**Week 2 - Task 3: Unit Testing & Validation**

**User-Generated Prompt for AI (Example)**

*"I need a Python script that:*\*

* **Loads a dataset and processes it using data cleaning and transformation functions.**
* **Implements unit tests using Python’s unittest module to verify function correctness.**
* **Tests edge cases for missing data, incorrect data types, and out-of-range values.**
* **Ensures test coverage for all major functions in the script.** Can you generate a Python script that includes unit testing for these tasks?"\*

**AI-Generated Python Code for Unit Testing & Validation**

import pandas as pd

import unittest

# Function to load and clean data

def load\_and\_clean\_data(file\_path):

df = pd.read\_csv(file\_path)

df.dropna(inplace=True) # Remove missing values

df.columns = df.columns.str.strip().str.lower() # Standardize column names

return df

# Function to transform data (Example: Normalize a column)

def normalize\_column(df, column\_name):

if column\_name not in df.columns:

raise ValueError("Column not found in DataFrame")

df[column\_name] = (df[column\_name] - df[column\_name].min()) / (df[column\_name].max() - df[column\_name].min())

return df

# Unit testing class

class TestDataProcessing(unittest.TestCase):

def setUp(self):

"""Creates a sample dataset for testing."""

data = {

'temperature': [70, 85, None, 90, 75],

'humidity': [0.5, 0.6, 0.7, None, 0.8]

}

self.df = pd.DataFrame(data)

def test\_load\_and\_clean\_data(self):

"""Test that missing values are removed."""

df\_cleaned = load\_and\_clean\_data("test\_data.csv")

self.assertFalse(df\_cleaned.isnull().values.any())

def test\_normalize\_column(self):

"""Test normalization of a numeric column."""

df\_normalized = normalize\_column(self.df.copy(), 'temperature')

self.assertTrue((df\_normalized['temperature'] >= 0).all() and (df\_normalized['temperature'] <= 1).all())

def test\_column\_not\_found(self):

"""Test error handling when column is missing."""

with self.assertRaises(ValueError):

normalize\_column(self.df.copy(), 'non\_existent\_column')

# Run tests

if \_\_name\_\_ == '\_\_main\_\_':

unittest.main()