**Abstract**

The world is full of problems, which implies there's learning to do. If you've got labels, I've got supervised learning techniques. Specifically, if the problem happens to be identifying the quality of used cars or identifying a non-linear classification with vast amounts of noise added, then the supervised learning algorithms for Decision Trees, Boosting, Artificial Neural Networks, K Nearest Neighbors and Support Vector Machines may be of help. For each of these datasets, and in turn each algorithm, the accuracy of classification was tested under cross validation over a variety of hyperparameters (learning rate, regularization, etc.) using sci-kit learn's GridSearchCV. The resulting hyperparameters, model performance, learning curve's and expressiveness' are examined. This analysis gives way to further insight into both the algorithms and problems.

# Datasets

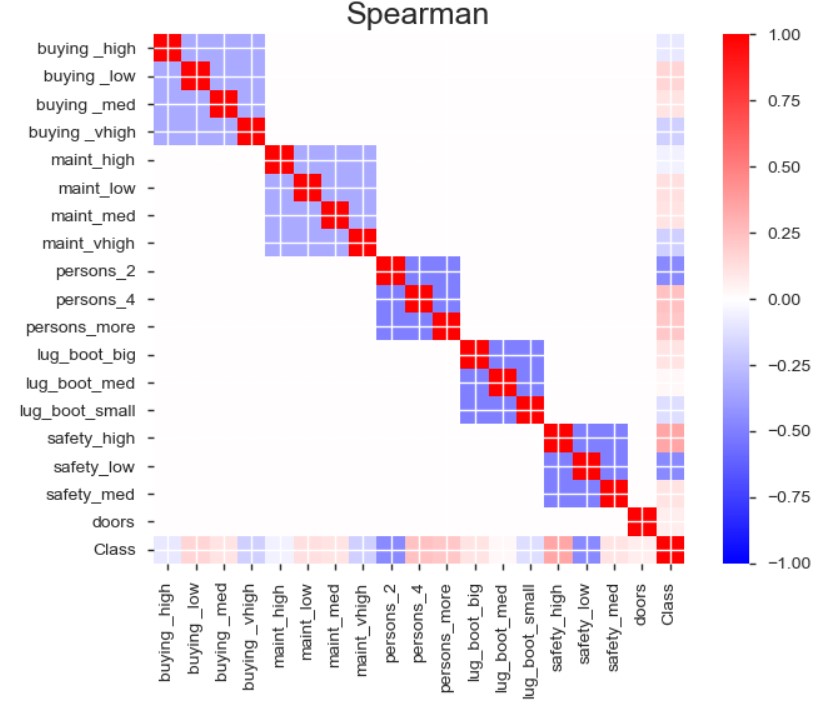
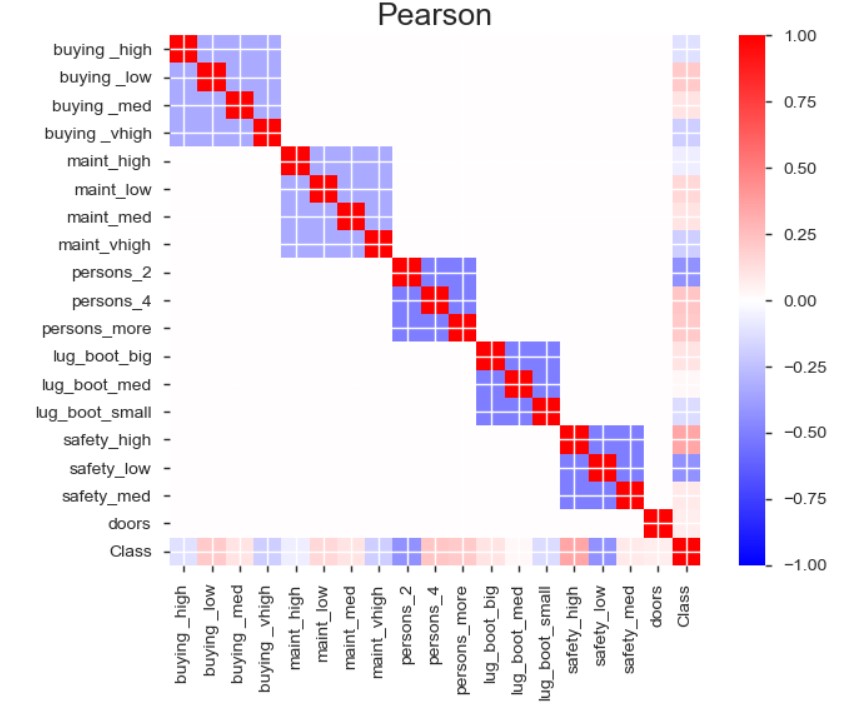
Interesting analysis requires interesting problems, and in order illuminate the strengths, weaknesses and quirks of the examined supervised learning algorithms two well known datasets from the UCI machine learning repository data are examined.

## 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).

There are no dead giveaways relating any of the categoric levels to the classes, as evidence by the correlation matrix. Since the labels are roughly ordinal in that a higher level indicates a higher quality, the Pearson and Spearman correlation should indicate a relationship for a categoric level being associated with an increase or decrease in label:



The bottom row of the figures highlight the relationship between the categoric levels and quality class. Most have a weak relationship (+- .25) at best, which will require the learner to recognize the combinations of the weak relationships in patterns related to determining class level.

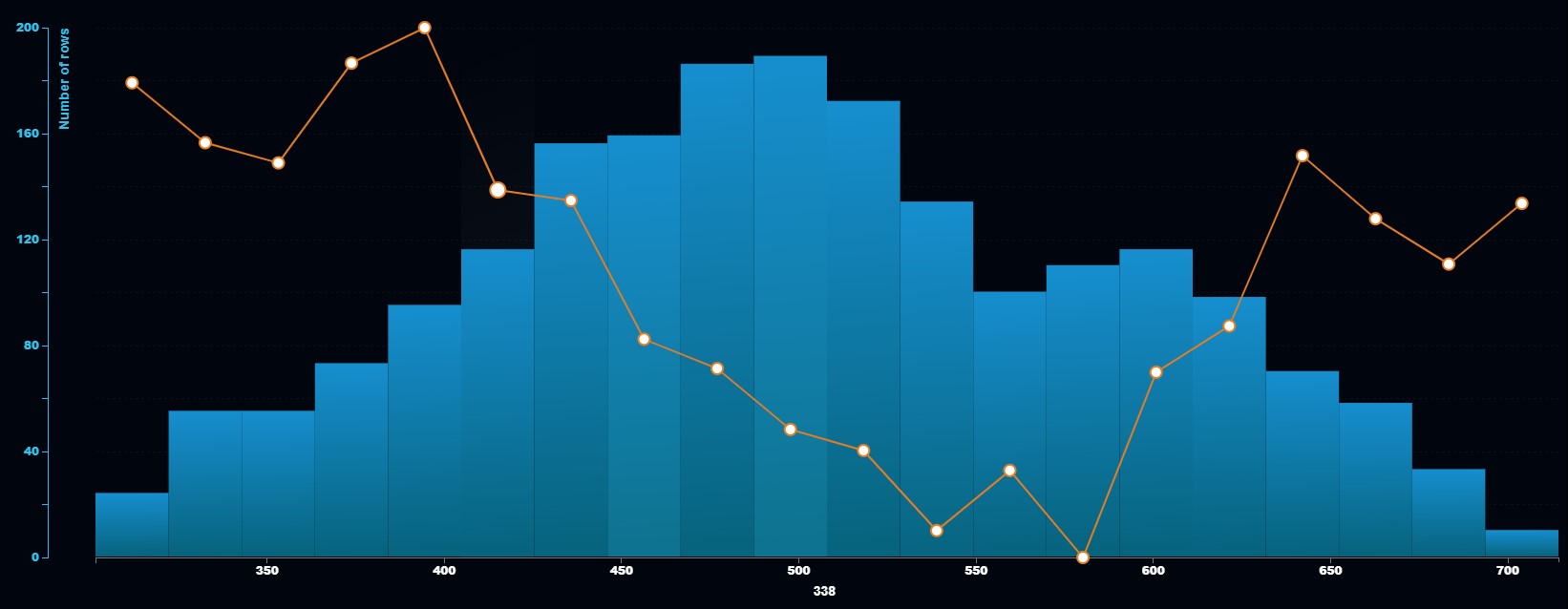
## Madelon

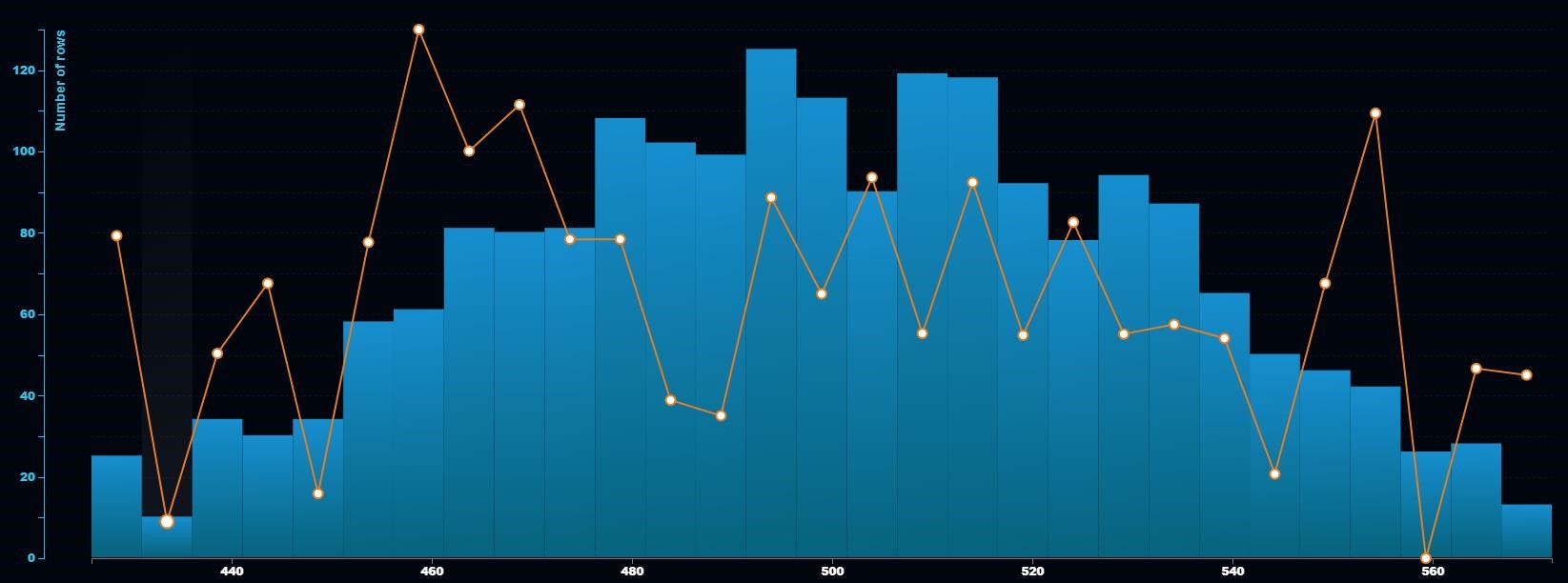
### Instances: 5000 | Attributes: 440 | Data Types: Continuous (440) | Classes: 0, 1

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 complicatedthe 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-tonoise 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. To alleviate some of the imbalance in signalto-noise ratio, sklearn's feature selection method SelectFromModel in tandem with a RandomForestClassifier was implemented. The feature selection was repeated four times with a threshold set to 'median', i.e. any feature deemed to be in the lower half of feature importance is dropped. In other words, the more important half of the features were kept with this repeated four times leaving 31 features for the algorithm to learn from. In the best case scenario, this would leave the 20 informative features and 11 noise features.

These plots reveal the difficult of the non-linearity and noisy features. Below is feature 338 with an orange dot representing the number of classes=1 for its associated bin. There appears to be a consistent pattern, albeit non linear with several inflections as the values increase. This is likely one of the 15 informative features.

 Feature 31 appears to have some potential information (especially with peaks around values 450 and 550) but is so noisy throughout that it is highly likely this is just chance and not actual signal.



The learners will have to distinguish between many of these noisy features to properly classify the Madelon dataset. In addition to the noise issues, the non-linearity of the problem presents an interesting challenge to the learning algorithms. Algorithms without the expressiveness to describe non-linear patterns, e.g. a linear SVM, may struggle on the dataset while others, e.g. an SVM with RBF kernel, may perform well.

# Algorithms & Methodology

All clustering and projection algorithms were implemented via the python machine learning package sci-kit learn. For each algorithm, the learner was five fold cross validation trained using balanced accuracy as the performance metric across a variety of hyperparameters. Balanced accuracy was chosen in light of the Cars dataset’s class imbalance, as without the cost balancing most learners would likely not focus on correctly classifying ‘good’ cars as they represent only 8% of the sample. The best parameters were stored, with the best performing classifier then trained on varying amounts of the data with its performance and wall clock time recorded to illustrate its learning curve and computation cost. The variance or expressiveness of the iterative/ensemble learners (ANN, Boosting, SVMs) was tested by measuring the train and test accuracy across an increasing number of iterations using hyper parameters with high expressiveness (i.e. regularization parameters set to very low values). All models are presented with their best selected hyperparameters and their respective learning curves and timing curves. If the graphs revealed something interesting, the ‘expressiveness’ curves were presented as well. All hyper parameter results and subsequent discussion can be cross checked with the result csv files under

‘reports/output/{learner}\_{dataset}\_reg.csv’. The change in hyper parameters across a dimension is discussed but rarely plotted due to space constraints. The files ‘reports/output/{learner}\_{dataset}\_reg\_pivot\_table.csv’ illustrate how scores varied across certain parameters, which is further discussed. All figures for the plots used (plus additional unused graphs) are under ‘reports/figures’.

## 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)

# Results

## For each, talk mostly about representation of the data. How well does it retain information?

* Across number of components: reconstruction error and other individual metric

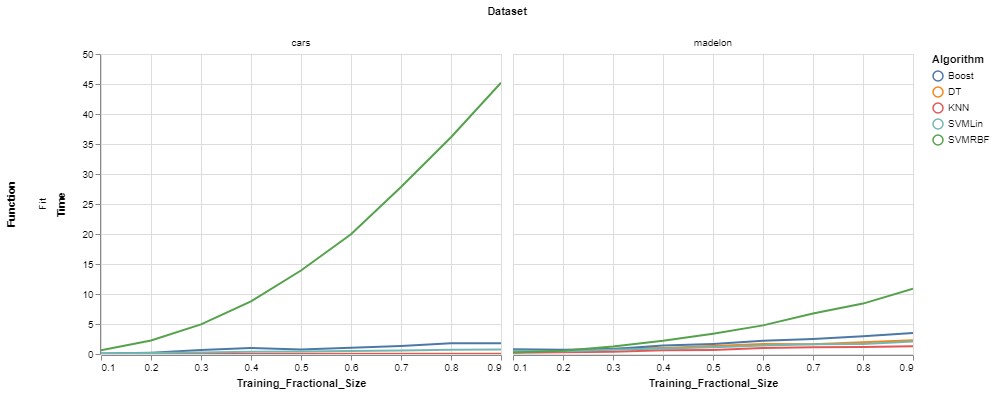
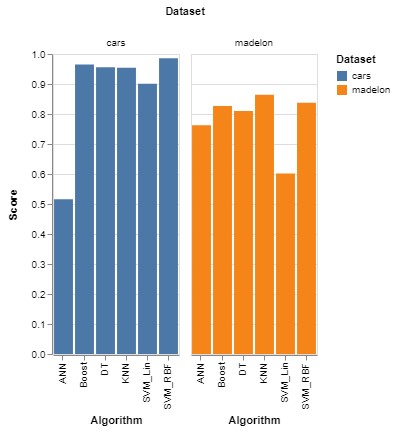
## Principal Component Analysis (PCA)

## Independent Component Analysis (ICA)

## Random Projection (RP)

## Random Forest Feature Selection (RF)

# Concluding Remarks



Left: 5 Fold CV Score

Bottom: Training Fit Times

(

ANN not included as it skewed the y axis as

to make comparison visually irrelevant)

The Cars and Madelon datasets provided contrasting problems for the supervised learners examined: one a small, high signal, hierarchical and sparse dataset (Cars) with the other a large (5000x500), high noise, non-linear dense dataset. The SVM with an RBF kernel performed the best overall with the highest score on Madelon and 2nd highest on Cars. Boosting did well as an ensemble method scoring 2nd best Cars and 3rd best on Madelon. The KNN proved to work best with the Madelon scoring a .86 overall (1st) but struggled on the Cars set (4th).

These problems and subsequent analysis further hammer home the necessity of aligning the problem space with the strengths of a learner and careful hyperparameter tuning to aid the learner with domain knowledge. Each of these learners possesses contrasting strengths and weaknesses- *this is not a bad thing* but rather an opportunity to craft a tailored solution to any problem. One size fits all solutions are never optimal, and such is the case here as evidenced by the datasets and the problems they pose. Before any model fitting takes place, it pays dividends to diagnose the unique aspects of the problem space and apply learners best suited to find a solution. Take the Cars and Madelon dataset as examples. It would make the most sense that a hierarchical problem would lend itself well to a decision tree. Lo and behold a boosted trees model scored very well on it, and with tuning performed very well with minimal computational expense. Similarly, for the Madelon data that was artificially created using points on a 5 dimensional hypercube was best approximated by KNN. KNN classified instances based on a weighted similarity score that came from a distance measure that lends itself well to high dimensions (manhatttan). The RBF SVM outperformed its linear counterpart due to the nonlinear nature of both datasets and its ability to project into higher dimensions.

Of course there are always more considerations when searching for the optimal solution, such as computational complexity. Although the RBF SVM performed best on the Cars dataset, it requires much more computation and time for convergence. If the use case involved massive amounts of data it would make far more sense to use a fast method such as a decision tree or boosted trees method which lends well to scale. Machines learning models are fast, accurate, and stupid. This of course is the case without careful analysis and thoughtful application, as this paper’s analysis illustrated the core importance of proper algorithm application and tuning involved in solving supervised learning problems.