INFO 7390 Midterm

CONTENTS

CODE AND OUTPUT DELIVERABLES

PART 1:

A Python/R script that takes the 2 input files as inputs and generates 1 output file that incorporates joins with the weather data.

- It should work without any human intervention

- It will be tested with new data (which may not be from Finland). If Google's geocoding api supports it and wunderground has the historical data, the script should work.

- For erroneous inputs, (for example wrong address such as No 1234, Taj Mahal, China, the script should gracefully exit with appropriate error messages written to the log)

Location: Midterm-Team-7\Part1\Part1_Ingestion.R

PART 2:

PREDICTION

One script which takes the output from part 1 as input and generates 4 things

- *78 MODELS*

  - A new file with all input columns + 1 prediction column + 1 outlier tag column
    Location: Part2\SingleBuilding_Prediction\RandomForest_Outlier\RF_Residuals\

  - A table in csv format with MAPE, MAE, RMSE for each of the 78 models for each technique and the selected model for each of the 78 models
    Location: Part2\SingleBuilding_Prediction\

- *1 MODEL*

  - A new file with all input columns + 1 prediction column + 1 outlier tag column.

    Location: Part2\AllBuilding_Script&Result\OutlierAnalysis\knn_model_Outliers_All.csv

- A table in csv format with MAPE, MAE, RMSE for each of the 78 models for each technique and the selected model for each of the 78 models.

  Location: Part2\AllBuilding_Script&Result\finland_allBuilding_allWeather.csv

- Your script is automated and the decision should be on MAPE, MAE, RMSE alone. No human intervention should be needed.

  Location: Part2\AllBuilding_Script&Result\AllBuildings_AllModel.R

- If you need to pass parameters to your script, you should do it through a configuration file in JSON format. No hardcoding of parameters and users shouldn't be expected to edit the code to change parameters (For example, if you expect the user to change the number of layers in the neural network, it should come from an external configuration file).

CLASSIFICATION

One script which takes the output from part 1 as input and generates 4 things.

- 78 MODELS

  - A new file with all input columns + 1 prediction column + 1 outlier tag column.

    Location: Part2\AllBuilding_Script&Result\OutlierAnalysis\knn_model_Outliers_All.csv

  - A table in csv format with Confusion matrix for each of the 78 models for each technique and the selected model for each of the 78 models.

    Location: Part2\SingleBuilding_Classification\

- 1 MODEL

  - A new file with all input columns + 1 prediction column + 1 outlier tag column.
    Location: Part2\AllBuilding_Script&Result\OutlierAnalysis\knn_model_Outliers_All.csv
  - A table in csv format with Confusion matrix for each technique and the selected model.
    Location: Part2\AllBuilding_Script&Result\ClassifcationOutput\

  - Your script is automated and the decision should be on Confusion matrix alone. No human intervention should be needed.

    Location: Part2\AllBuilding_Script&Result\ClassifcationOutput\

  - If you need to pass parameters to your script, you should do it through a configuration file in JSON format. No hardcoding of parameters and users shouldn't be expected to edit the code to change parameters (For example, if you expect the user to change the number of layers in the neural network, it should come from an external configuration file).

CLUSTERING

- One script which takes the output from part 1 as input and generates an output file with 2 additional columns
  - Cluster tags for the K-means clustering
  - Cluster tags for Hierarchical clustering
    Location: Part3\Part3_Clustering.R

- If you need to pass parameters to your script, you should do it through a configuration file in JSON format. No hardcoding of parameters and users shouldn't be expected to edit the code to change parameters (For example, if you expect the user to change the number of clusters / distance measures it should be done through a configuration file).