AutoML includes the option of adding data from a second disjoint dataset into the analysis of the primary dataset. For the present purposes, disjoint means there are not common elements in the datasets that would enable a simple equality join into a common autoML input file, using the common elements as a common key.

Conceptually for the cases of interest, each row in the primary dataset is related to instances (e.g. rows) in the second dataset through a 'distance function' chosen by the autoML user. The distance could be a physical distance, or it could be differences in age, height, salary, or any other factors or combination of factors that the user believes to be relevant.

To incorporate auxiliary data from the second dataset, the user needs to create an intermediate file in the format that autoML expects. We call this file the 'secondary' file. (auotML does not currently have to capability to generate this file automatically.) The user also needs to choose a type of distance function from a pre-defined list. Currently, the list includes the following:

- L1Norm(n)
- Euclidean(n)
- LinfinityNorm(n)
- distanceOnEarth(n)
- L1Norm_cat(n) for categorical variables encoded as {0,1}
- LinfinityNorm_cat(n) for categorical variables encoded as {0,1}

The parameter n is the dimensionality of the vectors used to compute distances, specified by the user.

Primary dataset:

In autoML, the primary dataset is a csv file of the following form:

X_1	X_2	…	X_n	X_{n+1}	X_{n+2}	…	X_m	Y

For supervised learning, the last column ("Y") is for labeling each row. For unsupervised learning, the columns end at X_m. The first n values of X are used in the distance function calculation, along with the corresponding n-dimensional vector from the auxiliary data. This has to be taken into account when preparing the primary csv file. In detail, for $X_{primary}=[X_1, X_2, \ldots X_n]^T$ and $X_{auxiliary}$, distance is a function of ($X_{primary}$, $X_{auxiliary}$). As explained above, the user specifies n along with the type of distance function when running an autoML analysis, and then autoML figures out where to get $X_{primary}$ and $X_{auxiliary}$.

Secondary dataset:

The secondary dataset is an intermediate dataset produced from the auxiliary data. Since the auxiliary data can exist in many different forms, we currently require the user to create the secondary dataset

from the auxiliary data, rather than providing a way to generate it automatically. This is a weakness in that it adds to the effort required before autoML results can be generated, but is a necessary compromise due to project resources at this time.

For each row in the primary dataset, there is a corresponding row in the secondary dataset. Each row in the secondary dataset contains instances from the auxiliary data that are candidates for associating with the primary dataset row. In any particular analysis, a cutoff radius will be used to make the final choice.

We anticipate users will often perform analyses for multiple cutoff radii using the same primary and auxiliary data, to achieve desired insights. (In later work, we will look at using a probability vs. radius and using random number draws to broaden the ways candidates can be selected.) For this reason, the secondary csv file is designed to have a *superset of candidates* that span a large range of possible cutoff radii. This allows one secondary csv file to be used in analyses with different cutoff radii.

Row info	Candidate 1	Candidate 2						
	ID:lat:long:next:…							

The entry for each Candidate is a semicolon-delimited list of values of the following form:

<ID>:X_1^{aux}: X_2^{aux}: … : X_n^{aux}:<additional terms as desired>:<weight for candidate, optional> (1)

X^{aux} is the distance vector for the auxiliary data candidate, and <ID> represents a text ID for that feature. There can be as many additional semicolon-delimited terms after X_n^{aux} as desired. If weights are used (an option in autoML), then the last item is the weight that autoML will use for that element.

Data fusion:

An element in the secondary dataset is 'fused' into the primary dataset when the distance between X^{aux} and $X^{primary}$ for that element is less than the specified cutoff radius. When this condition holds, the column named FusedFeature_<ID> is incremented by one, where <ID> is the first element in expression (1) above. Optionally, when weights are being used in autoML, the column is incremented by the weight for that candidate, provided at the end of expression (1).

When the fusion operation is completed, the primary dataset will have been supplemented with additional columns for the fused features, one column for each FusedFeature_<ID>. In terms of the table above for the primary dataset, there will be new X_{m+1}, … columns.

Sparsity:

In some cases, a user might want to remove fused columns with low incidence counts from the analysis. This is not yet incorporated into autoML. Currently, sparsity is a parameter that can be set when running autoML, but nothing is done based on the entry.

Example:

Consider the example where the primary dataset contains information about accidents. Each accident has a latitude and longitude, and for this example assume those are in the first two columns in the primary data and comprise the location vector for the accident.

Consider an auxiliary dataset of landmarks, located by latitude and longitude, where the dataset can be queried for landmarks close to a specified (latitude, longitude).

Research question: can including information about nearby landmarks increase the accuracy of a classifier for accident severity?

To study this question, a secondary dataset is formed by querying the auxiliary dataset for every landmark with some maximum radius R_0 of each accident. For example, we could set R_0 equal to 5 km, reasoning that landmarks further away should not be a strong determining factor in accident severity. With this secondary dataset, analyses can be performed for any cutoff radius less than 5 km. In any one analysis, a subset of the secondary data will be fused into the primary data, and then a classifier will be trained and compared with the control case of no landmark information.