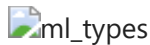


What is Machine Learning (ML) :

Machine learning is a powerful approach that enables computers to learn from data and make predictions or decisions without being explicitly programmed.

In []:

Types of Machine Learning



In classification, one major algorithm is missing, can you find it?

In []:

What is Classification?

Classification is the process of categorizing or grouping data into distinct classes or categories based on certain features or attributes.

Think of it as teaching a computer to recognize patterns, and then the computer learns to assign new or unseen data points to predefined categories.

In []:

Applications of Classification, why its important? :

Spam Detection: Automatically classifying emails as spam or not spam based on their content.

Image Recognition: Classifying images of objects, animals, or people in applications like self-driving cars or security systems.

Medical Diagnoses: Identifying whether a patient has a certain medical condition based on symptoms and test results.

In []:

Basic steps in Machine Learning coding:

Loading Dataset

Data Splitting

Vizualizing Dataset

Model Selection & Training

Model Evaluation

Prediction / Classification

```
In [1]: import matplotlib.pyplot as plt # for visualization
```

```
In [2]: def plot_img(img, label): # to plot an image with its ground truth as title
        plt.imshow(img.reshape(8, 8), cmap='gray')
        plt.title(label)
        plt.axis('off')
        plt.show()
```

See description of other sklearn toy datasets [here](#)

```
In [3]: # There are many built-in datasets in sklearn, eg - Iris, digits, breast_cancer etc
        # Importing 'load_digits' function from 'datasets' module of 'sklearn'.

        # Each datapoint is a 8 x 8 image of a digit.
        # Classes: 10 (0, 1, 2 .. 9)
        # Samples per class : 180
        # Total Samples : 1797

        from sklearn.datasets import load_digits
```

```
In [4]: # For training and testing a ML model we need to split the whole dataset into train
        # It is done using 'train_test_split' function from 'model_selection' module of 'sk
```

```
from sklearn.model_selection import train_test_split
```

```
In [5]: # Sklearn has different types of ML models in different modules, one such module is  
# Importing 'KNeighborsClassifier' for K-Nearest Neighbours  
  
from sklearn.neighbors import KNeighborsClassifier
```

Loading Dataset

```
In [6]: digits = load_digits()  
features = digits.data # Different pixel values  
targets = digits.target # Target values from 0 to 9
```

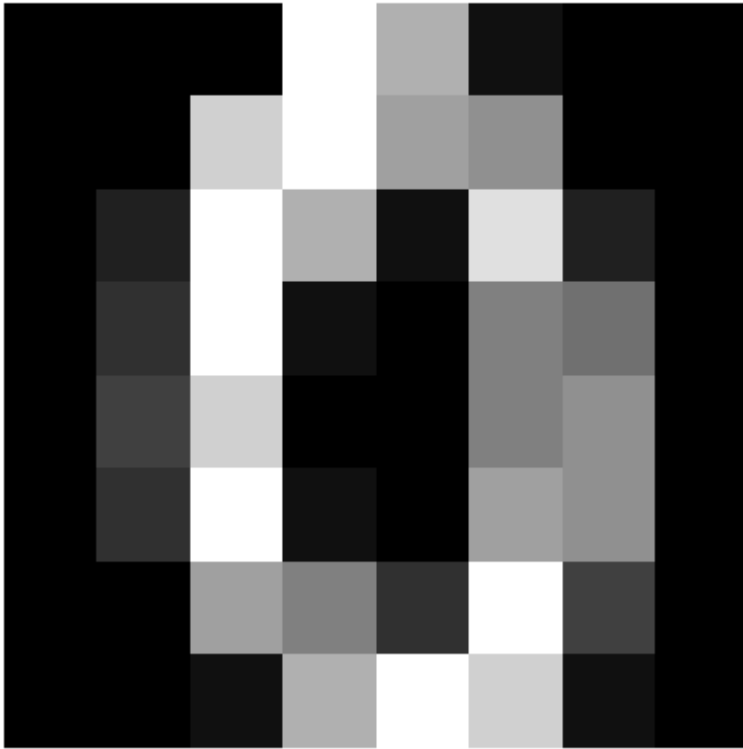
Data Splitting

```
In [7]: # Splitting the data into train and test sets  
# test_size is the amount of dataset that is allocated as testing set from the whole  
# 0.25 means 25% of the dataset is stored as testing set in X_test and their ground  
# random_state is a seed value for reproducibility (for giving consistent result even  
# if there's no random_state given, you will get different dataset partition, hence  
  
X_train, X_test, y_train, y_test = train_test_split(features, targets, test_size=0.25, random_state=42)
```

