**IMPLEMENTATION OVERVIEW**

1. **ENVIRONMENT**
2. **FOLDER CONTENT**
3. **EXPLANATION**
4. **WORKFLOW**

**DETAILS**

1. **ENVIRONMENT**

* CONFIGURATION
  + Python 3.6.5
  + Anaconda 5.2
  + IDE: Spyder
  + Window 10
* LIBRARIES
  + seaborn
  + matplotlib
  + pandas
  + numpy
  + keras
  + sklearn
  + git
  + functools
  + pylab
  + mpl\_toolkits
  + json
  + pymongo
  + pickle
  + collections
* DATABASE
  + MongoDB 3.6.5
* SOURCE CONTROL MANAGEMENT
  + GitLab 11.0.2

1. **FOLDER CONTENT**

GitLab repository

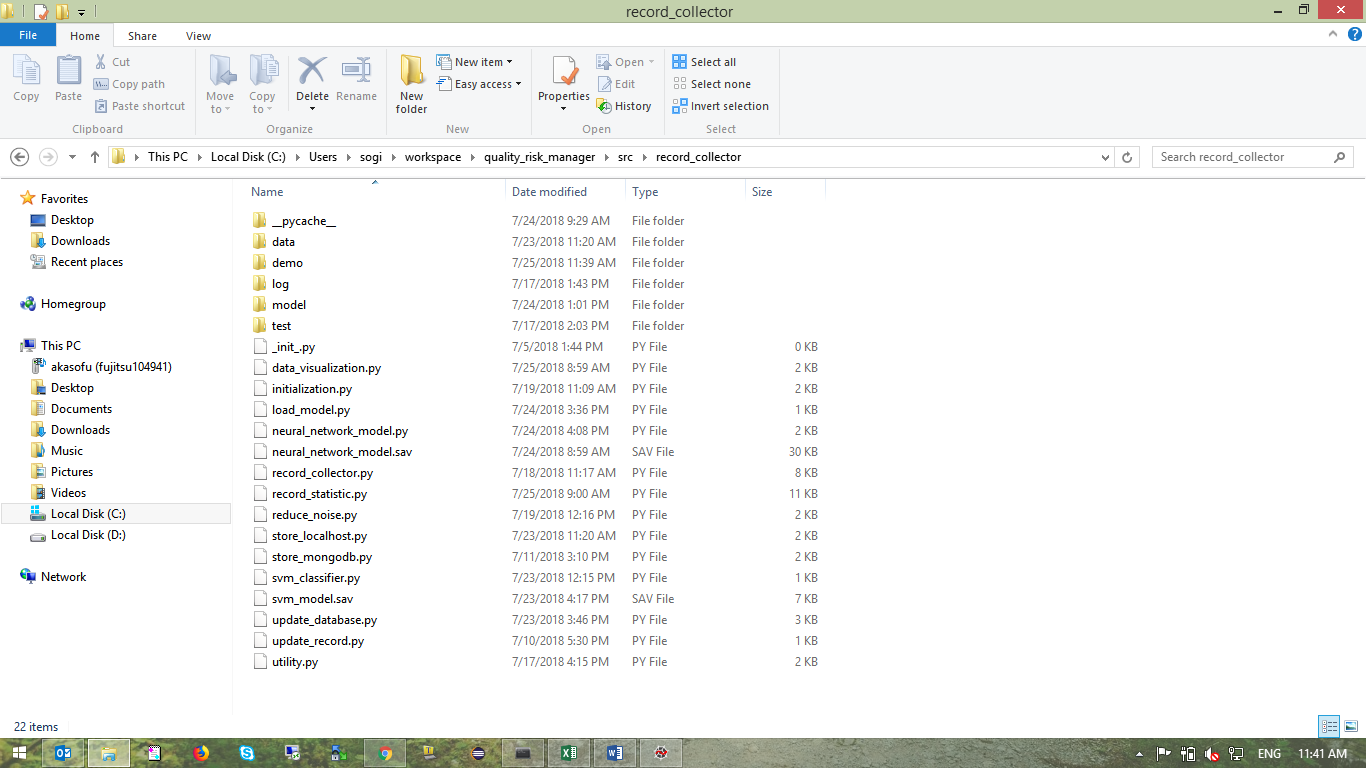
<http://10.130.110.31/develop_ai/quality_risk_manager/tree/bugfix_feat_record_collector/src/record_collector>

Branch: bugfix\_feat\_record\_collector

After cloning the repository, my local folder is

C:\Users\sogi\workspace\quality\_risk\_manager\src\record\_collector

Folder structure



|  |  |
| --- | --- |
| \_\_pycache\_\_ | Default cache of Python |
| data | Local database for training model |
| demo | Python files for demo and trial |
| log | Log files |
| model | Store learning models |
| test | Test functions |

1. **EXPLANATION**

|  |  |
| --- | --- |
| data\_visualization.py | Plot data visualization (bot plot, join plot) by Seaborn |
| initialization.py | Configuration of MongoDB location and local working directory |
| load\_model.py | Load model from file and evaluate model |
| neural\_network\_model.py | Get and split data into training set and test set Create neural network model Train and test model Evaluate and save model into SAV file |
| record\_collector.py | Connect to local GitHub repository Fetch all commits of ReactiveX/RxJava repository Divide commits into bug and non-bug Log the error and process Find required information of commit  Store into MongoDB |
| record\_statistic.py | Get and compare statistic information of data Plot data histogram 2D, 3D and scatter plot (data visualization)  Normalize data |
| reduce\_noise.py | Create query filter statement to reduce noise in data (use in demo hypothesis) |
| store\_localhost.py | Fetch data from MongoDB  Clean text data Save data into local CSV file |
| store\_mongodb.py | Connect to MongoDB server location Methods to insert, update, find data |
| svm\_classifier.py | Get and split data into training set and test set Create SVM model Train and test model Evaluate and save model into SAV file |
| update\_database.py | Update a local database CSV file into MongoDB |
| update\_record.py | Update one-by-one record into MongoDB |
| utility.py | Some utility functions (find number of insertion and deletion lines) |

1. **WORKFLOW**

