**Abstract**

Data is in the eye of the beholder, or at least this proves true for unsupervised learning. Through feature selection and transformation data alternate perspectives of data may provide useful insight into solving problems. Whether it be grouping similar observations via clustering or the more traditional supervised learning problems, unsupervised learning offers a new perspective on the data, one that aims to shed new light on familiar data. This paper examines two datasets in a variety of these contexts: via clustering, as well as data transformations to reduce dimensions by implementing PCA, ICA, Random Projection and subsequently clustering based on these projections. The strengths and weaknesses as well as insights provided by these steps are further examined.

# Datasets

Interesting analysis requires interesting problems, and in order illuminate the strengths, weaknesses and quirks of the examined supervised learning algorithms two well known datasets from the UCI machine learning repository data are examined.

## Madelon

### Instances: 5000 | Attributes: 440 | Data Types: Continuous (440) | Classes: 0, 1

The MADELON dataset is an artificial dataset created in 2003 for the NIPs conference as part of a feature selection challenge. The target class comes from a group of 32 clusters on the vertices of a five dimensional hypercube. Those points were randomly assigned a class (either 1 or -1). Additionally the five dimensions were transformed by linear combinations to form fifteen more features. To complicate the problem, 480 features of random noise were added to the dataset.

Of interest here is that the Madelon dataset presents a highly non-linear problem where the signal-to noise ratio is very low. 1% of the features are truly useful (the 5 dimensions) while 15 (3%) are superfluous albeit still informative. This leaves 96% as completely useless to learn from.

## Cars

### Instances: 1728 | Attributes: 6 (19 when one hot encoded) | Data Types: Categorical (6) | Classes: 0, 1, 2 for Cars with ratings of Unacceptable (0), Acceptable (1), Good/Very Good(2)

This dataset measures the hierarchical characteristics of cars to predict the overall quality/acceptability rating of the car. There are two overarching characteristics measured: price (buying and maintenance) and technical (number of doors, rider capacity, trunk size and safety rating) . Within those six attributes measured, each has multiple ordinal levels. The dataset was originally designed to showcase a simple hierarchical model for decision making. It serves as a test for a learner’s ability to recognize structure within the six dimensions. After processing the categorical values to a single dimension for each (e.g. rider capacity processed from one dimension with three levels into three dimensions with one level, each designating a separate condition (capacity-2, capacity4, capacity5+), the dataset expands to 1728 instances with 19 dimensions. A learner will have to distinguish which of the dimensions are related (e.g. capacity-2, capacity-4, capacity5 are not independent) and model them accordingly. Additionally, the classification problem is imbalanced with the vast majority of cars classified as unacceptable (70%), while 22% are acceptable and the final 8% classified as good/very good. The distribution of each category is balanced between levels (e.g. capacity-2, capacity-4, capacity5 each have 1/3 of the total instances in each level).

***NN Baseline Results***

In previous analysis, an artificial neural network was trained on each dataset. The optimal results are shown below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dataset** | **5 Fold CV Score** | **Activation Function** | **Learning Rate** | **Hidden Layer Size** |
| Cars | .5159 | Relu | .01 | (18) |
| Madelon | .7628 | Relu | .0001 | (62, 62, 62) |

These show very different architectures of neural networks. Cars is a small (single hidden layer) NN with mediocre performance while Madelon is a deep NN (3 hidden layers, each with 62 neurons) and a very small learning rate. This is an indication of the types of datasets for each, small and high signal for Cars and larger with complex non-linear relationships for Madelon. Also of note here is that the relu activation may have performed better than the logistic activation due to its sparsity in subsequent layers, a form of feature selection.

