Task-1 Image Classification With Logistic Regression

Importing Libraries

In [14]:
```
import numpy as np
from tensorflow.keras.datasets import mnist
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
import joblib
import cv2
```

Normalizing Pixel Values

In [15]:
```
(X_train, y_train), (X_test, y_test) = mnist.load_data()

X_train_flattened = X_train.reshape(X_train.shape[0], -1)
X_test_flattened = X_test.reshape(X_test.shape[0], -1)

X_train_flattened = X_train_flattened / 255.0
X_test_flattened = X_test_flattened / 255.0
```

In [16]:
```
X_train_flat, X_val_flat, y_train_flat, y_val_flat = train_test_split(X_train_flattened, y_train, test_size=0.2, random
```

In [17]:
```
scaler = StandardScaler()
X_train_flat = scaler.fit_transform(X_train_flat)
X_val_flat = scaler.transform(X_val_flat)
X_test_flattened = scaler.transform(X_test_flattened)

logistic_regression_model = LogisticRegression(max_iter=1000, multi_class='multinomial', solver='lbfgs')
logistic_regression_model.fit(X_train_flat, y_train_flat)
```

Out[17]:
```
LogisticRegression(max_iter=1000, multi_class='multinomial')
```

Evaluation Of The Model

In [18]:
```
y_val_pred = logistic_regression_model.predict(X_val_flat)
y_test_pred = logistic_regression_model.predict(X_test_flattened)

val_accuracy = accuracy_score(y_val_flat, y_val_pred)
test_accuracy = accuracy_score(y_test, y_test_pred)

print(f"Validation Accuracy: {val_accuracy:.4f}")
print(f"Test Accuracy: {test_accuracy:.4f}")
```

```
Validation Accuracy: 0.9150
Test Accuracy: 0.9179
```

In [19]:
```
import joblib
joblib.dump(logistic_regression_model, 'logistic_regression_mnist_model.pkl')
joblib.dump(scaler, 'scaler_mnist.pkl')
```

Out[19]:
```
['scaler_mnist.pkl']
```