Data Analytics II - Logistic Regression (Practical 5)

Theory - Logistic Regression & Confusion Matrix

Logistic Regression:

- Used for binary classification problems (0/1, Yes/No).

- Predicts probability that input belongs to a class.

- Uses sigmoid function to map values between 0 and 1.

Confusion Matrix:

- Summarizes prediction results for classification.

- Contains True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN).

- Metrics:

  Accuracy = (TP + TN) / Total

  Error Rate = 1 - Accuracy

  Precision = TP / (TP + FP)

  Recall = TP / (TP + FN)

Logistic Regression - Code with Explanation

```
# Import required libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score,
recall_score

# Load dataset
df = pd.read_csv('Social_Network_Ads.csv')

# Encode categorical Gender column
df['Gender'].replace({'Female': 0, 'Male': 1}, inplace=True)

# Feature and Target separation
```

Data Analytics II - Logistic Regression (Practical 5)

```
x = df[['User ID', 'Gender', 'Age', 'EstimatedSalary']]
y = df['Purchased']

# Split dataset into train and test
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=42)

# Create and train Logistic Regression model
model = LogisticRegression()
model.fit(x_train, y_train)

# Make predictions on test data
y_predict = model.predict(x_test)

# Evaluate model
model.score(x_train, y_train)
model.score(x, y)
model.score(x_test, y_test)

# Confusion Matrix and metrics
cm = confusion_matrix(y_test, y_predict)
tn, fp, fn, tp = cm.ravel()

accuracy = accuracy_score(y_test, y_predict)
error_rate = 1 - accuracy
precision = precision_score(y_test, y_predict)
recall = recall_score(y_test, y_predict)
```