

# DSBDA practical 6

May 14, 2023

```
[2]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
[3]: df= pd.read_csv("D:\College Practicals\DSBDApractical6\PlayTennis.csv")
```

```
[4]: df
```

```
[4]:      Outlook Temperature Humidity   Wind Play Tennis
0      Sunny           Hot      High   Weak          No
1      Sunny           Hot      High Strong          No
2  Overcast           Hot      High   Weak          Yes
3       Rain          Mild      High   Weak          Yes
4       Rain          Cool   Normal   Weak          Yes
5       Rain          Cool   Normal Strong          No
6  Overcast          Cool   Normal Strong          Yes
7      Sunny          Mild      High   Weak          No
8      Sunny          Cool   Normal   Weak          Yes
9       Rain          Mild   Normal   Weak          Yes
10     Sunny          Mild   Normal Strong          Yes
11  Overcast          Mild      High Strong          Yes
12  Overcast           Hot   Normal   Weak          Yes
13     Rain          Mild      High Strong          No
```

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

```
[5]:      Outlook Temperature Humidity   Wind Play Tennis
0      False          False      False False          False
1      False          False      False False          False
2      False          False      False False          False
3      False          False      False False          False
4      False          False      False False          False
5      False          False      False False          False
6      False          False      False False          False
7      False          False      False False          False
8      False          False      False False          False
9      False          False      False False          False
10     False          False      False False          False
```

11	False	False	False	False	False
12	False	False	False	False	False
13	False	False	False	False	False

```
[6]: ndf=df
ndf.fillna(0)
```

```
[6]:      Outlook Temperature Humidity   Wind Play Tennis
0      Sunny           Hot     High   Weak         No
1      Sunny           Hot     High Strong         No
2  Overcast           Hot     High   Weak         Yes
3       Rain           Mild     High   Weak         Yes
4       Rain           Cool  Normal   Weak         Yes
5       Rain           Cool  Normal Strong         No
6  Overcast           Cool  Normal Strong         Yes
7      Sunny           Mild     High   Weak         No
8      Sunny           Cool  Normal   Weak         Yes
9       Rain           Mild  Normal   Weak         Yes
10     Sunny           Mild  Normal Strong         Yes
11  Overcast           Mild     High Strong         Yes
12  Overcast           Hot   Normal   Weak         Yes
13     Rain           Mild     High Strong         No
```

```
[7]: from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
df['Temperature'] = le.fit_transform(df['Temperature'])
```

```
[8]: df
```

```
[8]:      Outlook Temperature Humidity   Wind Play Tennis
0      Sunny           1     High   Weak         No
1      Sunny           1     High Strong         No
2  Overcast           1     High   Weak         Yes
3       Rain           2     High   Weak         Yes
4       Rain           0  Normal   Weak         Yes
5       Rain           0  Normal Strong         No
6  Overcast           0  Normal Strong         Yes
7      Sunny           2     High   Weak         No
8      Sunny           0  Normal   Weak         Yes
9       Rain           2  Normal   Weak         Yes
10     Sunny           2  Normal Strong         Yes
11  Overcast           2     High Strong         Yes
12  Overcast           1  Normal   Weak         Yes
13     Rain           2     High Strong         No
```

```
[9]: # Define the independent and dependent variables
X = df[['Outlook', 'Temperature', 'Humidity', 'Wind']]
Y = df['Play Tennis']
```

```
[10]: # Print the first few rows of the X dataframe
print(X.head())
```

	Outlook	Temperature	Humidity	Wind
0	Sunny	1	High	Weak
1	Sunny	1	High	Strong
2	Overcast	1	High	Weak
3	Rain	2	High	Weak
4	Rain	0	Normal	Weak

```
[11]: # Print the first few rows of the Y dataframe
print(Y.head())
```

```
0    No
1    No
2    Yes
3    Yes
4    Yes
Name: Play Tennis, dtype: object
```

```
[25]: from sklearn.model_selection import train_test_split

# 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)
```

```
[58]: !pip install --upgrade pandas
```

```
Requirement already satisfied: pandas in d:\anaconda\lib\site-packages (2.0.1)
Requirement already satisfied: numpy>=1.20.3 in d:\anaconda\lib\site-packages
(from pandas) (1.21.5)
Requirement already satisfied: pytz>=2020.1 in d:\anaconda\lib\site-packages
(from pandas) (2022.1)
Requirement already satisfied: python-dateutil>=2.8.2 in d:\anaconda\lib\site-
packages (from pandas) (2.8.2)
Requirement already satisfied: tzdata>=2022.1 in d:\anaconda\lib\site-packages
(from pandas) (2023.3)
Requirement already satisfied: six>=1.5 in d:\anaconda\lib\site-packages (from
python-dateutil>=2.8.2->pandas) (1.16.0)
```

```
[59]: X = df.get_dummies(X, columns=['Outlook', 'Temperature', 'Humidity', 'Wind'])
```

```
-----
AttributeError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_4124\112829081.py in <module>
```

```

----> 1 X = df.get_dummies(X, columns=['Outlook', 'Temperature', 'Humidity',
↳ 'Wind'])

D:\Anaconda\lib\site-packages\pandas\core\generic.py in __getattr__(self, name)
    5573         Parameters
    5574         -----
-> 5575         n : int, default 5
    5576             Number of rows to select.
    5577

AttributeError: 'DataFrame' object has no attribute 'get_dummies'

```

```

[ ]: # Print the shape of the training sets
print("X_train shape:", X_train.shape)

```

```

[ ]: # Print the shape of the testing sets
print("X_test shape:", X_test.shape)

```

```

[ ]: # Print the shape of the training sets
print("Y_train shape:", Y_train.shape)

```

```

[ ]: # Print the shape of the testing sets
print("Y_test shape:", Y_test.shape)

```

```

[60]: from sklearn.naive_bayes import GaussianNB

# Create an instance of the Gaussian Naive Bayes model
model = GaussianNB()

# Train the model using the training set
model.fit(X_train, Y_train)

```

```

[60]: GaussianNB()

```

```

[61]: # Use the model to make predictions on the testing set
Y_pred = model.predict(X_test)

```

```

[62]: from sklearn.metrics import precision_score, confusion_matrix,
↳ accuracy_score, recall_score

accuracy= accuracy_score( Y_test, Y_pred)

```

```

[63]: print("Accuracy:", accuracy)

```

```

Accuracy: 0.6666666666666666

```

```

[64]: precision = precision_score(Y_test, Y_pred, average='micro')

```

```
[65]: print("Precision:", precision)
```

```
Precision: 0.6666666666666666
```

```
[66]: recall = recall_score(Y_test,Y_pred, average='micro')
```

```
[67]: print("Recall:", recall)
```

```
Recall: 0.6666666666666666
```

```
[68]: from sklearn.metrics import confusion_matrix
```

```
[69]: cm = confusion_matrix(Y_test,Y_pred)
```

```
[70]: print("Confusion matrix on training data:\n", cm)
```

```
Confusion matrix on training data:
```

```
[[1 0]
```

```
[1 1]]
```

```
[ ]:
```