# Algorithms & Methodology

## Clustering Algorithms:

## K-Means

## Expectation Maximization (Gaussian Mixture Model)

## Unsupervised Learning / Dimension Reduction / Alternate Data Representation Algorithms:

## Principal Component Analysis (PCA)

## Independent Component Analysis (ICA)

## Random Projection (RP)

## Random Forest Feature Selection (RF)

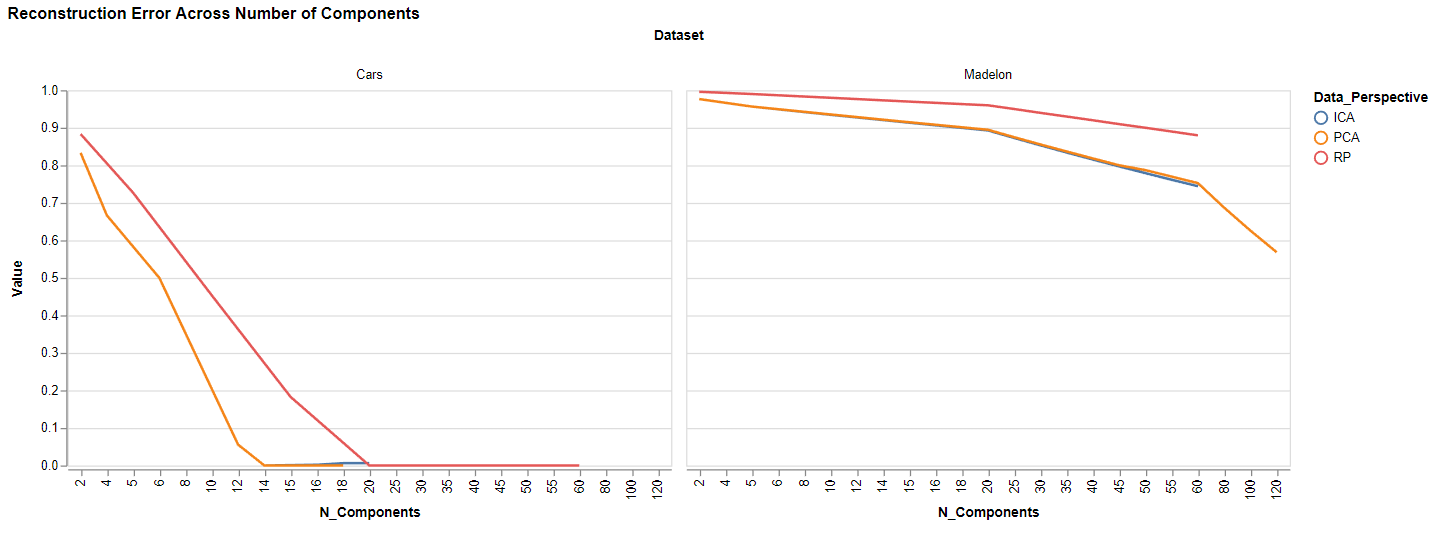
All clustering and projection algorithms were implemented via the python machine learning package sci-kit learn. For each feature transformation algorithm, both datasets were transformed across a varying number of dimensions. For each dimension size, two metrics were computed: one a quality of component metric and the other the reconstruction error measured by projecting the components back to the original dimensions and measuring squared pairwise distance. A 5 CV grid search across varying learning rates, hidden layer sizes, and projection components used was performed on each datasets. Based on subjective selection for the ‘best’ number of components, each projection was selected to be used for creating clusters. Both the original (BASE in the code) data and projected data was used for clustering the data. Clusters across varying number of cluster sizes were computed using K-Means and Expectation Maximization (implemented as a Gaussian Mixture Model). For each cluster method were measured using three metrics: a cluster quality metric (SSE for K-Means and likelihood for EM), mutual information with the target class and accuracy when used for classification. Clustering predictions were found by taking the cluster label and assigning as a prediction to all points in that cluster the majority target class label. The clusters were used as features in a 5 CV grid search across varying learning rates, hidden layer sizes, and projection components used was performed on each datasets. Finally, the data was reduced to 2 dimensions for visualization purposes using T-SNE. To contrast the feature transformation methods of PCA, ICA and RP a feature *selection* method was implemented by fitting a random forest classifier to the data and selecting the top X number of features based on their importance. This was further used in similar grid searches and clustering.

# Results

## Data ‘Perspective’ (Projections/Transformations/Selection, etc)

*Information Retention*

Three methods examined commonly fall under the category of ‘feature transformation’ or unsupervised learning. The goal being to represent the data in a new manner (i.e. gain perspective) while also retaining the maximum amount of information. Three such methods are further examined: PCA, ICA and RP. But how to determine the main parameter of these methods, mainly how many of components the data will be projected on should be kept? First, an examination of their respective ability to retain the maximum amount of information. This can be achieved by taking the projections, reconstructing to original dimensions and taking the distance as a measure of error:



## In both datasets PCA and ICA are closely entwined and do a similar job of reconstructing the data. It’s clear from the above graph that while the Cars dataset could be reconstructed nearly perfectly after a certain number of components, Madelon could not. This is due to the limitations of the algorithms and the nature of the Madelon dataset, namely that PCA/ICA/RP are all linear transformations whereas Madelon is highly nonlinear. Despite this, as the number of components increase the reconstruction error does decline. It’s interesting that PCA/ICA reconstructs the data to a nearly identical level since they are decomposing based on different criteria (variance vs independence). In the case of the Cars data this may indicate that the components are similar as I could see ICA decomposing each component into groups of categoric feature dimensions and PCA explaining variance in the same manner since those features account for most of the variance.

Of note here is the number of components at which point reconstruction error appears to level off somewhat. Picking an optimal parameter value over an increasing interval is often referred to as an ‘elbow’ method, and this case is no different. All methods have ‘elbow’ kinks which make for component number to use for projection of the data. In cases where there is a tradeoff between increase in performance and increase in parameter size, the bias is towards smaller values (i.e. simpler representations with fewer components).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Dataset*** | ***Method*** | ***Number of Components*** | | ***Dataset*** | ***Method*** | ***Number of Components*** |
| Cars | PCA | 14 |  | Madelon | PCA | 20 |
| Cars | ICA | 14 |  | Madelon | ICA | 20 |
| Cars | RP | 20 |  | Madelon | RP | 20 |

*Component Quality (Information Representation)*

For each method of data projection, a different metric is required to gauge the quality of components. PCA is the simplest and most interpretable. Each component is an eigenvector whose eigenvalue directly relates to the amount of variance explained in the data. The more variance explained, the better. The tradeoff is in the cumulative variance explained among the components. If every single dimension is kept, there is no dimension reduction and the 500th component will explain far less variance than the 1st component. For ICA, we measure the average absolute kurtosis of the components. With ICA we seek to decompose the data into independent components, and for this goal we attempt to maximize kurtosis. The idea behind this being that a normal distribution is the sum of independent observations and a high kurtosis indicates a highly non-Gaussian distribution, and if the data is made up of highly non-Gaussian components they will be independent of each other. For RP, the correlation between the pairwise distances of the original data and with the projected data is computed. This was repeated 10 times to reduce the variance from the ‘random’ aspect of random projection. If there is a distinct (linear) pattern in the projection it would have a larger correlation. The results graphed:

## 

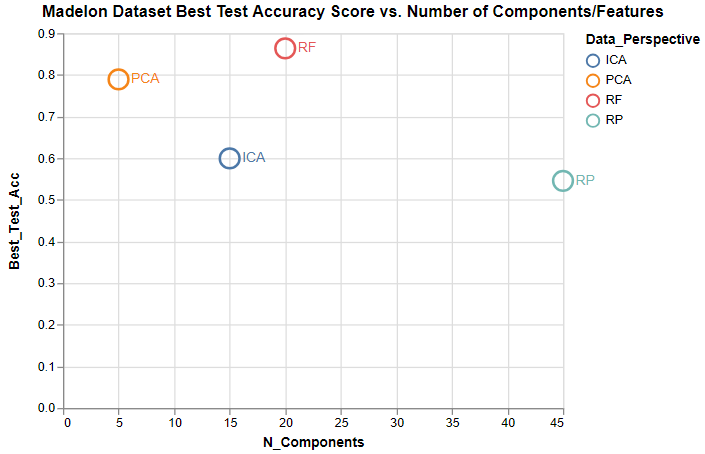
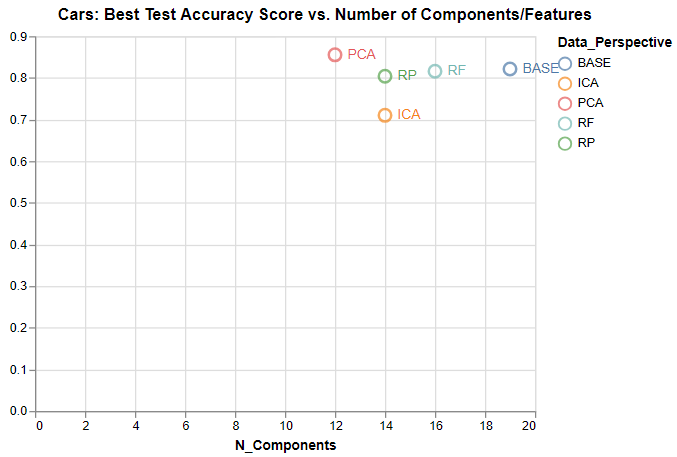
## The number of components varies with the dataset, as Cars (19 dimensions) has far fewer dimensions than Madelon (500). For ICA, the first thing that stands out is the bimodal peaks of the kurtosis along the number of components. It spikes at 9 before declining and then quickly rising to a maximum at 12 components. I believe the short spike at 9 indicates that ICA may have separated out each component to represent one of the nine categoric features in the data. For Madelon a similar occurrence appears to take place as the kurtosis spikes at 5 components and then levels off before eventually reaching the kurtosis reached with 5 components. This is likely due to ICA finding the 5 true features in the data. ICA is robust to the 15 semi-informative features that are linear combinations of the 5 true features, otherwise there would be an increasing kurtosis up to 20 components.

PCA paints a similar picture, however these results are even more interpretable. Both datasets see a sharp drop off in explained variance of components after a subset of the data. Cars at 8 components, or roughly ½ the size of the original dimensions while most of the variance in the Madelon set is captured by the first 5 components, reducing the dimensions down 99%. The Madelon data showcases the strength of PCA and the beauty of retaining a large portion of information while reducing data 99%.

Random Projection is more difficult to interpret due to its stochastic nature. We can interpret the results, but not the causal reasons behind it. To reduce the variance of results, random projection was implemented 10 times with the average correlation computed along with its standard deviation (not shown but available under the ‘results’ file). In any case, as the data is projected into more and more components it captures some type of linear relationship between the projected data and the original data with increasing correlation. The Cars data had a significantly higher correlation than the nonlinear Madelon set. However, the fact that Madelon is highly nonlinear but the correlation increases with number of components in a linear fashion make me skeptical of the effectiveness of random projection. At a certain point the correlations are likely spurious with a large number of dimensions capturing just as much noise as signal. By comparing with the reconstruction error of the data from previously, the Cars set has little reconstruction error at 20 components while Madelon has points with different levels of error (1, 20, 80). In favor of a simpler representation, 20 components are chosen as optimal for random projection. From these graphs, the selected number of components would be:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Dataset*** | ***Method*** | ***Number of Components*** | | ***Dataset*** | ***Method*** | ***Number of Components*** |
| Cars | PCA | 8 |  | Madelon | PCA | 5 |
| Cars | ICA | 5 |  | Madelon | ICA | 5 |
| Cars | RP | 20 |  | Madelon | RP | 20 |

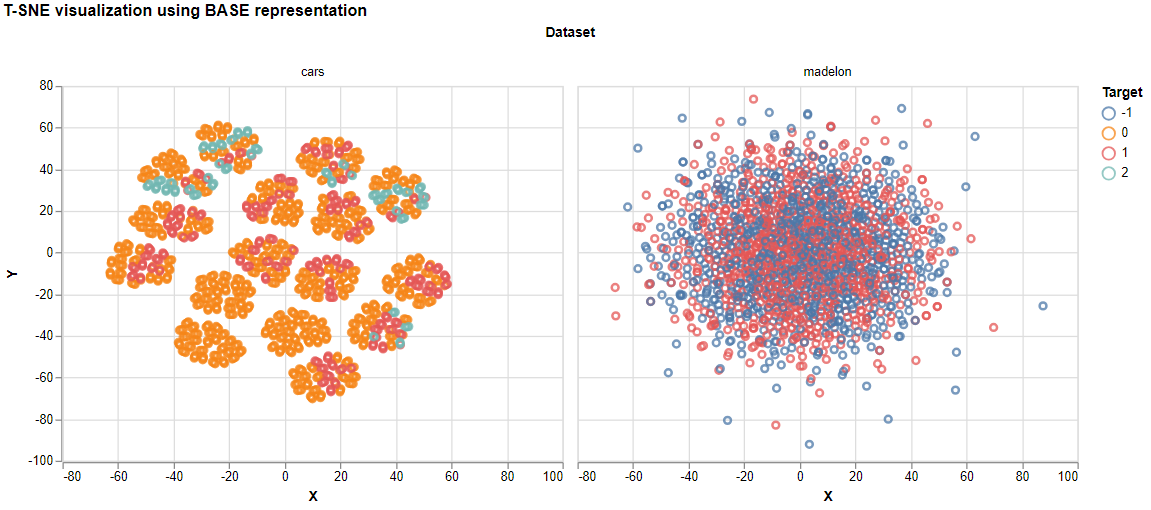
These perspectives of the data are generalized and indicate the range of components that capture general information well, but how do they perform when localized to a specific problem? For this section the number of components/features used was varied across a variety of hyperparameters for a NN on both datasets. Under the locality of the classification, the random forest feature selection along with the base data is included:

