

Name: **Varun Sudhir**

Reg No: **21BDS0040**

Data Mining Lab Digital Assignment – 3

Consider the dataset below, the independent variables in the dataset are {Age, sex, BP, Cholesterol, .Na_to_K} and the dependent variable is {Drug}.

Your task is to predict the type of the drug should be given if a patient has the following parameters

43 M LOW HIGH 15.376

First, let's load the dataset

```
# Varun Sudhir 21BDS0040
```

```
# Loading the dataset
```

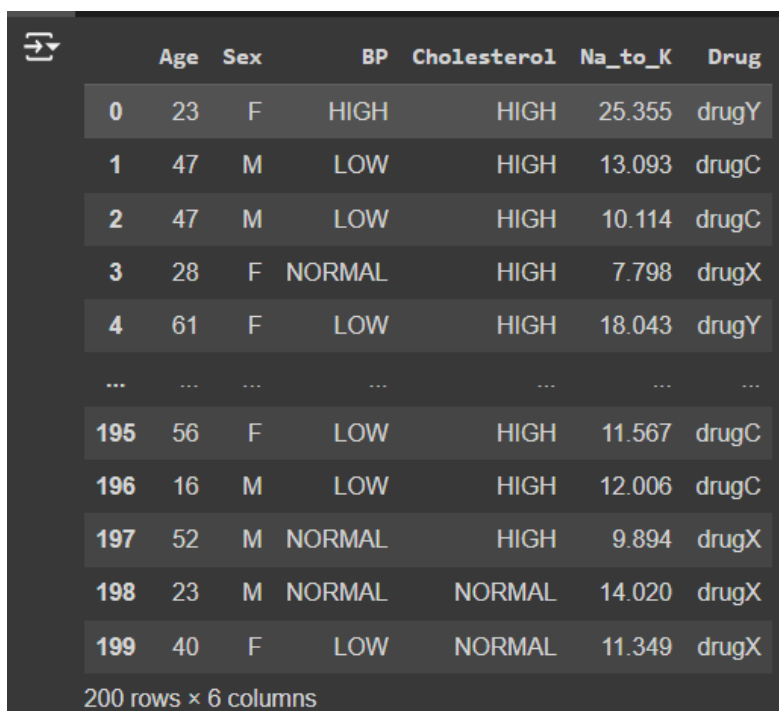
```
import pandas as pd
```

```
from sklearn.model_selection import train_test_split
```

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

```
df = pd.read_csv('drug_data.csv')
```

```
df
```



	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
...
195	56	F	LOW	HIGH	11.567	drugC
196	16	M	LOW	HIGH	12.006	drugC
197	52	M	NORMAL	HIGH	9.894	drugX
198	23	M	NORMAL	NORMAL	14.020	drugX
199	40	F	LOW	NORMAL	11.349	drugX

200 rows × 6 columns

Perform data preparation

```
# Varun Sudhir 21BDS0040
le_sex = LabelEncoder()
le_bp = LabelEncoder()
le_cholesterol = LabelEncoder()
le_drug = LabelEncoder()

# Fit encoders on respective columns in the dataset
df['Sex'] = le_sex.fit_transform(df['Sex'])
df['BP'] = le_bp.fit_transform(df['BP'])
df['Cholesterol'] = le_cholesterol.fit_transform(df['Cholesterol'])
df['Drug'] = le_drug.fit_transform(df['Drug'])

# Independent variables (features) and dependent variable (target)
X = df.drop('Drug', axis=1)
y = df['Drug']

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

1) Perform the prediction using decision tree classification with hyperparameter tuning and find the best parameter which gives higher accuracy.

```
# Varun Sudhir 21BDS0040

from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score

# Define the parameter grid
param_grid = {
    'max_depth': [3, 5, 7, 10],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
}

# Create a decision tree classifier
dt = DecisionTreeClassifier(random_state=42)

# Use GridSearchCV for hyperparameter tuning
grid_search = GridSearchCV(dt, param_grid, cv=5, scoring='accuracy')
grid_search.fit(X_train, y_train)
```

```

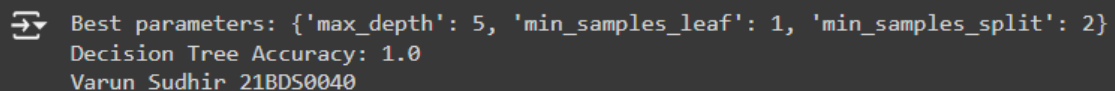
# Best parameters
best_params = grid_search.best_params_
print("Best parameters:", best_params)

# Predict on the test set
y_pred_dt = grid_search.predict(X_test)

# Calculate accuracy
accuracy_dt = accuracy_score(y_test, y_pred_dt)
print("Decision Tree Accuracy:", accuracy_dt)
print("Varun Sudhir 21BDS0040")

```

Output:



