Test run of data.csv

autoML.py defaults:

```
usage: autoML.py [-h] [-m MODEL_TYPE] [-i INPUT_FILE] [-file2 SECONDARY_FILE]
                 [-w SECONDARY_WEIGHTS] [-d DISTANCEFN] [-sprs SPARSITY]
                 [-r RADIUS] [-t MAX_TIME] [-n MAX_ITERATIONS] [-pca N_PCA]
                 [-e N_EXPERTS] [-s {none,manual,auto,manual_both,auto_both}]
                 [-f HIERARCHY_FOLDER]

optional arguments:
  -h, --help            show this help message and exit
  -m MODEL_TYPE, --model_type MODEL_TYPE
                        choose classification(default), regression, clustering
                        or outlier_detection
  -i INPUT_FILE, --input_file INPUT_FILE
                        primary input file to be analyzed (default=data.csv)
  -file2 SECONDARY_FILE, --secondary_file SECONDARY_FILE
                        optional secondary input file, triggers multi-dataset
                        analysis (default=None)
  -w SECONDARY_WEIGHTS, --secondary_weights SECONDARY_WEIGHTS
                        weights for features in secondary file, default=False
  -d DISTANCEFN, --distanceFn DISTANCEFN
                        choose L_1Norm(n), euclidean(n), L_infinityNorm(n),
                        distanceOnEarth(n), L_1Norm_cat(n), or
                        L_infinityNorm_cat(n), where n=1,2,3,... is the chosen
                        dimension for calculating distances. Default is
                        L_1Norm(1)
  -sprs SPARSITY, --sparsity SPARSITY
                        sparsity threshold for including records in secondary
                        input file
  -r RADIUS, --radius RADIUS
                        radius for cutoff of the distance function (default=1)
  -t MAX_TIME, --max_time MAX_TIME
                        maximum time in seconds for training all models. The
                        default value is 1440 seconds.
  -n MAX_ITERATIONS, --max_iterations MAX_ITERATIONS
                        max iterations for cross-validation of each individual
                        model fit. The default is 10 for clustering, 100 for
                        classification and 100 for regression.
  -pca N_PCA, --n_pca N_PCA
                        number of PCA components for outlier detection.
                        (default is 4)
  -e N_EXPERTS, --n_experts N_EXPERTS
                        number of experts for Ensemble scoring. (default is 5)
  -s {none,manual,auto,manual_both,auto_both}, --privatize_data {none,manual,aut
o,manual_both,auto_both}
                        choose none, manual, or auto for privatization of the
                        data using ARX. For manual, an ARX window will launch.
                        For privatization of primary and secondary datasets,
                        choose manual_both or auto_both. Default is manual.
```

```
-f HIERARCHY_FOLDER, --hierarchy_folder HIERARCHY_FOLDER
                    folder containing hierarchy files for sensitive data,
                    if provided by the user. Default is hierarchy
```

It currently handles four types of models: classification, regression, clustering and outlier detection. If the model type is classification or regression then the last column of the input data is assumed to be the dependent variable. Option to add a second dataset and a distance function: The distance function is used to assign elements of the second dataset to each row in the first dataset. A cutoff radius is used for the selection, with default initial value of 1. The -r option can be used to scale the distance function differently.

Output:

```
Python 3.5.2 |Anaconda 4.2.0 (64-bit)| (default, Jul 5 2016, 11:41:13) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.


In [12]: runfile('C:/Users/torres/Documents/GitHub/autoML-multiData/autoML.py',
wdir='C:/Users/torres/Documents/GitHub/autoML-multiData')
Reloaded modules: DeIdentify, DistanceFn, Model, FuseData, RandomizedSearchCluster, TimeSeries,
Experts, Data, Image, ClusterWrapper
Converting file to features
Dataset 'data.csv': (1310, 11)
        Column names: ['current_0', 'current_1', 'current_10', 'current_2', 'current_3',
'current_4', 'current_5', 'current_6', 'current_7', 'current_8', 'current_9']
        Target name: fault
        Target type: cat
        Target classes: ['F1' 'F2' 'F3' 'F4' 'F5' 'F6' 'F7' 'N']
        Target encoding: [0 1 2 3 4 5 6 7]
        Row 1: [-0.026      -2.24       0.         ...,  4.21776398 -2.88941615  0.        ] -> 7
        Row -1: [ -1.55555556e-03  -9.90000000e-01  0.00000000e+00 ...,    3.73608696e+00
  -2.74672464e+00   0.00000000e+00] -> 0
GradientBoost
Time to fit 3 instances of GradientBoost: 19.35s
AdaBoost
Time to fit 3 instances of AdaBoost: 1.85s
ExtraTrees
Time to fit 3 instances of ExtraTrees: 0.13s
LogisticRegression
Time to fit 3 instances of LogisticRegression: 0.72s
GaussianNB
Time to fit 3 instances of GaussianNB: 0.05s
DecisionTree
Time to fit 3 instances of DecisionTree: 0.06s
RandomForest
Time to fit 3 instances of RandomForest: 1.53s
SGD
Time to fit 3 instances of SGD: 0.11s
LDA
Time to fit 3 instances of LDA: 0.05s
KNeighbors
Time to fit 3 instances of KNeighbors: 0.07s
QDA
Time to fit 3 instances of QDA: 0.10s
Fitting GradientBoost (n_iterations=20, max_model_time=130s)
   Number of iterations: 20, Elapsed time: 114.22s
Fitting AdaBoost (n_iterations=100, max_model_time=130s)
   Number of iterations: 100, Elapsed time: 68.43s
Fitting ExtraTrees (n_iterations=100, max_model_time=130s)
```

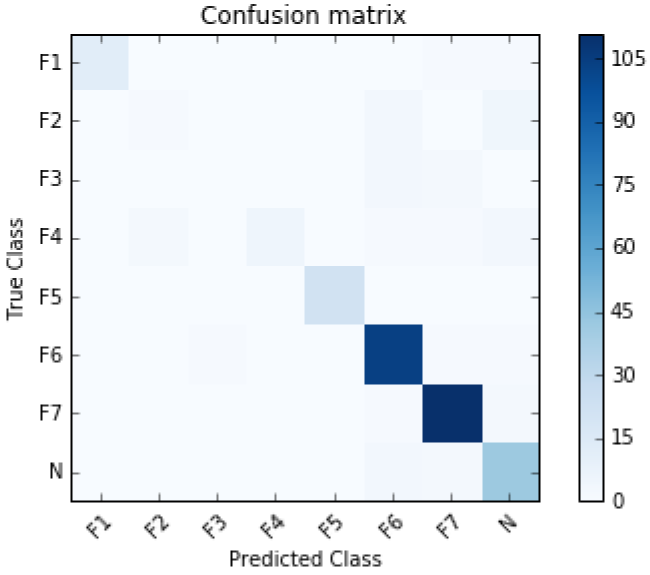
```
   Number of iterations: 100, Elapsed time: 3.91s
Fitting LogisticRegression (n_iterations=100, max_model_time=130s)
   Number of iterations: 100, Elapsed time: 24.86s
Fitting GaussianNB (n_iterations=100, max_model_time=130s)
   Number of iterations: 100, Elapsed time: 1.58s
Fitting DecisionTree (n_iterations=100, max_model_time=130s)
   Number of iterations: 100, Elapsed time: 2.09s
Fitting LDA (n_iterations=100, max_model_time=130s)
  Number of iterations: 100, Elapsed time: 1.78s
Fitting SGD (n_iterations=100, max_model_time=130s)
  Number of iterations: 100, Elapsed time: 4.71s
Fitting RandomForest (n_iterations=100, max_model_time=130s)
   Number of iterations: 100, Elapsed time: 61.04s
Fitting KNeighbors (n_iterations=100, max_model_time=130s)
   Number of iterations: 100, Elapsed time: 2.37s
Fitting QDA (n_iterations=100, max_model_time=130s)
   Number of iterations: 100, Elapsed time: 1.93s

        Number of models: 11
        Models: ['RandomForest: 0.902439', 'GradientBoost: 0.893293', 'ExtraTrees: 0.868902',
'DecisionTree: 0.865854', 'KNeighbors: 0.856707', 'LogisticRegression: 0.713415', 'GaussianNB:
0.707317', 'LDA: 0.704268', 'SGD: 0.682927', 'AdaBoost: 0.673780', 'QDA: 0.640244']
```

```
Ensemble Confusion Matrix (based on majority votes of top 5 models):
[[ 12   0   0 ...,   0   1   1]
 [  0   1   0 ...,   3   0   4]
 [  0   0   0 ...,   3   2   0]
 ..., 
 [  0   0   1 ..., 104   1   1]
 [  0   0   0 ...,   1 111   2]
 [  0   0   0 ...,   3   2  42]]
             precision    recall  f1-score   support

          0       1.00      0.86      0.92        14
          1       0.33      0.12      0.18         8
          2       0.00      0.00      0.00         5
          3       1.00      0.42      0.59        12
          4       1.00      1.00      1.00        21
          5       0.90      0.97      0.94       107
          6       0.94      0.97      0.96       114
          7       0.79      0.89      0.84        47

avg / total       0.89      0.90      0.89       328
```



In [13]: