Test run of test.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.

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 'test.data.csv': (4387, 254)
        Column names: ['acctype=A', 'acctype=B', 'acctype=C', 'acctype=D',
'acctype=E', 'acctype=F', 'acctype=G', 'acctype=H', 'contrib1=A', 'contrib1=E',
'contrib1=F', 'contrib1=G', 'contrib1=H', 'contrib1=I', 'contrib1=J', 'contrib1=K',
'contrib1=L', 'contrib1=M', 'contrib1=N', 'contrib1=O', 'drv_age', 'drv_inj=0',
'drv_inj=1', 'drv_inj=2', 'drv_inj=3', 'drv_inj=4', 'drv_sex=F', 'drv_sex=M',
'light=A', 'light=B', 'light=C', 'light=D', 'light=E', 'numvehs', 'object1=---',
'object1=1', 'object1=10', 'object1=11', 'object1=12', 'object1=13', 'object1=14',
'object1=15', 'object1=16', 'object1=17', 'object1=18', 'object1=19', 'object1=22',
'object1=23', 'object1=24', 'object1=26', 'object1=27', 'object1=28', 'object1=29',
'object1=3', 'object1=30', 'object1=40', 'object1=41', 'object1=42', 'object1=43',
'object1=44', 'object1=45', 'object1=46', 'object1=51', 'object1=6', 'object1=7',
'object1=98', 'object1=99', 'object1=V1', 'object1=V2', 'object1=V3', 'object1=V4',
'pop_group=1', 'pop_group=2', 'pop_group=3', 'pop_group=4', 'pop_group=5',
'pop_group=6', 'pop_group=7', 'pop_group=9', 'rdsurf=A', 'rdsurf=B', 'rdsurf=C',
'road_def1=A', 'road_def1=B', 'road_def1=C', 'road_def1=D', 'road_def1=E',
'road_def1=F', 'road_def1=G', 'road_def1=H', 'rodwycls=1', 'rodwycls=10',
'rodwycls=2', 'rodwycls=3', 'rodwycls=4', 'rodwycls=5', 'rodwycls=6', 'rodwycls=7',
'rodwycls=8', 'rodwycls=9', 'rodwycls=99', 'sobriety=<', 'sobriety=A', 'sobriety=B',
'sobriety=C', 'sobriety=D', 'sobriety=G', 'sobriety=H', 'vehtype=2', 'vehtype=A',
```

```
'vehtype=B', 'vehtype=C', 'vehtype=D', 'vehtype=E', 'vehtype=F', 'vehtype=G',
'vehtype=H', 'vehtype=I', 'vehtype=J', 'vehtype=K', 'vehtype=M', 'vehyr',
'weather1=A', 'weather1=B', 'weather1=C', 'weather1=D', 'weather1=E', 'weather1=F']
Target name: severity
Target type: cat
Target classes: ['0' '1' '2' '3' '4']
Target encoding: [0 1 2 3 4]
Row 1: [ 0. 0. 0. ..., 0. 0. 0.] -> 0
Row -1: [ 0. 0. 1. ..., 0. 0. 0.] -> 4
QDA
Time to fit 3 instances of QDA: 0.56s
KNeighbors
Time to fit 3 instances of KNeighbors: 1.99s
LogisticRegression
Time to fit 3 instances of LogisticRegression: 1.81s
GaussianNB
Time to fit 3 instances of GaussianNB: 0.36s
ExtraTrees
Time to fit 3 instances of ExtraTrees: 0.31s
SGD
Time to fit 3 instances of SGD: 0.59s
GradientBoost
Time to fit 3 instances of GradientBoost: 102.73s
AdaBoost
Time to fit 3 instances of AdaBoost: 12.74s
LDA
Time to fit 3 instances of LDA: 0.52s
RandomForest
Time to fit 3 instances of RandomForest: 2.97s
DecisionTree
Time to fit 3 instances of DecisionTree: 0.41s
Fitting AdaBoost (n_iterations=30, max_model_time=130s)
   Number of iterations: 30, Elapsed time: 131.61s
Fitting KNeighbors (n_iterations=100, max_model_time=130s)
   Number of iterations: 100, Elapsed time: 47.44s
Fitting DecisionTree (n_iterations=100, max_model_time=130s)
   Number of iterations: 100, Elapsed time: 22.52s
Fitting QDA (n_iterations=100, max_model_time=130s)
   Number of iterations: 100, Elapsed time: 16.00s
Fitting ExtraTrees (n_iterations=100, max_model_time=130s)
   Number of iterations: 100, Elapsed time: 8.07s
Fitting LogisticRegression (n_iterations=100, max_model_time=130s)
   Number of iterations: 100, Elapsed time: 406.49s
Fitting SGD (n_iterations=100, max_model_time=130s)
   Number of iterations: 100, Elapsed time: 16.46s
Fitting LDA (n_iterations=100, max_model_time=130s)
   Number of iterations: 100, Elapsed time: 13.97s
Fitting RandomForest (n_iterations=100, max_model_time=130s)
   Number of iterations: 100, Elapsed time: 70.99s
Fitting GaussianNB (n_iterations=100, max_model_time=130s)
   Number of iterations: 100, Elapsed time: 11.78s
Fitting GradientBoost (n_iterations=3, max_model_time=130s)
   Number of iterations: 3, Elapsed time: 578.86s

        Number of models: 11
```

```
Models: ['RandomForest: 0.810392', 'ExtraTrees: 0.807657', 'GradientBoost:
0.801276', 'LDA: 0.799453', 'LogisticRegression: 0.798541', 'GaussianNB: 0.795807',
'DecisionTree: 0.793072', 'QDA: 0.772106', 'SGD: 0.765725', 'KNeighbors: 0.724704',
'AdaBoost: 0.578851']

Ensemble Confusion Matrix (based on majority votes of top 5 models):
[[655   0   0   0   6]
 [  0   0   0   2   1]
 [  2   0  12   4   4]
 [ 44   1   0  69  22]
 [122   0   0   0 153]]
             precision    recall  f1-score   support

          0       0.80      0.99      0.88       661
          1       0.00      0.00      0.00         3
          2       1.00      0.55      0.71        22
          3       0.92      0.51      0.65       136
          4       0.82      0.56      0.66       275

avg / total       0.82      0.81      0.79      1097
```