Machine Learning Process

- Loading data,
- Preprocessing,
- Training a model,
- Evaluating the model,
- Making predictions.

In [1]:
```
# To check if Pandas and Scikit-Learn is installed

import pandas as pd
import sklearn

print("pandas version:", pd.__version__)
print("scikit-learn version:", sklearn.__version__)
```

```
pandas version: 2.0.3
scikit-learn version: 1.3.0
```

In [2]:
```
#Import necessary libraries
import pandas as pd
from sklearn.linear_model import LogisticRegression
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score
```

In [3]:
```
#Loading the Dataset and Displaying the First Few Rows
iris_data = pd.read_csv('iris.csv')
iris_data.head()
```

Out[3]:

	Id	sepal_length	sepal_width	petal_length	petal_width	species
0	1	5.1	3.5	1.4	0.2	setosa
1	2	4.9	3.0	1.4	0.2	setosa
2	3	4.7	3.2	1.3	0.2	setosa
3	4	4.6	3.1	1.5	0.2	setosa
4	5	5.0	3.6	1.4	0.2	setosa

In [4]:
```
#Split the data into features (X) and labels (y)

X = iris_data.drop(columns=['Id' , 'species'])
y = iris_data['species']
```

In [5]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=
```

In [6]:
```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

In [7]:
```
#Create a ML model
model = LogisticRegression()
```

In [8]:
```
#train the model
model.fit(X_train_scaled, y_train)
```

Out[8]:
```
▾ LogisticRegression
LogisticRegression()
```

In [9]:
```
#Evaluate the model on the testing set
y_pred = model.predict(X_test_scaled)
accuracy = accuracy_score(y_test, y_pred)
print("Acuracy: ", accuracy * 100,"%")
```

```
Acuracy:  100.0 %
```

In [10]:
```
#Sample new data for prediction
new_data = np.array([[5.1,3.5,1.4,0.2],
                     [6.3,2.9,5.6,1.8],
                     [4.9,3.0,1.4,0.2]])
```

In [11]:
```
#Standardize the new data
new_data_scaled = scaler.transform(new_data)
```

```
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py:464: UserWarning: X does n
ot have valid feature names, but StandardScaler was fitted with feature names
  warnings.warn(
```

In [12]:
```
#make perdictions
predictions = model.predict(new_data_scaled)
```

In [13]:
```
print("predictions:",predictions)
```

```
predictions: ['setosa' 'virginica' 'setosa']
```