Logistic Regression in easy way:

Logistic regression is a type of regression analysis used to predict the probability of a binary outcome based on one or more predictor variables. It's commonly used in machine learning for binary classification problems.

Basic Concept:

In logistic regression, we're interested in modeling the probability that a given input belongs to a particular category.

Unlike linear regression, where the output is a continuous value, logistic regression models the output using the logistic function (also known as the sigmoid function).

Training:

During training, the logistic regression model learns the optimal values for the coefficients that best fit the training data.

This is typically done using optimization algorithms like gradient descent or Newton's method to minimize a cost function such as cross-entropy loss.

Prediction:

Once the model is trained, we can use it to make predictions on new data.

Given the input features of a new data point, we compute the linear combination and pass it through the sigmoid function to obtain the predicted probability.

If the predicted probability is above a certain threshold (usually 0.5), we classify the data point into one category. Otherwise, we classify it into the other category.

Example:

Suppose we're predicting whether a student will pass (1) or fail (0) an exam based on their study hours and attendance.

We collect data on study hours , attendance and exam outcomes (pass or fail).

We train a logistic regression model on this data to learn the relationship between study hours, attendance, and the probability of passing the exam.

Given a new student with known study hours and attendance, we use the trained model to predict the probability of passing the exam for that student.

Advantages:

Simple and easy to implement.

Outputs probabilities, making it interpretable.

Efficient for binary classification tasks with linear decision boundaries.

Limitations:

Assumes a linear relationship between the predictors and the log odds of the outcome.
Cannot capture complex nonlinear relationships between predictors and outcome.
Sensitive to outliers.

Logistic regression is a powerful and widely used technique in machine learning, especially for binary classification problems where interpretability is important. However, it's essential to understand its assumptions and limitations when applying it to real-world problems.

Program:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression

 Sample data: Study hours and attendance (features) and exam outcome (target)
study_hours = [2, 6, 3, 8, 5, 1, 7, 4, 9, 5]
attendance = [1, 1, 0, 1, 0, 0, 1, 1, 1, 0]
exam_outcome = [0, 1, 0, 1, 1, 0, 1, 1, 1, 0]

 Convert lists to numpy arrays
X = np.array([study_hours, attendance]).T
y = np.array(exam_outcome)

 Fit logistic regression model
model = LogisticRegression()
model.fit(X, y)

 Predict exam outcomes for new students
new_students = np.array([[6, 1], [3, 0], [7, 1]])
predicted_outcomes = model.predict(new_students)
```

P/F P/F P/F

```
 Print predicted outcomes
for i, outcome in enumerate(predicted_outcomes):
    print(f"Predicted outcome for student {i+1}: {'Pass' if outcome == 1 else 'Fail'}")

 Plot decision boundary
x_values = np.linspace(0, 10, 100)
```

→ 100 cut

```
y_values = -(model.coef_[0][0]  x_values + model.intercept_[0]) / model.coef_[0][1]
plt.plot(x_values, y_values, label='Decision Boundary')
plt.scatter(study_hours, attendance, c=exam_outcome, cmap='bwr', label='Data Points')
plt.xlabel('Study Hours')
plt.ylabel('Attendance')
plt.title('Logistic Regression: Exam Outcome Prediction')
```

```
plt.legend()
plt.grid(True)
plt.show()
```