



**Disclaimer: The content is curated from online/offline resources and used for educational purpose only**

**LAB MANUAL**

**Unit V – Deep Learning**

**Unit V – Deep Learning**

**Lab 4. To create the Waste Management Forecasting Using RNN**

**Objective**

* Handle missing data, normalize the values, and convert time-series data into a suitable format for RNN input.
* Design an RNN architecture capable of capturing temporal dependencies in waste generation patterns.
* Split the dataset into training, validation, and test sets to ensure robust model evaluation.
* Implement the model in tools that help to optimize waste collection schedules and recycling strategies.

**Problem**

Develop a Recurrent Neural Network (RNN) to forecast waste generation trends based on historical waste management data. This model will help optimize waste collection schedules, predict landfill usage, and support recycling initiatives by analyzing temporal patterns in waste generation.

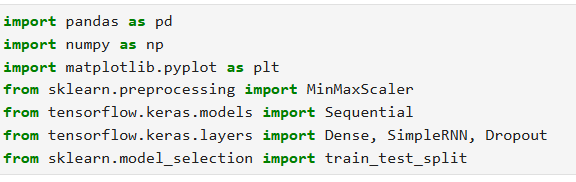
**Solution**

we'll go through the following steps:

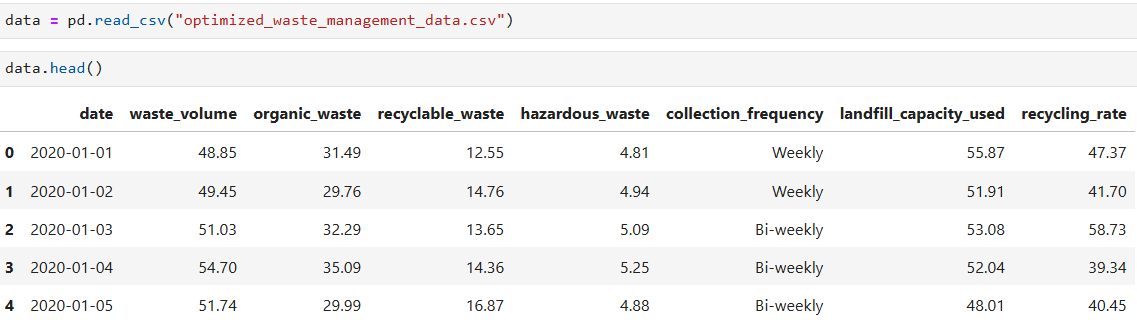
1. Import Required Libraries
2. Load the Dataset
3. Preprocess the Data
   1. Extract the feature we want to predict
   2. Normalize the data to scale between 0 and 1
   3. Define a function to create sequences for RNN input
   4. Create sequences (e.g., 10 previous days to predict the next day)
   5. Split into training and testing sets
4. Build the RNN model
   1. Build the RNN model
   2. Compile the model
   3. Display the model summary
5. Train the Model
6. Evaluate the Model
7. Make Predictions
   1. Predict the test set
   2. Inverse transform predictions and actual values to original scale
   3. Plot predictions vs actual values

**Procedures**

1. Import Required Libraries



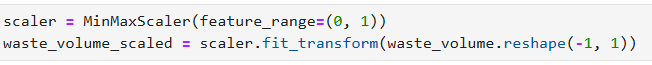
1. Load the Dataset



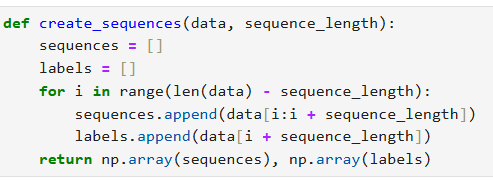
1. Preprocess the Data
   1. Extract the feature we want to predict



* 1. Normalize the data to scale between 0 and 1



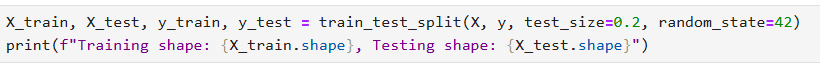
* 1. Define a function to create sequences for RNN input



