**part1:**

Crawler\_SummaryFiles.py: Download sample files, pre-process, and generate summary files.

PlotSummary.ipynb: Plot graphs based on summary files.

EDA.ipynb: EDA on quarterly data for 2007, 2008, and 2009.

/part2/classification/part1:

Classification-download-process.py: Download Q12005 and Q22005 data and pre-process.

**Part 2**

**Regression:**

Part1:

1. I am using Prediction-download-process.py this script to programmatically downloads Q12005 and Q22005 origination data and pre-processes it.
2. We implemented different algorithms of selection features including Forward, Backward, Stepwise and Exhaustive search methods.
3. We use RMS MAE and MAPE to evaluate model for both training and testing datasets, Based on these features selection algorithm. We get the best model.
4. We use Random Forest, Neural Network and linear regression to run those training and testing dataset. We find that the model from linear regression is the best one.

Part 2:

Financial crisis

Year 2007

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | [Q12007-Q22007](http://localhost:8888/tree/regression/part2/Financial_crisis/Q12007-Q22007) | [Q22007-Q32007](http://localhost:8888/tree/regression/part2/Financial_crisis/Q22007-Q32007) | [Q32007-Q42007](http://localhost:8888/tree/regression/part2/Financial_crisis/Q32007-Q42007) | [Q42007-Q12008](http://localhost:8888/tree/regression/part2/Financial_crisis/Q42007-Q12008) |
| RMS\_train | 0.10688309201 | 0.126880971738 | 0.122513824743 | 0.152964538776 |
| RMS\_test | 0.134330320461 | 0.248386103081 | 0.24715271411 | 0.380467807683 |
| MAE\_train | 0.24613616259 | 0.271616882853 | 0.263885362469 | 0.298263094825 |
| MAE\_test | 0.277144858888 | 0.413502977896 | 0.403162493959 | 0.513861534474 |
| MAPE\_train | 3.96899547935 | 4.31553355037 | 3.99715800431 | 4.71375033731 |
| MAPE\_test | 4.34662784467 | 6.07057398458 | 6.5722414547 | 9.17140731194 |

Year 2009

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | [Q12009-Q22009](http://localhost:8888/tree/regression/part2/Financial_crisis/2yearslater/Q12009-Q22009) | [Q22009-Q32009](http://localhost:8888/tree/regression/part2/Financial_crisis/2yearslater/Q22009-Q32009) | [Q32009-Q42009](http://localhost:8888/tree/regression/part2/Financial_crisis/2yearslater/Q32009-Q42009) | [Q42009-Q12010](http://localhost:8888/tree/regression/part2/Financial_crisis/2yearslater/Q42009-Q12010) |
| RMS\_train | 0.116152781721 | 0.116152781721 | 0.129393552851 | 0.105073403271 |
| RMS\_test | 0.111132835018 | 0.111132835018 | 0.14621141151 | 0.103227283832 |
| MAE\_train | 0.252468398783 | 0.252468398783 | 0.282460678066 | 0.255568966305 |
| MAE\_test | 0.256823041007 | 0.256823041007 | 0.303300994933 | 0.256225927924 |
| MAPE\_train | 5.05680971744 | 5.05680971744 | 5.55339786502 | 5.25238321958 |
| MAPE\_test | 5.34903482807 | 5.34903482807 | 6.3993219563 | 5.27882053976 |

Economic boom

Year 1999:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Q11999-Q21999 | Q21999-Q31999 | Q31999-Q41999 |
| RMS\_train | 0.100532035939 | 0.130278891723 | 0.170748208746 |
| RMS\_test | 0.181691662316 | 0.553041543207 | 0.829235976447 |
| MAE\_train | 0.226705834433 | 0.270791160283 | 0.297754254626 |
| MAE\_test | 0.317206971076 | 0.650263143364 | 0.860501737681 |
| MAPE\_train | 3.25585587009 | 3.77161542349 | 3.86172274038 |
| MAPE\_test | 4.3117063459 | 8.17884335375 | 12.6261559086 |

Year 2013:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Q12013-Q22013 | Q22013-Q32013 | Q32013-Q42013 |
| RMS\_train | 0.148429481785 | 0.16907523783 | 0.256752086326 |
| RMS\_test | 0.17615082327 | 0.632410724332 | 0.215068774879 |
| MAE\_train | 0.311309646826 | 0.328611334354 | 0.407445242418 |
| MAE\_test | 0.343117207649 | 0.684186099508 | 0.366523535643 |
| MAPE\_train | 9.54048372014 | 9.83466986656 | 10.4241465177 |
| MAPE\_test | 10.0221759662 | 15.7244728123 | 8.63537649173 |

**Classification:**

Part1:

1. Programmatically download Q12005 and Q22005:  
   We should Classification-download-process.py will use cookies to download all the files and do pre-processing
2. In [logistic\_regression.ipynb](https://github.com/XunPeng715/ADS_midterm/blob/master/part2/classification/part1/logistic_regression.ipynb) we build logistic regression model and forward stepwise and cross-validation to select features. We use accuracy to evaluate the model. And then we get confusion Matrix and ROC curve for training and testing datasets respectively
3. [Random\_forest.ipynb](http://localhost:8888/notebooks/classification/part1/Random_forest.ipynb) and [Neural\_network.ipynb](http://localhost:8888/notebooks/classification/part1/Neural_network.ipynb) is built based on Random forest and Neural network algorithm respectively.
4. Compare difference algorithms with AUC, accuracy and time complexity we decided to use logistic regression.
5. [logistic\_regression-multiProcessing.py](http://localhost:8888/edit/classification/part1/logistic_regression-multiProcessing.py) is used for above classifications and computing matrix, we utilize the python multiprocessing technic to process model in parallel. And

[matrix\_classification.csv](http://localhost:8888/view/classification/part1/matrix_classification.csv) is the output csv for matrix

Part2:

Evaluate the model and compute those metrics for Q12016

we are using TPR(True Positive Rate) and FPR(False Positive Rate)

TPR = TP / (TP + FN) and FPR = FP / (FP + TN)

TP = True Positive FN = Flase Negative

FP = Flase Positve TN = True Negative