Logistic Regression Study Guide

Introduction to Logistic Regression

Logistic regression is a statistical model used for binary classification problems.
It predicts the probability of an event occurring, such as whether a person has a job (1) or not (0), based on input features.

Step-by-Step Explanation

Step 1: Importing Required Libraries
```python
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report, roc_curve, roc_auc_score, f1_score, precision_score, recall_score
```
Why?
- `numpy` for numerical operations
- `matplotlib` and `seaborn` for visualization
- `sklearn.model_selection` for data splitting
- `sklearn.linear_model` for logistic regression
- `sklearn.metrics` for model evaluation

Step 2: Creating a Sample Dataset
```python
age = np.array([16, 24, 31, 29, 40, 50, 21, 33, 45, 39])
job = np.array([0, 1, 1, 1, 1, 1, 0, 1, 1, 1])
```

Step 3: Understanding shape and reshape
- `.shape` returns the dimensions of an array.

- `reshape(-1,1)` is used to convert a 1D array into a column vector for machine learning models.

Example:

```python
age = age.reshape(-1, 1)
print(age.shape)  # Output: (10, 1)
```

Step 4: Splitting Data into Training and Testing Sets

```python
X_train, X_test, y_train, y_test = train_test_split(age, job, test_size=0.2, random_state=42)
```

Step 5: Training the Logistic Regression Model

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

Step 6: Making Predictions

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

Step 7: Evaluating the Model

```python
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
print("Accuracy:", accuracy_score(y_test, y_pred))
```

Step 8: Plotting the ROC Curve & AUC Score

```python
fpr, tpr, _ = roc_curve(y_test, model.predict_proba(X_test)[:,1])
plt.plot(fpr, tpr, label="ROC Curve")
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC Curve")
```

```
plt.legend()
plt.show()

print("AUC Score:", roc_auc_score(y_test, model.predict_proba(X_test)[:,1]))
```