What is a Model in ML?

- A model is an object that we need to train using data and after training this model can be used to make predictions about similar type problems.
- exp:
  - if we have trained a model on even/odd data then this model can only be used to make prediction about even/odd of an unseen number(that was not present in training).

In [17]:

```
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
```

In [18]:

```
df=pd.read_csv("f:/dataset/classification/fruits.csv")
df
```

Out[18]:

	diameter	weight	FruitName
0	3.0	30	Banana
1	6.0	100	Apple
2	6.1	95	Apple
3	3.2	35	Banana
4	5.5	80	Apple
5	7.1	120	Banana
6	2.5	60	Banana
7	2.3	100	Banana
8	4.8	70	Apple
9	4.8	79	Apple
10	5.8	120	Apple
11	2.6	85	Banana
12	6.0	110	Apple
13	6.3	95	Apple
14	3.0	40	Banana
15	3.5	25	Banana
16	5.5	100	Apple
17	7.5	120	Apple
18	2.5	50	Banana
19	2.7	40	Banana
20	4.8	90	Apple
21	5.8	90	Apple

In [19]:

```
model=KNeighborsClassifier()
```

In [20]:

```
#model.fit(2d array_like,1d array_like) #generally we use numpy arrays
#model.fit(feature_matrix,target_vector)
```

In [21]:

```
X=df.iloc[:,:-1].values  #feature matrix
y=df.iloc[:,-1].values   #target vector
```

In [22]:

```
model.fit(X,y)
```

Out[22]:

```
▾ KNeighborsClassifier
KNeighborsClassifier()
```

In [23]:

```
#pred as_1d array=model.predict(2d array_like)
#pred as_1d array=model.predict(feature matrix of samples)
```

In [24]:

```
model.predict([[4.7,80],[2.1,40]])
```

Out[24]:

```
array(['Apple', 'Banana'], dtype=object)
```

In [25]:

```
model.predict([[3.0,30],[2.6,85],[4.8,70]])
```

Out[25]:

```
array(['Banana', 'Apple', 'Apple'], dtype=object)
```

In [26]:

```
model.predict(X)
```

Out[26]:

```
array(['Banana', 'Apple', 'Apple', 'Banana', 'Apple', 'Apple', 'Banana',
       'Apple', 'Apple', 'Apple', 'Apple', 'Apple', 'Apple', 'Apple',
       'Banana', 'Banana', 'Apple', 'Apple', 'Banana', 'Banana', 'Apple',
       'Apple'], dtype=object)
```

In [27]:

```
y
```

Out[27]:

```
array(['Banana', 'Apple', 'Apple', 'Banana', 'Apple', 'Banana', 'Banana',
       'Banana', 'Apple', 'Apple', 'Apple', 'Banana', 'Apple', 'Apple',
       'Banana', 'Banana', 'Apple', 'Apple', 'Banana', 'Banana', 'Apple',
       'Apple'], dtype=object)
```

In [28]:

```
19/22
```

Out[28]:

```
0.8636363636363636
```

In [29]:

```
from sklearn.metrics import accuracy_score
```

In [30]:

```
accuracy_score([1,2,3,2,1],[2,1,3,1,2])
```

Out[30]:

```
0.2
```

In [31]:

```
accuracy_score(['apple','mango','apple'],['apple','apple','apple'])
```

Out[31]:

```
0.6666666666666666
```

In [32]:

```
#accuracy_score(true_output,pred_output)
pred=model.predict(X)
accuracy_score(y,pred)
```

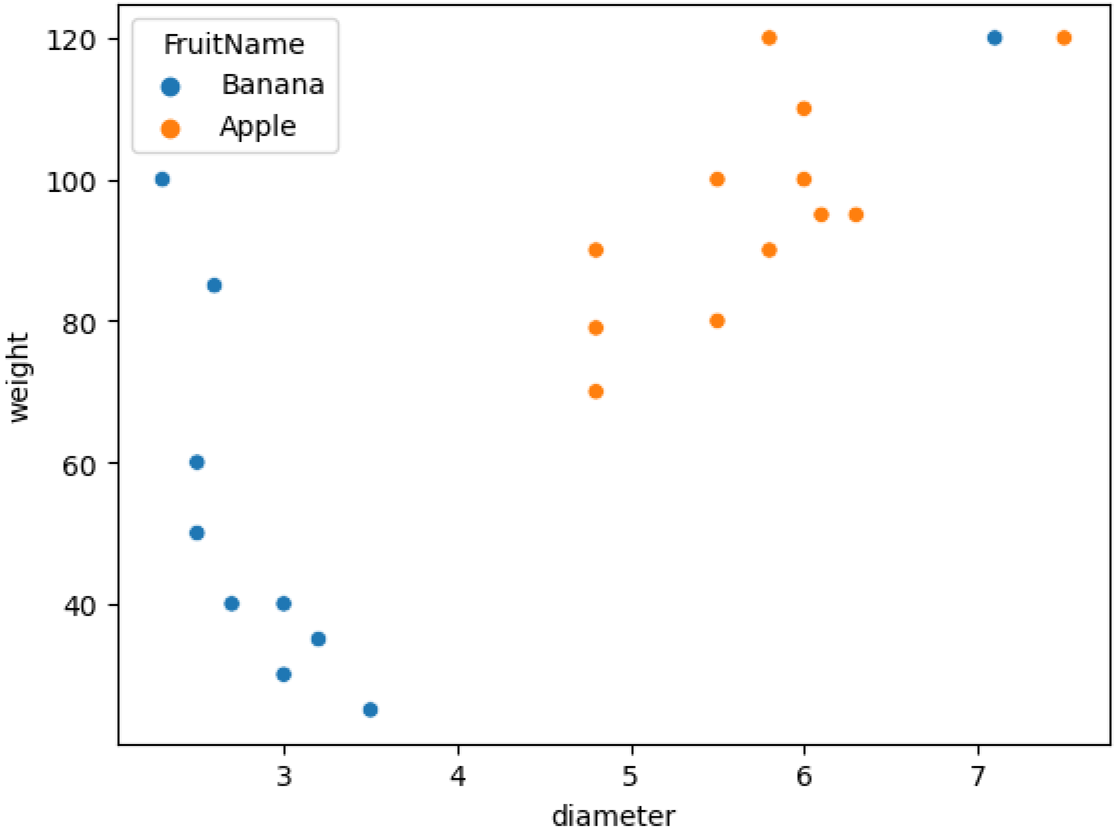
Out[32]:

```
0.8636363636363636
```

In [33]:

```
import matplotlib.pyplot as plt
import seaborn as sb
```

```
In [35]: sb.scatterplot(x=df.diameter,y=df.weight,hue=df.FruitName)
         plt.show()
```



```
In [41]: #Without removing outlier
         df=pd.read_csv("f:/dataset/classification/fruits.csv")
         X=df.iloc[:,:-1].values  #feature matrix
         y=df.iloc[:,-1].values   #target vector
         model.fit(X,y)
         pred=model.predict(X)
         accuracy_score(y,pred)

Out[41]: 0.8636363636363636
```

```
In [42]: #After removing outlier
         df=pd.read_csv("f:/dataset/classification/fruits.csv")
         df.drop(5,inplace=True)
         X=df.iloc[:,:-1].values  #feature matrix
         y=df.iloc[:,-1].values   #target vector
         model.fit(X,y)
         pred=model.predict(X)
         accuracy_score(y,pred)

Out[42]: 0.9047619047619048
```

```
In [ ]:
```