sion-tree-alogorithm-using-drug200

March 25, 2025

```
[8]: import pandas as pd
     from sklearn.tree import DecisionTreeClassifier
     from sklearn.model_selection import train_test_split
     from sklearn.metrics import accuracy_score, mean_squared_error
     df=pd.read_csv("/content/drug200.csv")
     df
```

```
[8]:      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 x 6 columns]
```

```
[21]: df.head()
```

```
[21]:    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
```

```
[18]: df.tail(10)
```

```
[18]:      Age Sex      BP Cholesterol  Na_to_K   Drug
      190   58   M    HIGH        HIGH   18.991  drugY
      191   23   M    HIGH        HIGH    8.011  drugA
      192   72   M     LOW        HIGH   16.310  drugY
```

```
193   72   M     LOW         HIGH   6.769  drugC
194   46   F    HIGH         HIGH  34.686  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
```

[20]:
```
df.sample(10)
```

[20]:
```
     Age Sex      BP Cholesterol  Na_to_K   Drug
142   60   M    HIGH      NORMAL    8.621  drugB
58    60   M  NORMAL      NORMAL   10.091  drugX
18    23   M     LOW        HIGH    7.298  drugC
129   32   F  NORMAL        HIGH    7.477  drugX
167   57   F  NORMAL        HIGH   14.216  drugX
124   53   F    HIGH      NORMAL   12.495  drugB
101   45   F    HIGH        HIGH   12.854  drugA
185   57   F  NORMAL      NORMAL   25.893  drugY
128   47   M     LOW      NORMAL   33.542  drugY
3     28   F  NORMAL        HIGH    7.798  drugX
```

[22]:
```
df.isnull()
```

[22]:
```
       Age    Sex     BP  Cholesterol  Na_to_K   Drug
0    False  False  False        False    False  False
1    False  False  False        False    False  False
2    False  False  False        False    False  False
3    False  False  False        False    False  False
4    False  False  False        False    False  False
..     …      …      …           …      …      …
195  False  False  False        False    False  False
196  False  False  False        False    False  False
197  False  False  False        False    False  False
198  False  False  False        False    False  False
199  False  False  False        False    False  False

[200 rows x 6 columns]
```

[23]:
```
df.duplicated()
```

[23]:
```
0      False
1      False
2      False
3      False
4      False
       …
```

```
195    False
196    False
197    False
198    False
199    False
Length: 200, dtype: bool
```

[24]:

```
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: 9.5+ KB
```

[25]:

```
df.describe()
```

[25]:

```
              Age     Na_to_K
count  200.000000  200.000000
mean    44.315000   16.084485
std     16.544315    7.223956
min     15.000000    6.269000
25%     31.000000   10.445500
50%     45.000000   13.936500
75%     58.000000   19.380000
max     74.000000   38.247000
```

[29]:

```
df.columns
```

[29]:

```
Index(['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K', 'Drug'], dtype='object')
```

[30]:

```
df.dtypes
```

[30]:

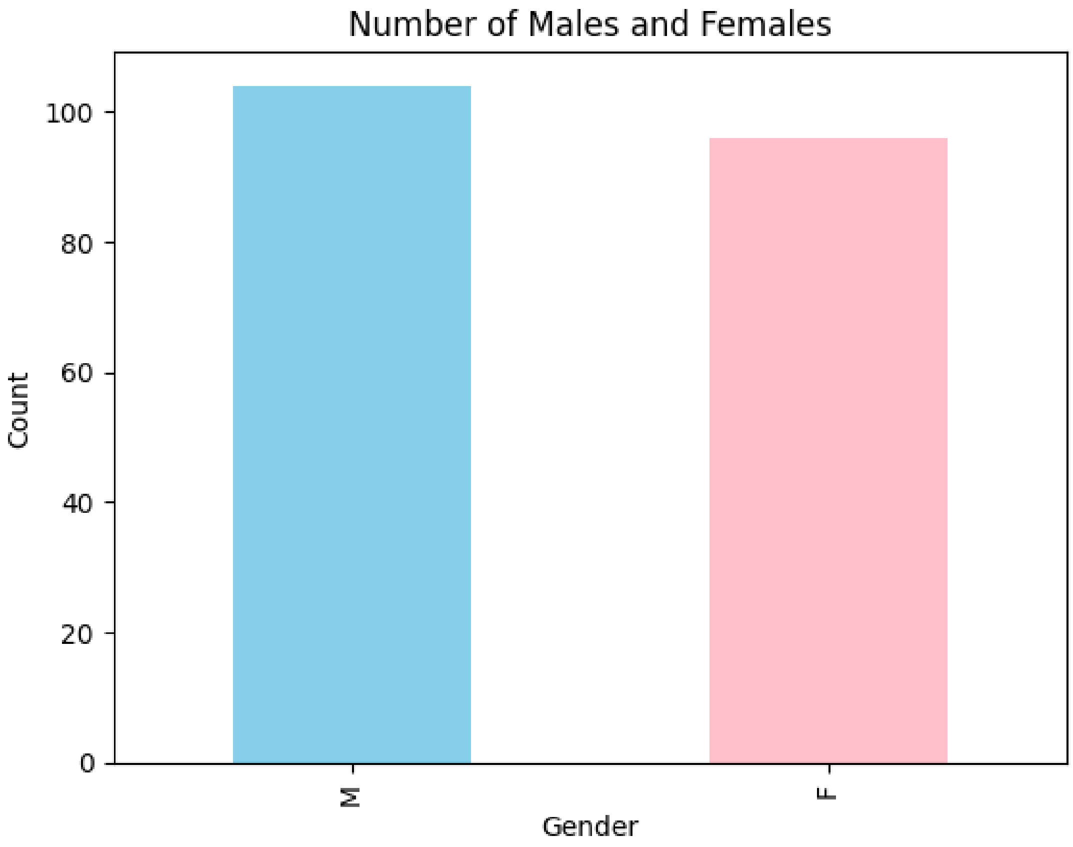
```
Age              int64
Sex             object
BP              object
Cholesterol     object
Na_to_K        float64
Drug            object
```

```
dtype: object
```