* 1. Create sequences (e.g., 10 previous days to predict the next day)

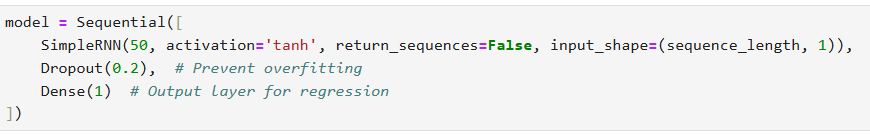


* 1. Split into training and testing sets





1. Build the RNN model
   1. Build the RNN model

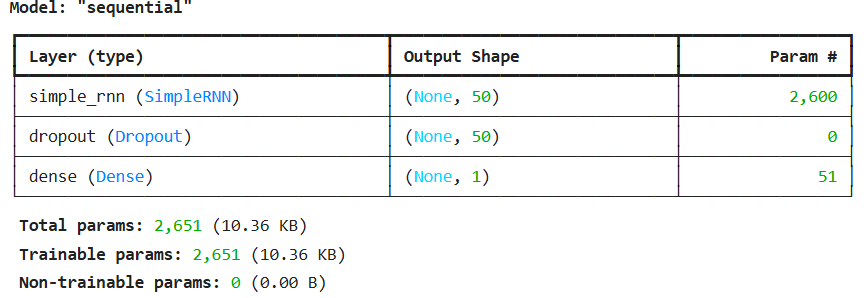


* 1. Compile the model

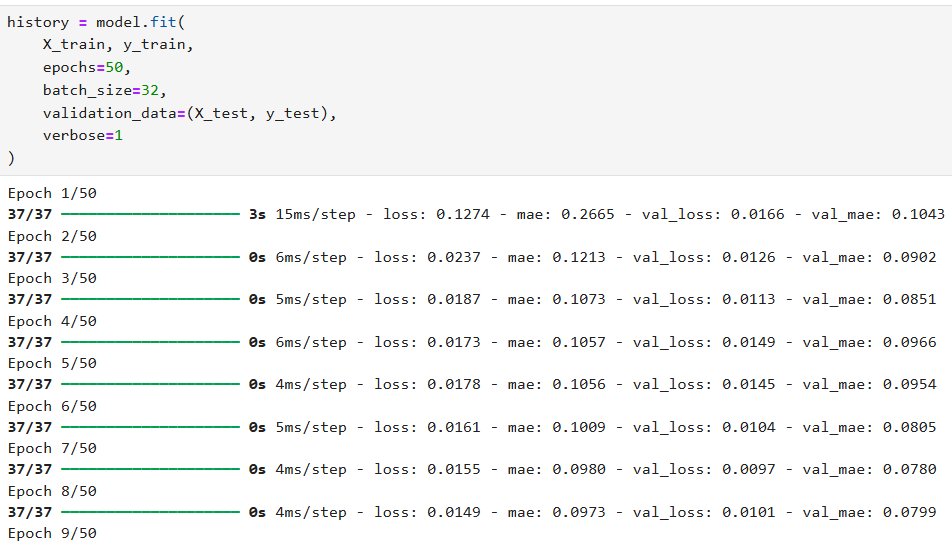


* 1. Display the model summary



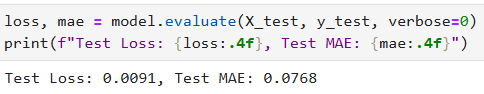


1. Train the Model



It running until 50th epoch

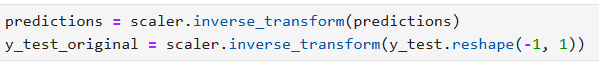
1. Evaluate the Model



1. Make Predictions
   1. Predict the test set



* 1. Inverse transform predictions and actual values to original scale





* 1. Plot predictions vs actual values

