Logistic Regression is a popular supervised learning algorithm used for binary classification tasks, where the target variable has two classes.

Here's a step-by-step guide to implementing Logistic Regression using Python and the popular machine learning library, scikit-learn:

Step 1: Import the required libraries

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report
```

Step 2: Prepare the dataset

```
# Assuming X and y are NumPy arrays or pandas DataFrames
X = ...  # Input features
y = ...  # Target variable
```

Step 3: 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)
```

Step 4: Create and train the Logistic Regression model

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

Step 5: Make predictions

```
y_pred = model.predict(X_test)
```

Step 6: Evaluate the model

```
accuracy = accuracy_score(y_test, y_pred)
classification_report = classification_report(y_test, y_pred)
```

Step 7: Print the evaluation results

```
print(f"Accuracy: {accuracy}")
print(f"Classification Report:\n{classification_report}")
```