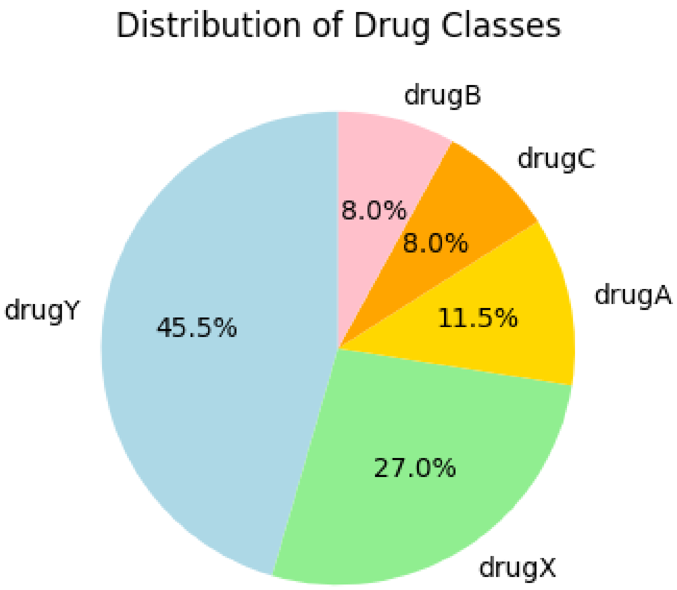
```
[31]: df.shape #number rows and columns
```

```
[31]: (200, 6)
```

```
[37]: #find the number of male and female
      import matplotlib.pyplot as plt
      gender_counts = df['Sex'].value_counts()
      gender_counts.plot(kind='bar', color=['skyblue', 'pink'])
      plt.title('Number of Males and Females')
      plt.xlabel('Gender')
      plt.ylabel('Count')
      plt.show()
```



```
[44]: #finding the number of class present in drug column
      drug_counts = df['Drug'].value_counts()
      plt.figure(figsize=(6, 4))
      plt.pie(drug_counts.values, labels=drug_counts.index, autopct='%1.1f%%',␣
        ↪startangle=90,
              colors=['lightblue', 'lightgreen', 'gold', 'orange', 'pink'])
      plt.title('Distribution of Drug Classes')
      plt.show()
```



[53]:

```
from sklearn.preprocessing import LabelEncoder
# Create a LabelEncoder
le = LabelEncoder()
df['Sex'] = le.fit_transform(df['Sex'])
df['BP'] = le.fit_transform(df['BP'])
df['Cholesterol'] = le.fit_transform(df['Cholesterol'])
df['Drug'] = le.fit_transform(df['Drug'])
df
```

[53]:

```
      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
..    …    …   ..           …       …     …
195    56    0   1            0   11.567     2
196    16    1   1            0   12.006     2
197    52    1   2            0    9.894     3
198    23    1   2            1   14.020     3
199    40    0   1            1   11.349     3
```

```
[200 rows x 6 columns]
```

[47]:

```
X = df[['Age','Sex','BP','Cholesterol','Na_to_K']]
Y = df['Drug']
```

[48]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2,␣
 ↪random_state=42)
```

[49]:

```
from sklearn.tree import DecisionTreeClassifier, export_text
dtree=DecisionTreeClassifier(criterion='gini',max_depth=3, random_state=42)
dtree.fit(X_train,Y_train)
```

[49]:

```
DecisionTreeClassifier(max_depth=3, random_state=42)
```

[50]:

```
Y_pred = dtree.predict(X_test)
```

[51]:

```
from sklearn.metrics import confusion_matrix
print("Accuracy:",accuracy_score(Y_test, Y_pred))
print("\nConfusion Matrix:\n",confusion_matrix(Y_test,Y_pred))
```

```
Accuracy: 0.875

Confusion Matrix:
 [[ 6  0  0  0  0]
 [ 0  3  0  0  0]
 [ 0  0  0  5  0]
 [ 0  0  0 11  0]
 [ 0  0  0  0 15]]
```