

In [1]:

```
import numpy as np
import pandas as pd
```

In [2]:

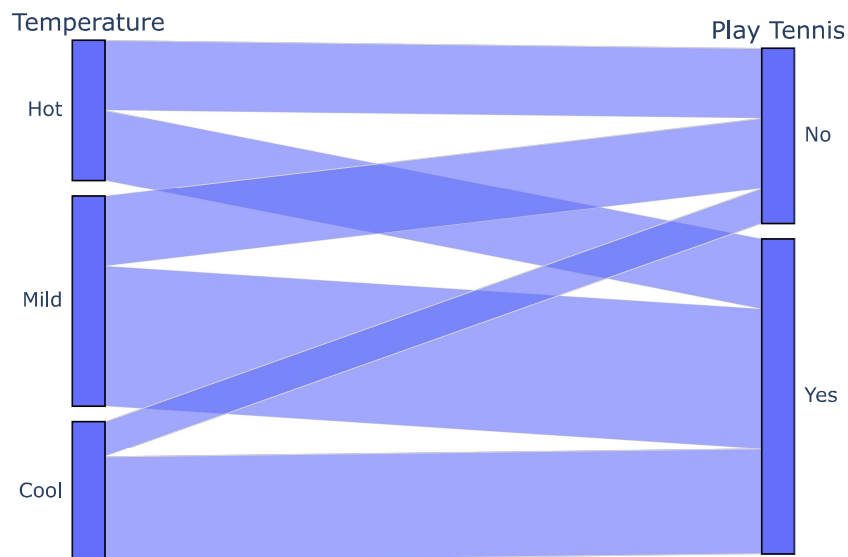
```
play_tennis_df=pd.read_csv('PlayTennis.csv')
play_tennis_df.head(5)
```

Out[2]:

	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

In [3]:

```
import plotly.express as px
fig = px.parallel_categories(play_tennis_df[['Temperature','Play Tennis']],
                           width=600,height=400)
fig.show()
```



In [4]:

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

In [5]:

```
number_encoder=LabelEncoder()
```

In [6]:

```
play_tennis_df['Outlook']=number_encoder.fit_transform(play_tennis_df['Outlook'])
play_tennis_df['Temperature']=number_encoder.fit_transform(play_tennis_df['Temperature'])
play_tennis_df['Humidity']=number_encoder.fit_transform(play_tennis_df['Humidity'])
play_tennis_df['Wind']=number_encoder.fit_transform(play_tennis_df['Wind'])
play_tennis_df['Play Tennis']=number_encoder.fit_transform(play_tennis_df['Play Tennis'])
```

In [7]:

```
play_tennis_df.head()
```

Out[7]:

	Outlook	Temperature	Humidity	Wind	Play Tennis
0	2	1	0	1	0
1	2	1	0	0	0
2	0	1	0	1	1
3	1	2	0	1	1
4	1	0	1	1	1

In [8]:

```
X=play_tennis_df.drop(columns='Play Tennis',axis=1)
X.head()
```

Out[8]:

	Outlook	Temperature	Humidity	Wind
0	2	1	0	1
1	2	1	0	0
2	0	1	0	1
3	1	2	0	1
4	1	0	1	1

In [9]:

```
y=play_tennis_df['Play Tennis']
y.head(3)
```

Out[9]:

```
0    0
1    0
2    1
Name: Play Tennis, dtype: int32
```

In [16]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.10, random_state=101)
```

In [11]:

```
from sklearn.naive_bayes import GaussianNB
model=GaussianNB()
```

In [12]:

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

Out[12]:

GaussianNB()

Xq = outlook = sunny, temprature = cool , Humidity = high, wind = strong yq = ?

In [13]:

```
predict=model.predict([[2, 0, 0, 0]])
predict
```

Out[13]:

array([0])

In [14]:

```
y_pred=model.predict(X_test)
y_pred
```

Out[14]:

array([1, 1])

In [15]:

```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)
```

Out[15]:

1.0