Task #2 Giorgi GHIBRADZE

Logistic regression is a statistical method used to model the probability of a binary or categorical outcome variable (dependent variable) based on one or more independent variables (predictors). It is commonly used in classification problems where the goal is to predict whether an instance belongs to one of two (or more) classes.

The logistic regression model can be expressed mathematically as:

$$P(y = 1 | X) = 1 / (1 + e^{(-\beta 0 - \beta 1 * x1 - \beta 2 * x2 - ... - \beta n * xn)})$$

Where:

P(y = 1 | X) - is the probability that the dependent variable **y** is equal to 1 (or the positive class) given the values of the independent variables **X = (x1, x2, ..., xn)**.

β0 is the intercept term, and **β1**, **β2**, ..., **βn** are the regression coefficients, which represent the change in the log-odds of the dependent variable being 1 for a one-unit change in the corresponding independent variable.

e is the base of the natural logarithm (approximately 2.718).

The logistic regression model uses the logistic function (also known as the sigmoid function) to map the linear combination of the independent variables to a probability betIen 0 and 1. This probability can then be used to classify instances into the two (or more) classes.

Practical Example

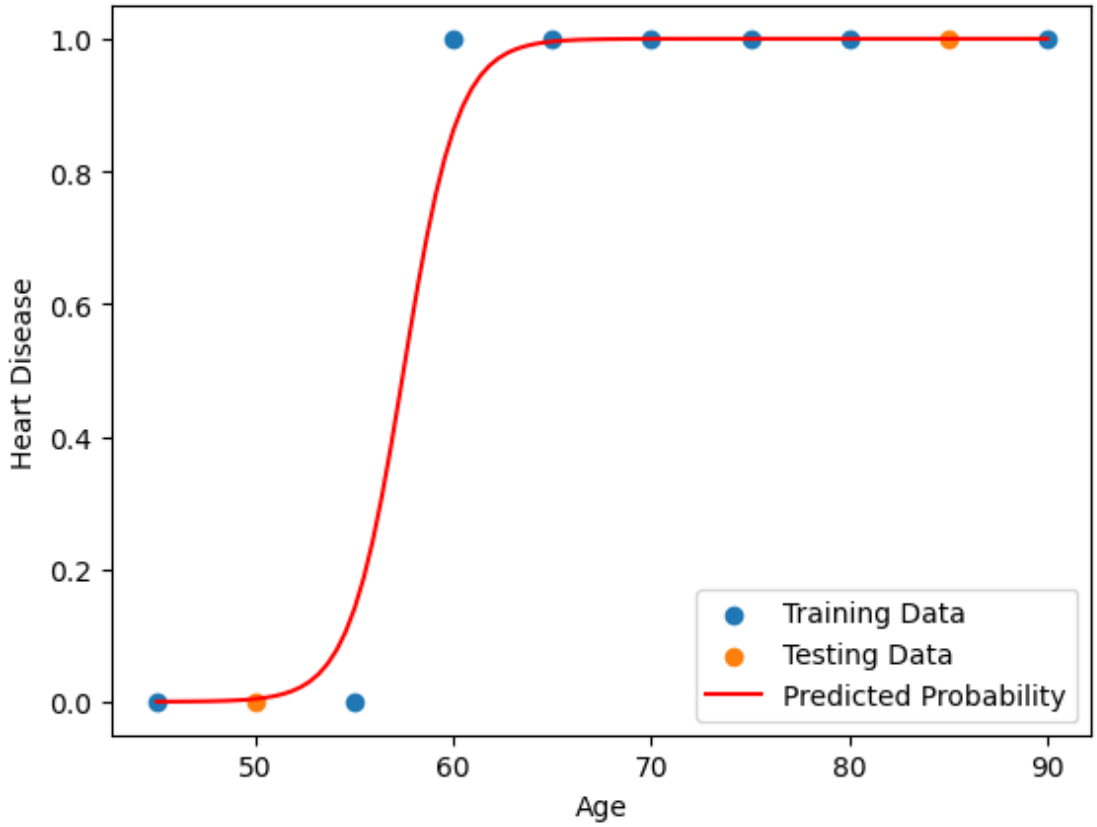
Let's consider a dataset that represents the relationship betIen age and whether a person has heart disease or not (binary outcome). I'll use the `sklearn` library in Python to build a logistic regression model.

```
import numpy as np

import pandas as pd

from sklearn.linear_model import LogisticRegression

from sklearn.model_selection import train_test_split

from sklearn.metrics import accuracy_score, confusion_matrix, classification_report


data = pd.DataFrame({

    'age': [45, 50, 55, 60, 65, 70, 75, 80, 85, 90],

    'heart_disease': [0, 0, 0, 1, 1, 1, 1, 1, 1, 1]
```

```
})

X_train, X_test, y_train, y_test = train_test_split(data['age'].values.reshape(-1, 1),
                                                    data['heart_disease'].values,
                                                    test_size=0.2,
                                                    random_state=42)

model = LogisticRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
cm = confusion_matrix(y_test, y_pred)
report = classification_report(y_test, y_pred)

print(f'Accuracy: {accuracy:.2f}')
print('Confusion Matrix:')
print(cm)
print('Classification Report:')
print(report)

import matplotlib.pyplot as plt
plt.scatter(X_train, y_train, label='Training Data')
plt.scatter(X_test, y_test, label='Testing Data')

x = np.linspace(min(data['age']), max(data['age']), 100)
y_pred_proba = model.predict_proba(x.reshape(-1, 1))[:, 1]
plt.plot(x, y_pred_proba, color='r', label='Predicted Probability')
```

```
plt.xlabel('Age')

plt.ylabel('Heart Disease')

plt.legend()

plt.show()
```

```
Accuracy: 1.00
Confusion Matrix:
[[1 0]
 [0 1]]
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         1
           1       1.00      1.00      1.00         1

    accuracy                           1.00         2
   macro avg       1.00      1.00      1.00         2
weighted avg       1.00      1.00      1.00         2
```



In this example, I have a dataset with two columns: `age` and `heart_disease` (0 for no heart disease, 1 for heart disease). I split the dataset into training and testing sets, create a logistic regression model using `sklearn.linear_model.LogisticRegression`, and fit the model to the training data.

I then evaluate the model's performance on the testing data using accuracy score, confusion matrix, and classification report. The accuracy score measures the overall correctness of the model's predictions, the confusion matrix shows the number of true positives, true negatives, false positives, and false negatives, and the classification report provides more detailed metrics such as precision, recall, and F1-score.

Finally, I plot the actual data points and the predicted probability of having heart disease as a function of age.

The output of the code will show the accuracy score, confusion matrix, and classification report, as well as a plot of the actual data and the predicted probabilities. The logistic regression model can then be used to make predictions on new, unseen data.

This is a simple example of a logistic regression model, but the concepts and techniques can be extended to more complex scenarios with multiple independent variables and multiple classes. The `sklearn` library provides a wide range of tools and methods for building and evaluating logistic regression models, as well as other machine learning algorithms.