Logistic Regression Model

Logistic regression is a statistical modeling technique used to model the relationship between a binary dependent variable (a variable that takes on only two possible values, usually 0 and 1) and one or more independent variables. The goal is to find the best-fitting equation that describes the probability of the dependent variable being 1 given the values of the independent variables. The equation takes the form:

$\ln(p / (1 - p)) = \beta_o + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p$

Where:

p is the probability of the dependent variable being 1

$x_1, x_2, ..., x_p$ are the independent variables

$\beta_o, \beta_1, ..., \beta_p$ are the coefficients that represent the change in the log odds of the dependent variable being 1 for a one unit change in the corresponding x

The coefficients are estimated using maximum likelihood estimation (MLE), which finds the values of the coefficients that maximize the likelihood of observing the data given the model.

Code:

```
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

# Load data
data = pd.DataFrame({
    'default': [1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1],
    'income': [25, 50, 40, 30, 80, 35, 60, 55, 45, 20, 70, 65, 28, 75, 33],
    'credit_score': [550, 700, 630, 590, 800, 610, 720, 690, 600, 530, 760,
730, 570, 790, 580]
})

# Extract variables
X = data[['income', 'credit_score']]
y = data['default']

# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
stratify=y, random_state=42)

# Fit logistic regression model
model = LogisticRegression()
model.fit(X_train, y_train)

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

# Evaluate model performance
print(f"Accuracy: {accuracy_score(y_test, y_pred):.3f}")
```

```
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
```

Accuracy: 1.000

Confusion Matrix:

[[3 0]

[0 2]]