```

➡ Best parameters: {'max_depth': 5, 'min_samples_leaf': 1, 'min_samples_split': 2}
Decision Tree Accuracy: 1.0
Varun Sudhir 21BDS0040

```

2. Visualize the decision tree

```

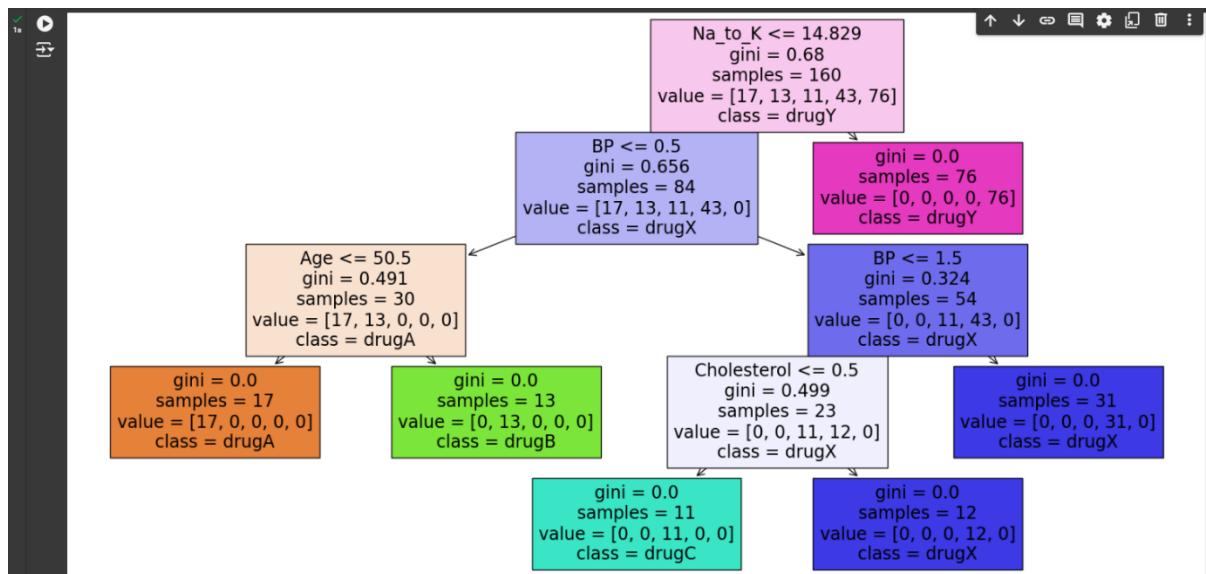
# Varun Sudhir 21BDS0040

from sklearn import tree
import matplotlib.pyplot as plt

# Plot the tree
plt.figure(figsize=(20,10))
tree.plot_tree(grid_search.best_estimator_, filled=True,
feature_names=X.columns, class_names=['drugA', 'drugB', 'drugC', 'drugX',
'drugY'])
plt.show()
print()

```

```
print("Varun Sudhir 21BDS0040")
```



3) For the same data perform a Bayesian classification with any Bayesian classifier and compare the accuracy between decision tree and Bayesian classifier models.

```
# Varun Sudhir 21BDS0040
```

```
from sklearn.naive_bayes import GaussianNB
```

```
# Create a Naive Bayes classifier
```

```
nb = GaussianNB()
```

```
# Fit the model
```

```
nb.fit(X_train, y_train)
```

```
# Predict on the test set
```

```
y_pred_nb = nb.predict(X_test)
```

```
# Calculate accuracy
```

```
accuracy_nb = accuracy_score(y_test, y_pred_nb)
```

```
print("Naive Bayes Accuracy:", accuracy_nb)
```

```
print("Varun Sudhir 21BDS0040")
```



```
Naive Bayes Accuracy: 0.925
Varun Sudhir 21BDS0040
```

4. Plot the accuracy of both the classifiers

```
# Varun Sudhir 21BDS0040
```

```
import matplotlib.pyplot as plt
```

```
# Plot the accuracy of both classifiers
```

```
accuracies = [accuracy_dt, accuracy_nb]
```

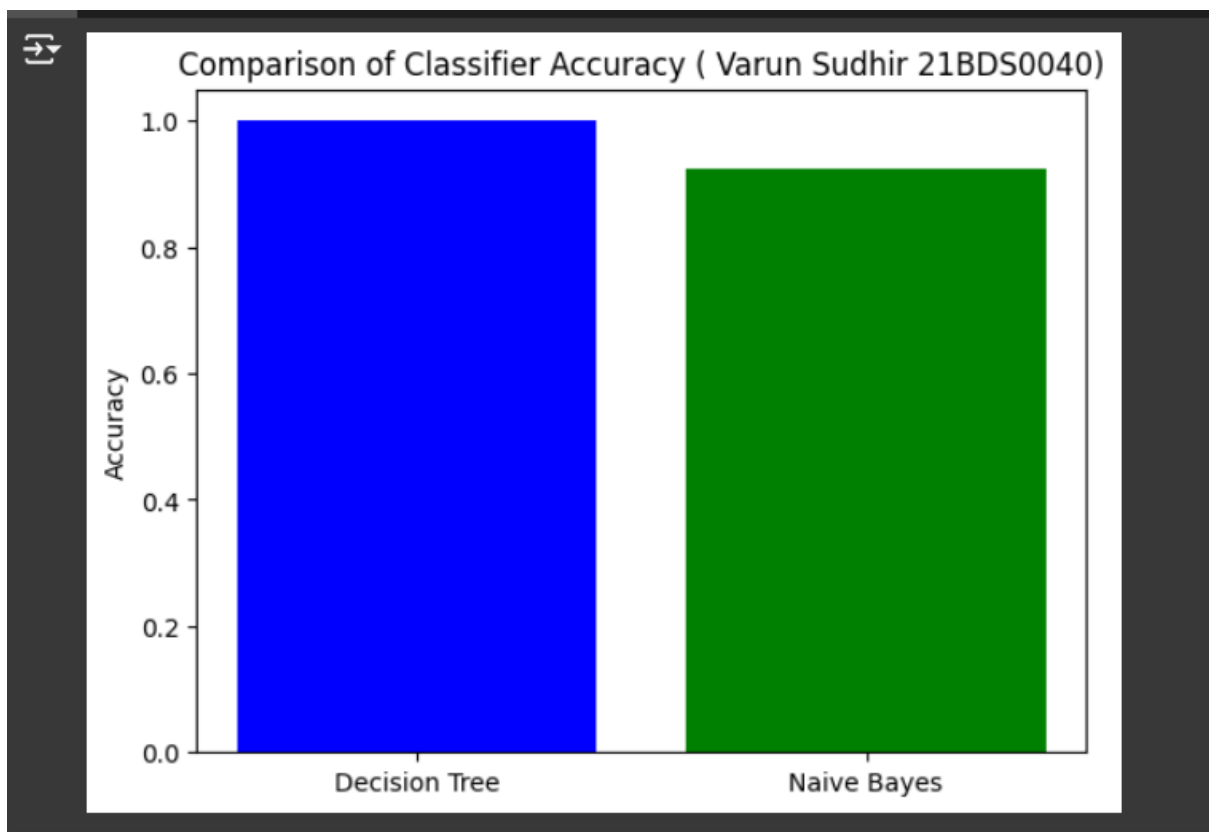
```
labels = ['Decision Tree', 'Naive Bayes']
```

```
plt.bar(labels, accuracies, color=['blue', 'green'])
```

```
plt.ylabel('Accuracy')
```

```
plt.title('Comparison of Classifier Accuracy ( Varun Sudhir 21BDS0040)')
```

```
plt.show()
```



Predicting the outcome for the following parameters

43 M LOW HIGH 15.376

```
# New patient data (for prediction)
new_patient = pd.DataFrame({
    'Age': [43],
    'Sex': le_sex.transform(['M']),
    'BP': le_bp.transform(['LOW']),
    'Cholesterol': le_cholesterol.transform(['HIGH']),
    'Na_to_K': [15.376]
})

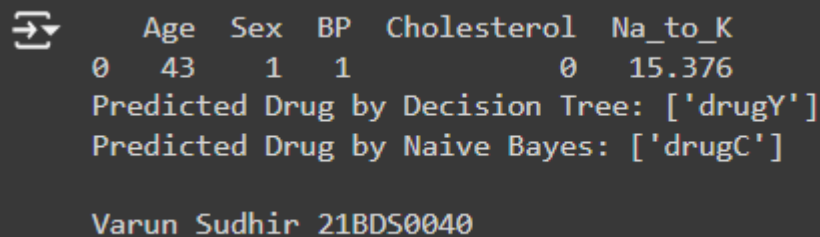
print(new_patient)

# Predict the drug using the decision tree model
predicted_drug_dt = grid_search.predict(new_patient)
print("Predicted Drug by Decision Tree:",
      le_drug.inverse_transform(predicted_drug_dt))

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

print()
print("Varun Sudhir 21BDS0040")
```

Output:



```
➡ Age Sex BP Cholesterol Na_to_K
0 43 1 1 0 15.376
Predicted Drug by Decision Tree: ['drugY']
Predicted Drug by Naive Bayes: ['drugC']

Varun Sudhir 21BDS0040
```