Visualizing Dataset

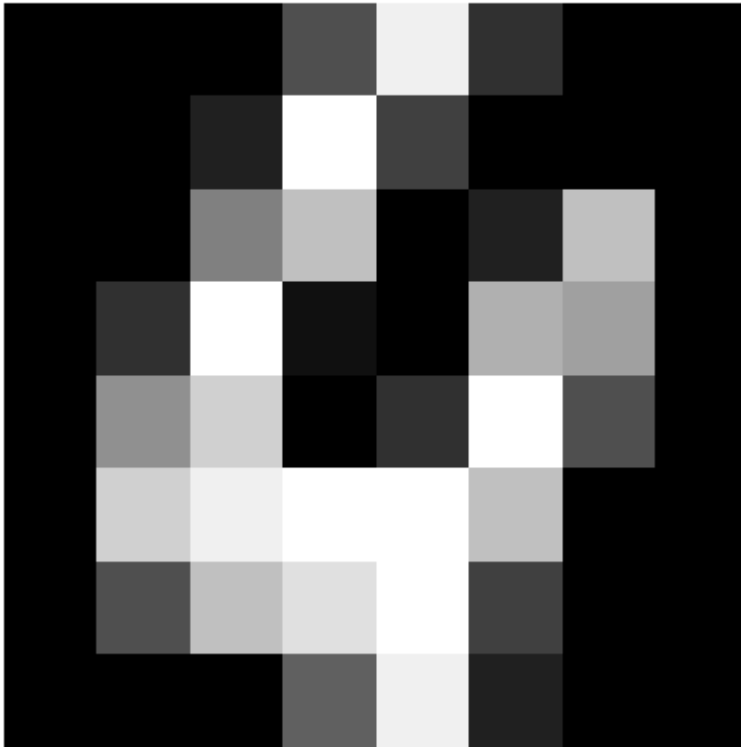
```
In [8]: # Visualizing training set  
  
i = 17  
plot_img(X_train[i], y_train[i])
```

0



```
In [9]: # Visualizing testing set  
i = 24  
plot_img(X_test[i], y_test[i])
```

4



Model Training

```
In [10]: # Train the classifier, use 'n_neighbors' argument to give the number of nearest ne

knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train) # fitting knn ml model to our training set
```

```
Out[10]: ▼      KNeighborsClassifier
KNeighborsClassifier(n_neighbors=3)
```

Model Evaluation

```
In [11]: acc = knn.score(X_test, y_test) # checking the accuracy of knn model on test set us
print(f"Accuracy: {round(acc * 100, 2)}%")
```

Accuracy: 97.78%

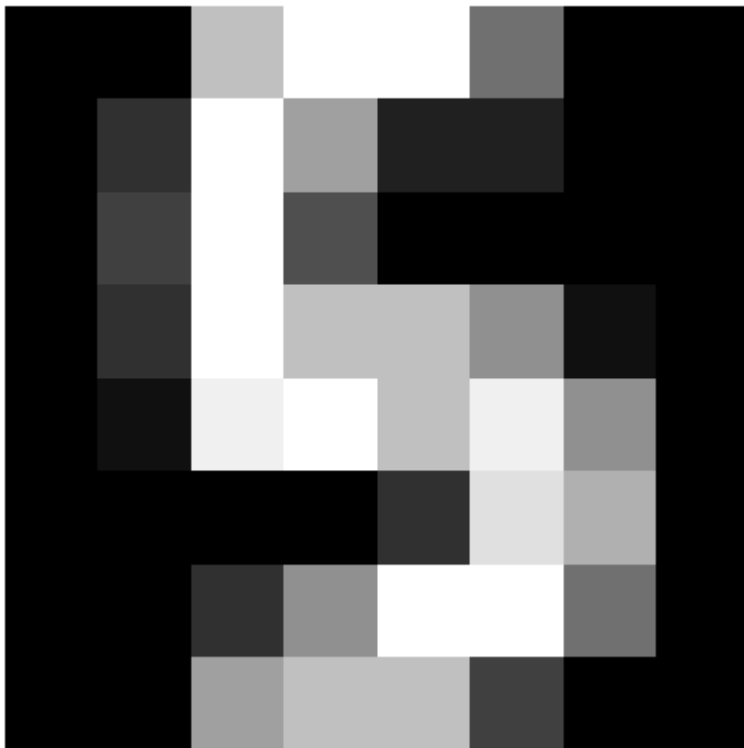
Classification

```
In [12]: j = 12
test_img = X_test[j].reshape(1, -1)
```

```
y_pred = knn.predict(test_img) # Making prediction using 'predict' function
print(f"Classified as {y_pred[0]}")
plot_img(X_test[j], y_test[j])
```

Classified as 5

5



In []: