lab
10-09-2024

Loading the dataset

In [1]:
```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier, export_graphviz
from sklearn.preprocessing import LabelEncoder
import graphviz
from sklearn.model_selection import train_test_split
```

In [4]:
```
df = pd.read_csv('/content/drive/MyDrive/drug_data.csv')
```

In [3]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

In [5]:
```
df.head()
```

Out[5]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	drugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	drugY

In [6]:
```
sex_encoder = LabelEncoder()
df['Sex'] = sex_encoder.fit_transform(df['Sex'])

bp_encoder = LabelEncoder()
df['BP'] = bp_encoder.fit_transform(df['BP'])

cholesterol_encoder = LabelEncoder()
df['Cholesterol'] = cholesterol_encoder.fit_transform(df['Cholesterol'])

drug_encoder = LabelEncoder()
df['Drug'] = drug_encoder.fit_transform(df['Drug'])
```

In [7]:
```
X = df.drop('Drug', axis=1)
y = df['Drug']
```

In [8]:
```
df.head()
```

Out[8]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	0	0	0	25.355	4
1	47	1	1	0	13.093	2
2	47	1	1	0	10.114	2
3	28	0	2	0	7.798	3
4	61	0	1	0	18.043	4

In [9]:
```
y = df['Drug']
X = df.drop(['Drug'], axis = 1)
```

In [10]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_sta
```

In [11]:
```
from sklearn.model_selection import RandomizedSearchCV
```

In [12]:
```
parameter = {
    'max_depth': (10,30,50,70,90,100),
    'criterion' : ('gini', 'entropy'),
    'max_features' : ('sqrt', 'log2'),
    'min_samples_split' : (2,4,6)
}
```

In [13]:
```
DT_grid = RandomizedSearchCV(DecisionTreeClassifier(), param_distributions = parame
DT_grid.fit(X_train, y_train)
```

```
Fitting 5 folds for each of 10 candidates, totalling 50 fits
```

Out[13]:
```
▸       RandomizedSearchCV
▸ estimator: DecisionTreeClassifier
      ▸ DecisionTreeClassifier
```

In [14]:
```
DT_grid.best_estimator_
```

Out[14]:
```
▾                 DecisionTreeClassifier
DecisionTreeClassifier(max_depth=70, max_features='log2')
```

In [15]:
```
DT_model = DecisionTreeClassifier(criterion='entropy', max_depth=30, max_features='
DT_model.fit(X_train, y_train)
DT_predicted = DT_model.predict(X_test)
```

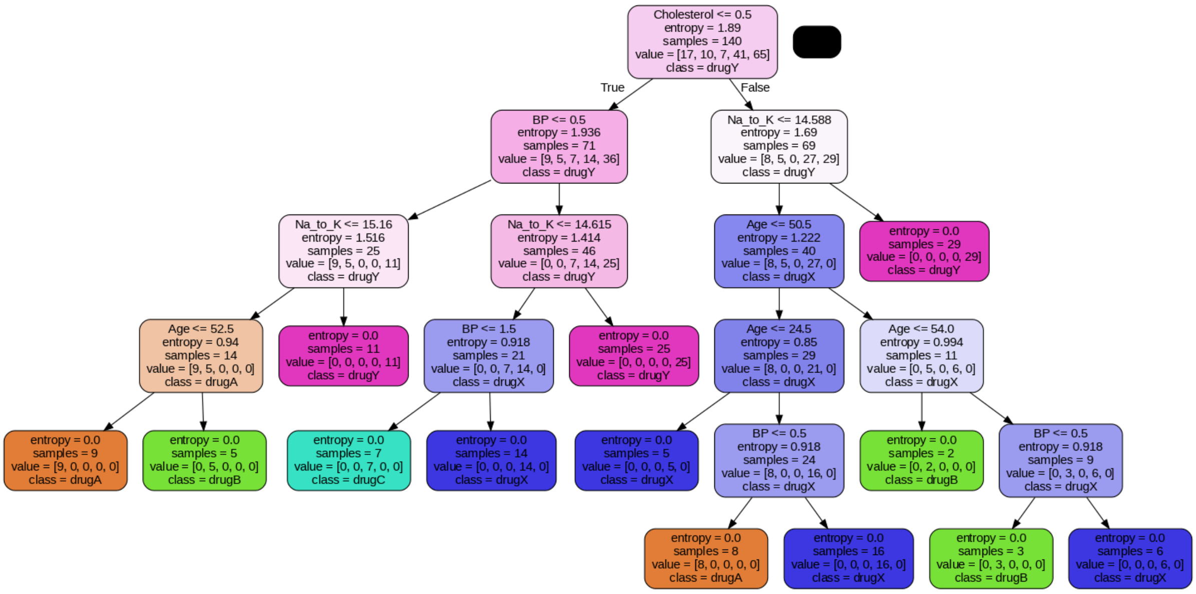
In [16]:
```
print(f'Test accuracy is {DT_model.score(X_test, y_test)}')
```

```
Test accuracy is 0.9666666666666667
```

Decision Tree plot

In [17]:
```
import pydotplus
from IPython.display import Image
dot_data = export_graphviz(DT_model,
                                feature_names=X.columns,class_names=['drugA', 'drug
'drugY'],
                                out_file=None,
                                filled=True,
                                rounded=True)

pydot_graph = pydotplus.graph_from_dot_data(dot_data)
pydot_graph.set_size('"15,15!"')
Image(pydot_graph.create_png())
```

Out[17]:



Naive Bayes

In [18]:

```
from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
model.fit(X_train, y_train)
predicted_bayes = model.predict(X_test)
```

Comparison

In [19]:

```
print(f'Decision Tree accuracy is {DT_model.score(X_test, y_test)}')
print(f'Bayesian accuracy is {model.score(X_test, y_test)}')
```

```
Decision Tree accuracy is 0.9666666666666667
Bayesian accuracy is 0.8666666666666667
```

In [20]:

```
new_patient = pd.DataFrame({
'Age': [43],
'Sex': sex_encoder.fit_transform(['M']),
'BP': bp_encoder.fit_transform(['LOW']),
'Cholesterol': cholesterol_encoder.fit_transform(['HIGH']),
'Na_to_K': [15.376]
})
```

In [21]:

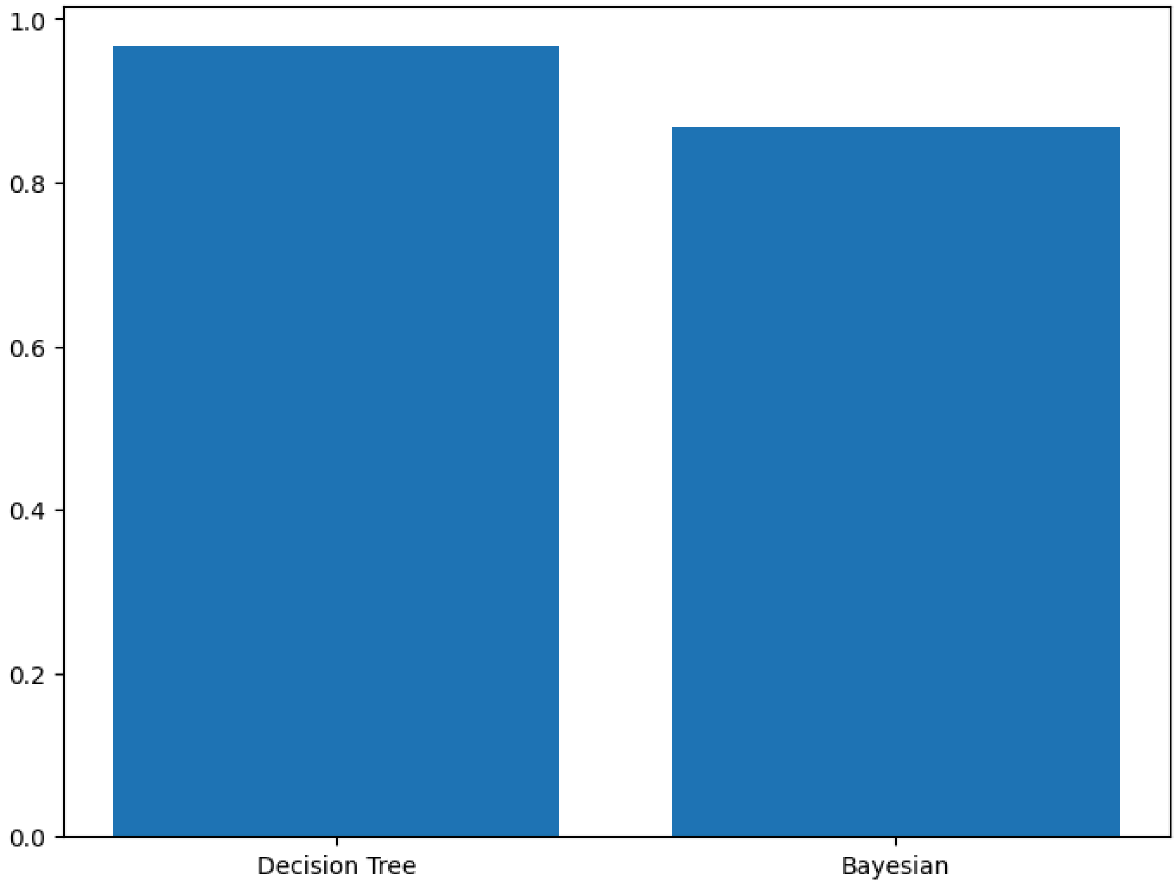
```
# Predict the drug using the Decision Tree model
predicted_drug_dt = DT_model.predict(new_patient)
print("Predicted Drug by Decision Tree:",
      drug_encoder.inverse_transform(predicted_drug_dt))

# Predict the drug using the Naive Bayes model
predicted_drug_nb = model.predict(new_patient)
print("Predicted Drug by Naive Bayes:",
      drug_encoder.inverse_transform(predicted_drug_nb))
```

```
Predicted Drug by Decision Tree: ['drugY']
Predicted Drug by Naive Bayes: ['drugA']
```

In [22]:

```
import matplotlib.pyplot as plt
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
classifiers = ['Decision Tree', 'Bayesian']
```

```
accuracies = [DT_model.score(X_test, y_test), model.score(X_test, y_test)]
ax.bar(classifiers,accuracies)
plt.show()
```


Decision Tree
Bayesian