What is Supervised Learning?

- Supervised learning is a type of machine learning where the algorithm is trained on a labeled dataset, which means that the input data used for training is paired with corresponding output labels. The goal of supervised learning is to learn a mapping or relationship between the input data and the output labels so that the algorithm can make accurate predictions or classifications on new, unseen data.

- In a supervised learning scenario, the algorithm is provided with a training set that includes input-output pairs. During the training process, the algorithm adjusts its internal parameters to minimize the difference between its predicted output and the actual output (ground truth) in the training data. This process involves iteratively adjusting the model based on the feedback provided by the labeled examples in the training set.

Comparison with unsupervised Learning:-

Supervised learning requires labeled training data, and the algorithm is trained to predict or classify based on known outcomes.

Unsupervised learning works with unlabeled data, aiming to uncover hidden patterns or relationships without specific guidance on what to look for.

Common Tasks:

Supervised: Classification and regression.

Unsupervised: Clustering and dimensionality reduction.

Applications:

Supervised learning is often used in scenarios where there's a clear target variable to predict.

Unsupervised learning is more exploratory and is used when the goal is to understand the inherent structure of the data.

What is Linear Regression in ML?

Linear Regression is a supervised learning algorithm used for predicting a continuous outcome variable (also called the dependent variable) based on one or more predictor variables (independent variables). It assumes a linear relationship between the input features and the target variable. In other words, it tries to fit a straight line to the data.

The general form of a simple linear regression model with one predictor variable is given by the equation:

$y=mx+b$

y is the dependent variable (the variable we want to predict).

x is the independent variable (the variable used to make predictions).

m is the slope of the line, representing the change in y for a unit change in x.

b is the y-intercept, representing the value of y when x is 0.

Code :-

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt

# Load your dataset (replace 'YourDataset.csv' with the actual file name)
csv_file_path = r"F:\RIDHESH DOC\Task-3\YourDataset.csv"
df = pd.read_csv(csv_file_path)

# Assuming you have a column 'X' as the independent variable and 'y' as the dependent variable
X = df[['X']]
y = df['y']

# 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)

# Create a linear regression model
model = LinearRegression()

# Fit the model on the training set
```

```
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Calculate and print the mean squared error
mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error: {mse}")

# Plot the regression line
plt.scatter(X_test, y_test, color='black', label='Actual data')
plt.plot(X_test, y_pred, color='blue', linewidth=3, label='Regression line')
plt.xlabel('Independent Variable (X)')
plt.ylabel('Dependent Variable (y)')
plt.title('Linear Regression')
plt.legend()
plt.show()
```