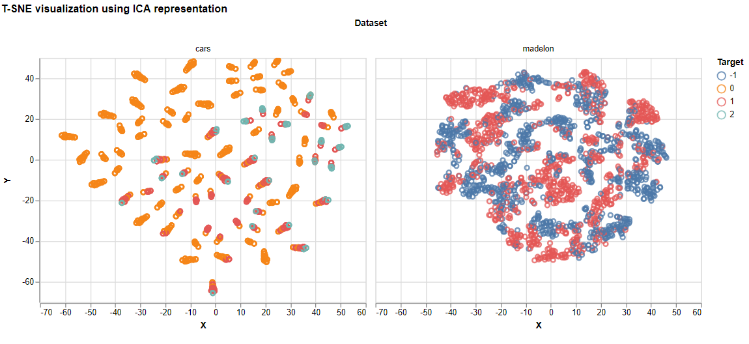
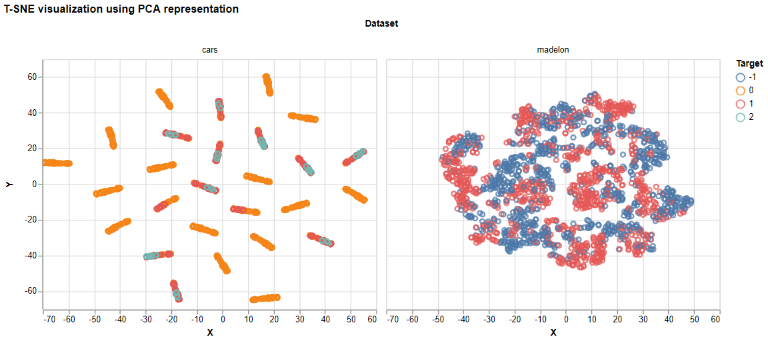


*Note: Madelon BASE data not shown as it utilized all 500 features and skews the X axis too much. Test score for Madelon BASE: 55%*

PCA and RF performed well in both datasets while ICA struggled. This may be a problem where independence of dimensions is simply not the best way to model the data. It’s interesting that PCA outperformed RF on Cars but not Madelon. I have a sense that this could be from overfitting. RF selected the likely 5 true features and 15 linear combinations of those 5 while PCA identified the 5 true features, yet the NN performed better when it was provided with the 15 weakly informative features. To me it seems the expressiveness of the NN allows it to overfit with the huge number of parameters those 15 features add (15 \* size of each hidden layer). Perhaps increasing the number of CV folds from 5 to 10 or 15 would change the results. ICA performed poorly in both cases while RP did well in Cars but poorly in Madelon. The close number of components and scores on Cars indicates that there is some sub structure of importance within the dataset and RP is picking this up somehow. However in a large dimensional setting like Madelon there’s too oppurutnity to pick up noise, as evidenced by its 45 components and low test score that is similar to using all 500 features.

|  |  |  |  |
| --- | --- | --- | --- |
| Data\_Perspective | Dataset | N\_Components | Best\_Test\_Acc |
| RF | Madelon | 20 | 0.863736 |
| PCA | Cars | 12 | 0.855324 |
| BASE | Cars | 19 | 0.821181 |
| RF | Cars | 16 | 0.815972 |
| RP | Cars | 14 | 0.803819 |
| PCA | Madelon | 5 | 0.789011 |
| ICA | Cars | 14 | 0.710069 |
| ICA | Madelon | 15 | 0.599451 |
| BASE | Madelon | 500 | 0.553846 |
| RP | Madelon | 45 | 0.545604 |

It’s tricky to get a grasp for the structure of the transformed feature space. One such method for doing so is T-SNE which projects points down to two dimensions while attempting to retain similar points close together and maintain distance from dissimilar points:



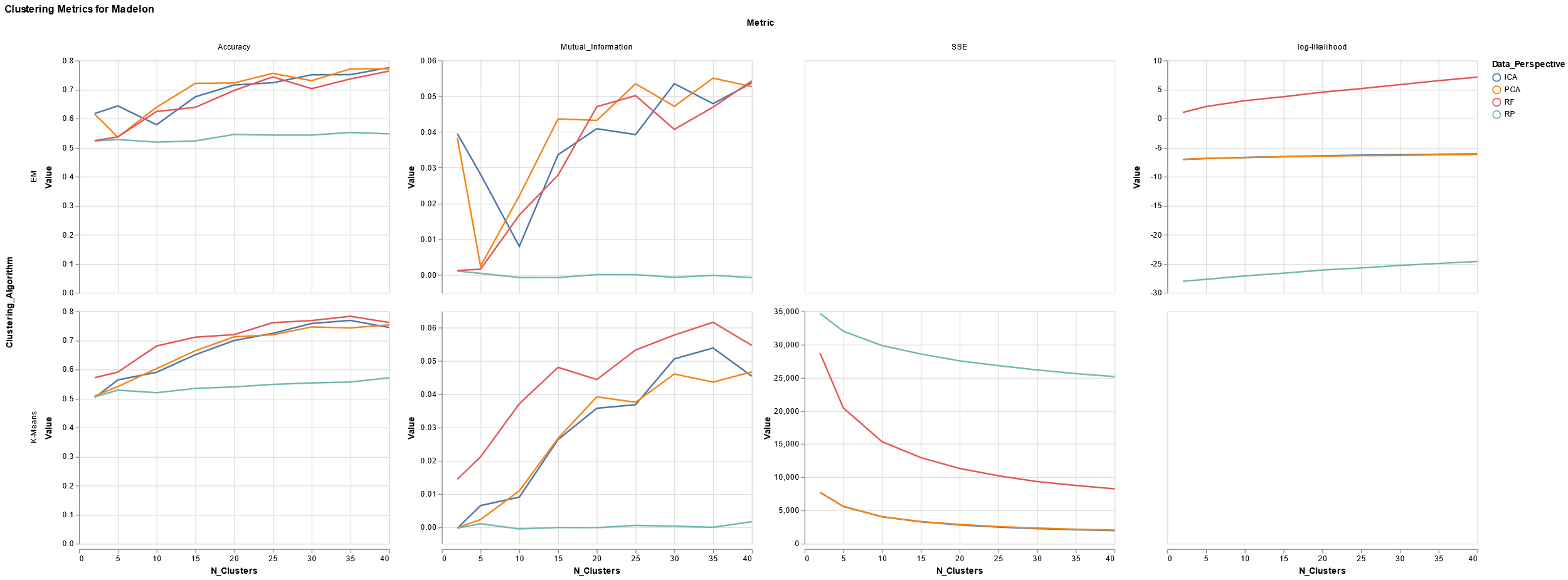
# C:\Users\afurrier\AppData\Local\Microsoft\Windows\INetCache\Content.Word\t-sne-rf.pngC:\Users\afurrier\AppData\Local\Microsoft\Windows\INetCache\Content.Word\t-sne-rp.png

# The t-sne visualizations do a good job of capturing how the feature transformations see the data. The RF feature selection (bottom left) shows groups of similar data points that are likely the end nodes of its overall decision tree. For the most part, the are pure nodes with a few exceptions. 3 of the clusters have a mix of classes 0 (poor car) and 2 (excellent car) which is interesting as one would expect these to be very different. All transfromations struggled with the high dimensional, nonlinear Madelon set. PCA/ICA/RF did a decent job of separating the data points and all look somewhat similar to each other, suggesting that perhaps they are all doing similar work in idenfitying the 5 true features and 15 weakly informative features but in different ways. RP suffered on the Madelon set as it returned a Gaussian blob, suggesting that its random projection was creating all noise and finding no structure. Each of the algorithms found strong structure within the Cars data and in different forms. PCA/ICA divided into small clusters while RP separated into 4 large clusters and RF/BASE numerous clusters . PCA performed quite well as all of the clusters either contain only poor car observations or good/very good.

## Clustering

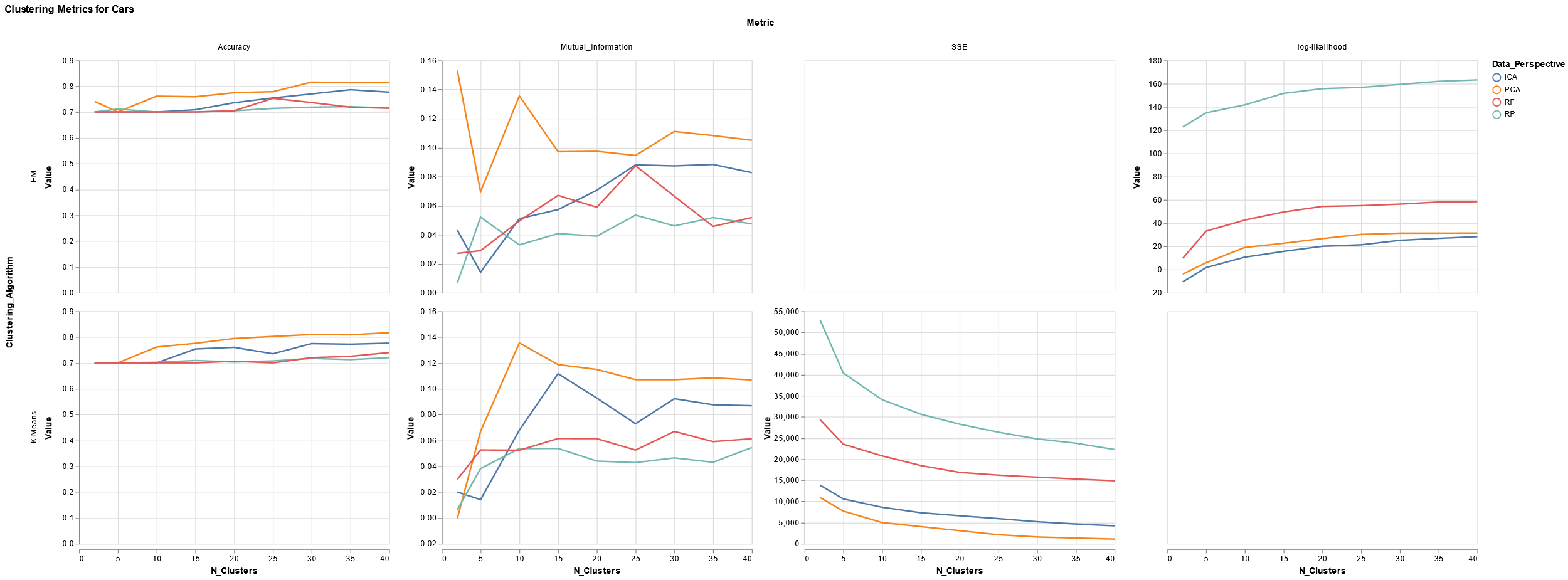
Identifying subsets of data which are more similar to members of the subset than non-members is the goal for clustering algorithms. The two main clustering algorithms examined are K-Means and Expectation Maximization. K-Means assigns K cluster centers and adjusts until the SSE within clusters is minimized while EM assigns a probability of each cluster label for each data point through a similar iterative process. Quality of clusters can be measured by SSE for K-Means (where smaller is better) while EM is measured with log-likelihood (where higher is better and indicates the fit of the data to the clusters). These are generalized metrics, but localized analysis can also be performed. The mutual information between cluster label and the target label can be computed, as well as using the cluster label as a predictor for labels (majority cluster target label is the predicted label). Additionally a neural network was trained across a variety of parameters and clusters: K-Means and EM clusters generated from all perspectives of the data (BASE, ICA, PCA, RP, RF) as well as the non-clustered data for comparison. Averaging across 5 fold CV results and averaging across a large range of hyper parameter configurations for learning rate and hidden layer sizes gives an idea for how the test accuracy and training time vary across the number of clusters (components for non clustered data).

*Madelon Cluster Quality Results*



Here Madelon exposes random projection as noise, with it performing the worst in every metric among all data representations by a considerable margin. Particularly in mutual information, the clusters created from the random projection is nearly independent of the target (~0). The clusters themselves do have decent predictive power in that they beat a naïve classifier with accuracies in the 60-70s range. The two clustering algorithms are similar but EM performed a bit better with low number of clusters (particularly with 5 or so clusters). This is reflected by the mutual information chart, where very low clusters have high mutual information for ICA/PCA while K-Means does not. The rest of the clusters have similar performance across data representation and clustering algorithm with the general trend being positive. RF and K-Means performed well with mutual information but worse in the cluster quality metrics than ICA/PCA. In general ICA/PCA performed nearly identically in SSE/likelihood where the elbow method begins to flatten out after 10 clusters. Seeing as the mutual information begins to flatten out after 15 clusters and a bias for simpler models, 10 clusters appears to be optimal. There’s an odd discrepancy between the RF selected features with cluster quality and mutual information. The RF clusters produced the highest likelihood for EM and worse clusters for K-Means compared to PCA/ICA, but the mutual information ranks are reversed. One would expect better clusters to have greater mutual information, but in this case it is reversed. When PCA/ICA have worse clusters than RF their clusters have higher mutual information and vice versa. This may just be due to the localized nature of mutual information, in that the clusters retain more general information but it doesn’t pick up meaningful information in regards to the target class. Since RP performed so poorly the optima number of clusters is roughly uniform: they are all bad, which implies 2 clusters if the data must be clustered given the bias for simple models. RF could be set anywhere between 10-20, but if forced to pick one perhaps 10 would be a decent choice.

*Cars Cluster Quality Results*

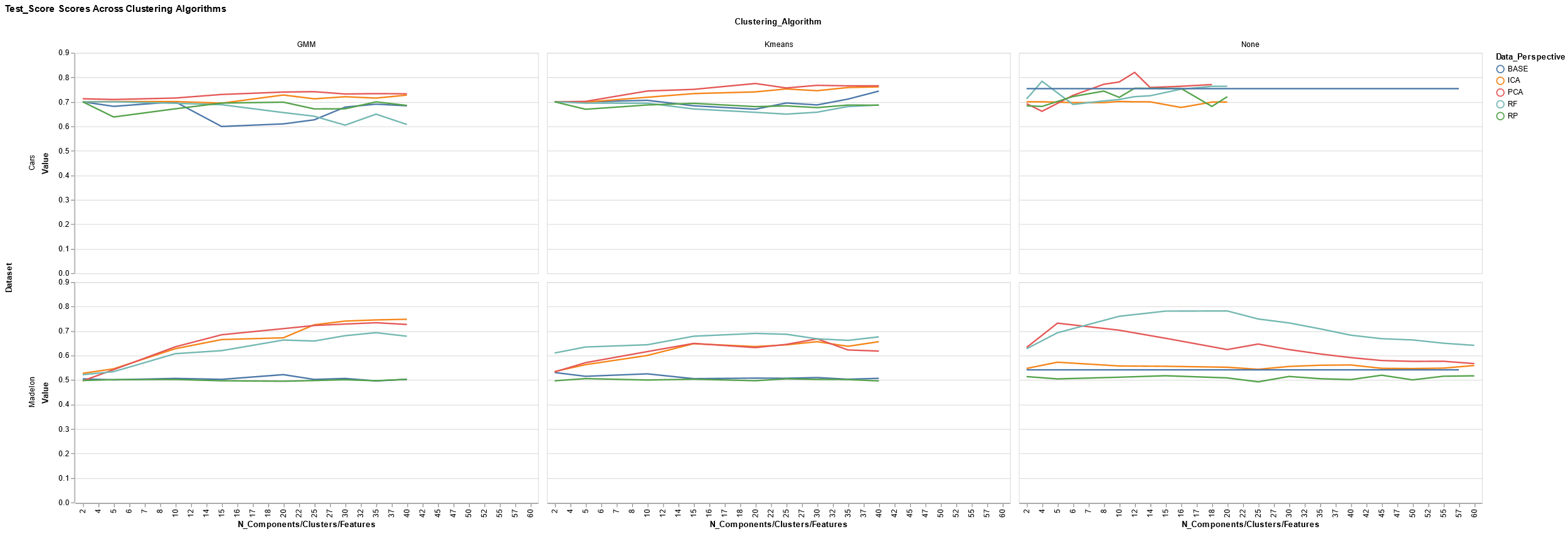
**

As opposed to the Madelon set, RP performed quite well in at least one metric on the Cars set. It had far and away the highest EM likelihood with a large gulf between it and the other data representations. The accuracy and mutual information of these clusters is at the lower bound of all the data representations which would still imply they are noise that the EM algorithm is overfitting (for K-Means RP had the highest SSE – another indicator of potential overfitting). PCA consistently performed best across all metrics while ICA lagged slightly behind. This is a departure from Madelon where they seemed to result in identical clusters. This would imply that their components and the data has variance resulting from non-independent sources, which differs from Madelon in that ICA/PCA identified the same linear combinations of the 20 informative features whereas here they do not. With this high signal dataset, PCA/ICA consistently outperform RP and RF across all metrics. That is, they retain information better as long as the information is out there whereas RF performed well when it needed to discern signal from noise. RP did neither. In terms of mutual information, PCA/ICA performed notably well 10-15 clusters across both K-Means and EM. From these results the optimal number of clusters given the tradeoff between cluster quality and localized information (accuracy, mi) appear to be 10 for PCA, 15 for ICA, 15 for RF and 10 for RP. With a high-signal, low noise dataset like Cars the number of optimal clusters is relatively close for all clustering algorithms and data representations.

|  |  |  |
| --- | --- | --- |
| Data\_Perspective | Dataset | N\_Clusters |
| PCA | Cars | 10 |
| RF | Cars | 15 |
| RP | Cars | 15 |
| ICA | Cars | 15 |
| RF | Madelon | 10 |
| PCA | Madelon | 10 |
| ICA | Madelon | 10 |
| RP | Madelon | 2 |

*NN Clustering Results*

*Test Scores across Number of Clusters*

**

With the cars dataset across number of clusters there was not a large amount of variance between the clustering algorithms and a bit with non-clustering data representations. This implies that with a high signal dataset there isn’t a great deal of lift to be found by representing the data in a different manner. While there isn’t large variance, clustering may be best suited for exploring the data for structure as opposed to using it as a tool for supervised learning of high signal data. PCA performed the best out the data representations and ICA had a noticeable bump when used in conjunction with K-Means clusters. In general, clustering performed better than the data without any clustering. Adding in clustering to a supervised learner may make the learner more robust, as even random projection largely outperformed the BASE data.

The usefulness of clustering can be seen in the Madelon results. Both K-Means and EM performed well across most cluster sizes. The main difference is seen in the lower end of the cluster numbers (<10 clusters) where K-means performed better than EM. Once again PCA performed well, but as opposed to the Cars dataset the RF feature selection was a top performer in this set. For low signal data, RF and PCA arrive at the same goal but from opposite ends. PCA identifies where the variance in the data lies even if it’s scattered across a vast amount of noisy data and combines into the essential components. RF does not combine but rather discerns which of the existing components is best suited for a targeted task (i.e. identifying a label) and in doing so discards most of the noisy data. In general clustering, both K-Means and EM, did not outperform the data without clustering. For such a low signal dataset it may improve the performance if instead of keeping all clusters the few which were relevant to the target (e.g. had large mutual information) were kept while the rest were discarded. K-Means and EM can identify substructures of the data, but in a supervised learning setting not all of those structures are important, just those that are relevant to target labels.

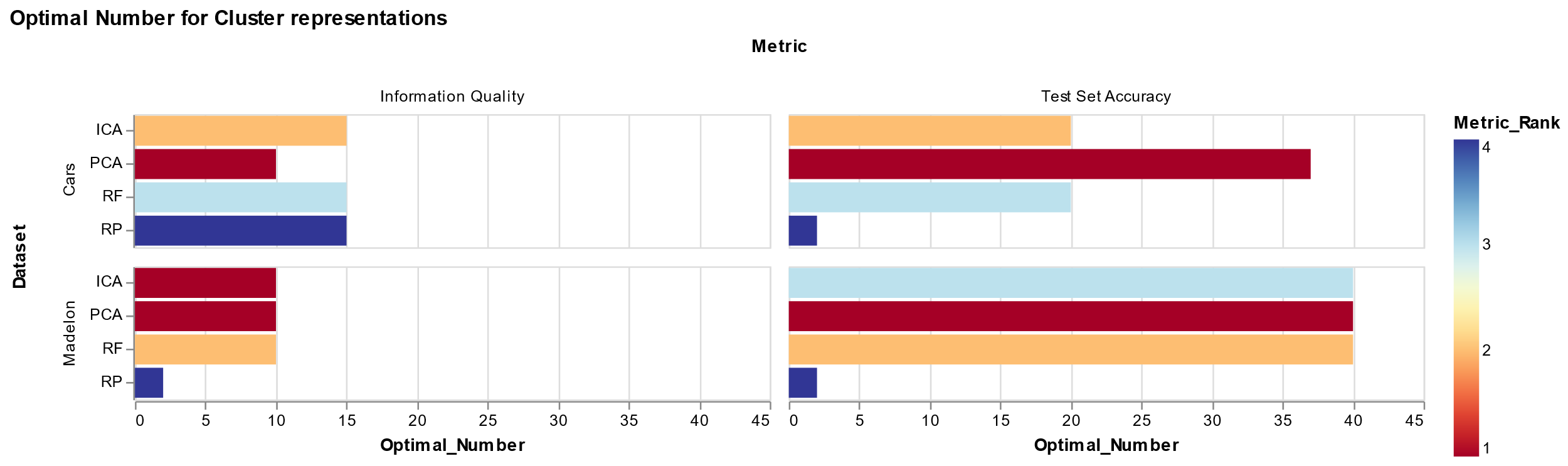
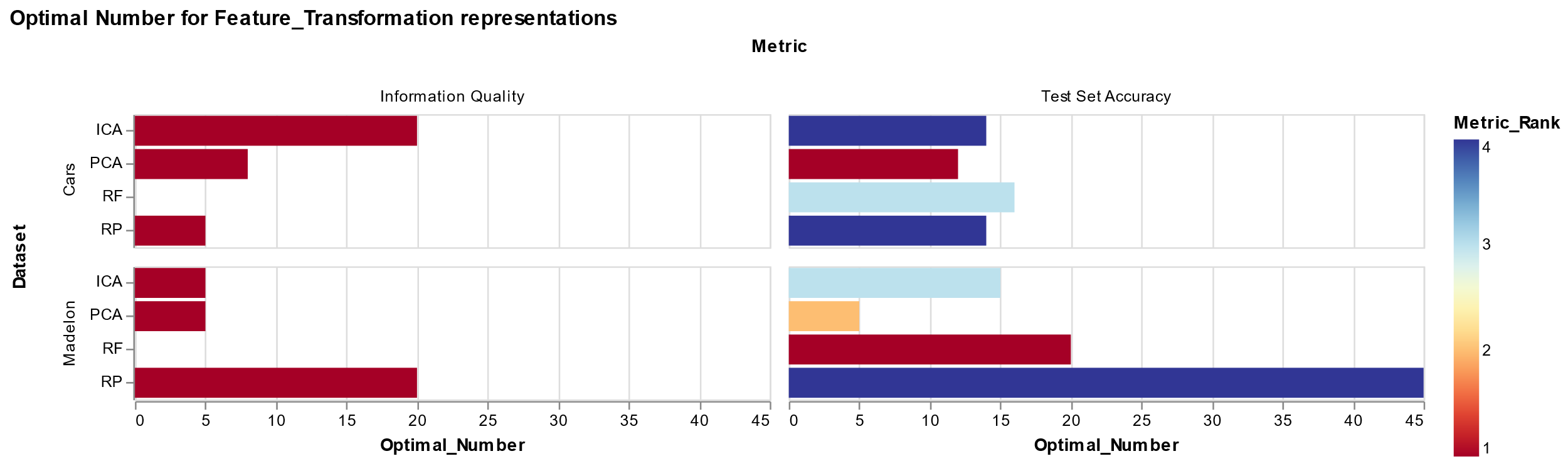
*Fit Times across Number of Clusters*

The curves for fit times show more variance than the test performance. In general RP and ICA took roughly twice as long to train as PCA and RF when clustering the data. With unclustered data RF took about 4x as long the others with RP taking the least amount of time. After 20 components ICA began to sharply increase its train time. These all line up with the underpinnings of each algorithms construction. In general, PCA will be very fast as it’s a closed form analytic solution for singular value decomposition (but would increase with data size and if the data was not standardized). ICA is not closed form but is iterative and needs to converge which is why it consistently takes more time than others. RF takes the longest with the base data because it has to train over a huge amount of possible splits in its bagged decision trees and then aggregate the most important ones. Once that has been done it most of the work is finished, but with the base data considered it repeats the process with only the selected features. Its train time is reduced sharply when the clustering algorithms get to work because it has largely removed the noisy features, i.e. what the clustering algorithms are left to work with is most likely the core structure of the data. Computing clusters from this subset should be minimal compared to testing out many possible subsets and combinations of features. Random projection is the opposite of RF’s computational cost. In the base data, the time is minimal. There’s not much signal to the RP so training over many components/features doesn’t require much time. However this is contrasted by the clustering algorithms which try their best to find structure where there isn’t much.

# Concluding Remarks

*Data Perspective vs optimal number of components/clusters*

*Color shows the rank from 1-4 between each data perspective and its respective metric (Red = better, blue=worse)*



These graphs illustrate the variety that unsupervised learning methods bring. In general, some patterns remain consistent (PCA performed well across datasets and cluster/non-clustering) while others varied greatly (optimal RP number on test set accuracy for Madelon was 45 unclustered and 2 when clustered). Depending on the objective of the data transformation optimal number of components/clusters varied quite a bit. Each method adds a different perspective of the data, and just as there’s no free lunch when viewing things pessimistically we can see the opposite: each method illuminates specific parts of the data that otherwise may be missed.

Through the analysis presented in this paper it is clear to see the contrasting strengths of each method: RF is good at discarding noise while PCA/ICA soaks up as much information as possible, including the noise. Clusters work the same way as they information, but represented a different way. Like a painting of the same subject, but from different artists. Some emphasize certain colors, others specific viewpoints of the subject. The truth is objective, but the representation is subjective based on the goal of the artist (maximize variance, independence, or the key characteristics relating that identify the painting). As an example, imagine a painting of person holding an apple while sitting on a bench. PCA would decompose the picture searching for the largest change such as greyscale. ICA may look for individual subjects within the picture: the person, the apple, the bench. RF would search for what in the picture stands out when identifying a specific subject. E.g. if trying to identify if there a human in the picture it would notice all edges that combine to form a human shape and color hue of skin tones while simultaneously ignoring the apple and bench. Random projection is random and may pick up any of the above. Add in clustering and potential to grasp combinations of characteristics grows deeper. While each method seeks to optimize specific goals, their overarching purpose is the same: retain more information with regards to a goal and ignore all other impertinent data.