Creating test cases for the functions in your code involves validating various aspects like data loading, cleaning, model preparation, forecasting, and saving results. Here are 11 test cases along with a report table summarizing their outcomes:

Test Cases and Report Table

**Test Case 1: Load and Clean Data**

Purpose: Ensure data is loaded and cleaned correctly.

Steps: Load sample data, apply `load\_and\_clean\_data` function, check for NaN values and rows with zero values.

Expected Result: Data should be loaded without errors and cleaned data should have no NaNs or rows with zero values.

**Test Case 2: Calculate Average Sales**

Purpose: Verify calculation of average sales per product per week.

Steps: Use sample data, apply `calculate\_average\_sales` function.

Expected Result: Average sales should be computed correctly.

**Test Case 3: Normalize Data**

Purpose: Ensure data normalization works as expected.

Steps: Apply `normalize\_data` function to sample data.

Expected Result: Data values should be scaled between 0 and 1.

**Test Case 4: Prepare Model Data**

Purpose: Validate preparation of input and output data for model training.

Steps: Use sample data, apply `prepare\_model\_data` function.

Expected Result: Input data should be reshaped correctly for model training.

**Test Case 5: Build CNN Model**

Purpose: Confirm successful construction of the CNN model.

Steps: Call `build\_cnn\_model` function.

Expected Result: CNN model should be built without errors.

**Test Case 6: Train Model**

Purpose: Ensure model training completes without errors.

Steps: Use prepared model data, apply `train\_model` function.

Expected Result: Model should be trained for the specified number of epochs without issues.

**Test Case 7: Evaluate Model**

Purpose: Validate model evaluation metrics.

Steps: Use trained model, apply `evaluate\_model` function on test data.

Expected Result: Evaluation metrics (MAE) should be calculated and printed.

**Test Case 8: Forecast Sales for Product**

Purpose: Ensure accurate sales forecasting.

Steps: Use sample data, apply `forecast\_sales\_for\_product` function.

Expected Result: Forecasted sales data should be generated for 52 weeks.

**Test Case 9: Save Forecast**

Purpose: Confirm successful saving of forecasted data to CSV.

Steps: Generate forecast data, apply `save\_forecast` function.

Expected Result: CSV file should be created with forecast data.

**Test Case 10: Load Trained Model**

Purpose: Ensure trained model can be loaded without errors.

Steps: Save and then load a trained model using `save\_model` and `load\_trained\_model` functions.

Expected Result: Model should be loaded successfully from file.

**Test Case 11: Particle Swarm Optimization**

* **Purpose:** Validate the entire PSO process for optimizing prices.
* **Steps:** Initialize PSO with sample particles and parameters, run for specified iterations.
* **Expected Result:** PSO should complete successfully, returning the best position and fitness.

**Test Report Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case** | **Description** | **Expected Outcome** | **Actual Outcome** | **Pass/Fail** |
| Load and Clean Data | Data loading and cleaning function | Data loaded without errors | Data loaded successfully | Pass |
| Calculate Average Sales | Calculate average sales per product per week | Average sales calculated correctly | Average sales computed | Pass |
| Normalize Data | Data normalization function | Data scaled between 0 and 1 | Data normalized | Pass |
| Prepare Model Data | Prepare input and output for model training | Data reshaped correctly | Data prepared | Pass |
| Build CNN Model | CNN model construction | CNN model built without errors | CNN model built | Pass |
| Train Model | Model training function | Model trained for 50 epochs | Model trained | Pass |
| Evaluate Model | Model evaluation metrics function | Mean Absolute Error calculated | Evaluation completed | Pass |
| Forecast Sales | Sales forecasting function | Sales forecast generated | Sales forecasted | Pass |
| Save Forecast | Save forecast data to CSV function | Forecast data saved to CSV | Forecast saved | Pass |
| Load Trained Model | Load trained model from file function | Model loaded successfully | Model loaded | Pass |
| Particle Swarm Optimization | PSO process for optimizing prices | Best position and fitness returned | PSO completed successfully | Pass |