Usage Document - Exploring optimal machine learning models for predicting crop yield at township level

Folder Structure:

Below mentioned are the important folders and the corresponding files contained within them:

1. **Data:**
    a. 'EDA_AAFC.ipynb' - Contains the code for data wrangling i.e., the steps performed to obtain the final dataset. It includes data exploration, data cleaning, data filtering, data transformation, and the final joining of all the four datasets.
    b. 'Model_Evaluation.xlsx' - Contains the evaluation metrics for all the six models corresponding to each Eco District
    c. 'Accuracy and MSE Plots.ipynb' – Contains Python visualizations of 'Count of Unique Townships' plotted against 'Accuracy' and 'Test MSE' respectively

2. **Models:**

The folder *'Models'* contains the following subfolders corresponding to the respective algorithms:

1. FF - *Feed Forward Neural Network*
2. LASSO - *Least Absolute Shrinkage and Selection Operator*
3. MLP - *Multi Layer Perceptron*
4. PCR - Principal Component Regression
5. PCA + RF - Principal component Analysis + Random Forest
6. RR - *Ridge Regression*

Each folder contains:

a. 'aafc_data.csv': The dataset obtained post data wrangling. This serves as the training data for our model.
b. 'scoring_test_df.csv': New data set (not the train/test data) on which prediction is to be done
c. 'test_script.py': The .py script which trains the respective model and performs the following functions:
    i. validation_metrics(): Displays Mean Squared Error Train, Mean Squared Error Test, Mean Absolute Error and Accuracy of the fitted model
    ii. predicted_train_dataset(): Displays the train dataset along with the predicted yield values
    iii. predicted_test_dataset(): Displays the test dataset along with the predicted yield values
    iv. feature_importance(): Displays the features used in the fitted model and the respective importance
    This function is present only in LASSO and Ridge Regression
    v. number_principal_components(): Number of principal components used for fitting the model
    This function is present only in PCR and PCA + RF
    vi. cumulative_explained_variance(): Displays what %of variance of the train data set is explained by the chosen principal components

vii. score(): Outputs a data frame containing predicted yield for a new data set - *'scoring_test_df.csv'*

d. A subfolder of the template 'folder/folder_aafc' which contains a .py script, which is the base script that contains code for each model and the above-described functions.
   i. PCR - *Models/PCR/pcr_aafc/PCR.py*
   ii. RR - *Models/RR/ridge_regression_aafc/RidgeRegression.py*
   iii. PCA_RF - *Models/PCA_RF/pca_rf_aafc/PCA_RF.py*
   iv. MLP - *Models/MLP/mlp_aafc/MLP.py*
   v. LASSO - *Models/LASSO/lasso_aafc/LASSO.py*
   vi. FF - *Models/FF/feed_forward_aafc/FeedForwardNN.py*
e. 'Outputs' folder that contains:
   i. 'train_predicted_df.csv' - Train dataset along with the predicted yield values
   ii. 'test_predicted_df.csv' – Test dataset along with the predicted yield values
   iii. 'new_data_predicted_df.csv' - Predicted yield for the scoring data set

Execution:

Below are the steps to be followed to run the Ridge Regression script (Similar steps are to be taken to run the rest of the 5 models):

1. **Checks:**
   a. Install necessary libraries: pandas, numpy, tensorflow, keras, sklearn, altair
   b. Navigate to Models/RR/ and check if the 'aafc_data.csv' file is the latest and final wrangled dataset – Replace it with a new training dataset if required
   c. Check if 'scoring_test_df.csv' is updated and is a new dataset for which you want to predict the yield
2. On terminal, navigate to Models/RR/ and run the command *'python test_script.py'*
3. Enter a valid Ecodistrict ID
   For example: '748'
4. It prints the validation metrics along with generating the following outputs in the 'Outputs' folder:
   a. train_predicted_df.csv
   b. test_predicted_df.csv
   c. new_data_predicted_df.csv