Applied Machine Learning - Assignment 4

Anurag Sethi

Dec 08, 2019

1 Introduction

This project focuses on clustering and dimentionality reduction techniques. While algorithm application is a small but critical part of the project **the foremost motivation of the project is to get interpret clustering results, measure different clustering results with benchmark metrics and draw inferences regarding effect of dimensionality reduction on both prediction and clustering efficiency** . In first section, we do precrocessing and feature engineering, section 2 we implement clustering, dimensionality reduction, feature transformation algorithms on the *UCI Appliance Energy Consumption* dataset. In section 3, same series of algorithm implementations and model performance comparison are done for *Kickstarter Campaign Success Prediction dataset*. **In last section, section 4, the project concluding remarks which summarize the key learning takeaways of the project are presented, we compare the results and complete the experiments in section 2 and 3 itself**. **Note: Section 1 for this assignment is similar to assignment two, as we are using the same two datasets, hence they will have same preprocessing and feature engineering steps. Also, for kickstarter campaign prediction a smaller set of 10k data points is used due to computational constraints for implementation of Neural Networks.**

1.1 Data Preprocessing - Feature Engineering

The project makes use of two very interesting datasets Appliance Energy Consumption Dataset hosted at UCI machine learning repository a popular archive portal for open datasets for academic use. We create the new variable 'spike' where the recorded consumption exceeds 150 Kwh. By definition, our dependent variable indicates to the user potential spike in the electricity usage, such a working model helps the users to anticipate spikes and regulate the appliance usage to reduce the energy consumption and subsequently reduce their carbon footprint. There are multiple factors associated with selecting this particular dataset over others; from business point of view it is an interesting problem, crowdfunding is a recent phenomenon and it would be great insight if we know the factors that lead to success in crowdfunding campaigns. There can be many lessons for a person or a company aiming to launch their campaign, that could maximize their chances of raising money. Furthermore, this data-set had ample scope of feature engineering, which is always a great exercise, a model built with rudimentary parameter optimization with intelligent feature engineering can often beat a model built with advanced parameter optimization on raw features.

UCI Appliance Energy Consumption :

The dataset has 29 features to start with, there is dependent variable Appliances showing the energy consumption by appliances of the house. The other features can be broadly divided into four further categories, Weather Related Features,Humidity related features,Temperature Related Features: each category capturing the various entities at different parts of the house, also there are features like light, date, rv1 and rv2 which are put together as miscellaneous. In the pre-processing step numeric features are scaled using the z-score, implemented using the standard scaler operand inbuilt in python

Engineered Variable Name	Description
Month	Descriptor labelled 1 to 5 representing month
Is Weekend	Flag if the day is a weekend or weekday
Time of Day	Section of the Day eg. night time, evening time, work hours etc.

We engineer new feature from date variable, which are relevant based on the understanding of the context. There are listed in the table below. The rationale behind this is that intuitively appliance consumption is affected by seasons (captured by month), day of week (captured by isweekend), and section of the day

(captured by time of the day). The dependent variable is defined as *spike alert*; here category is defined as 1 when the consumption for the hour exceeds 150 kwh. In the dataset, the class split at this mark is 80 and 20 percent, which is a reasonable split from the prospective of modelling ease.

Kickstarter Campaign Success Prediction: The original set of raw variables can be obtained from here (https://www.kaggle.com/kemical/kickstarter-projects). Over these features, we have engineering a set of new features, these help us inject some domain knowledge in our prediction. The categorical columns are converted using one hot encoding and we get 42 features in our set.

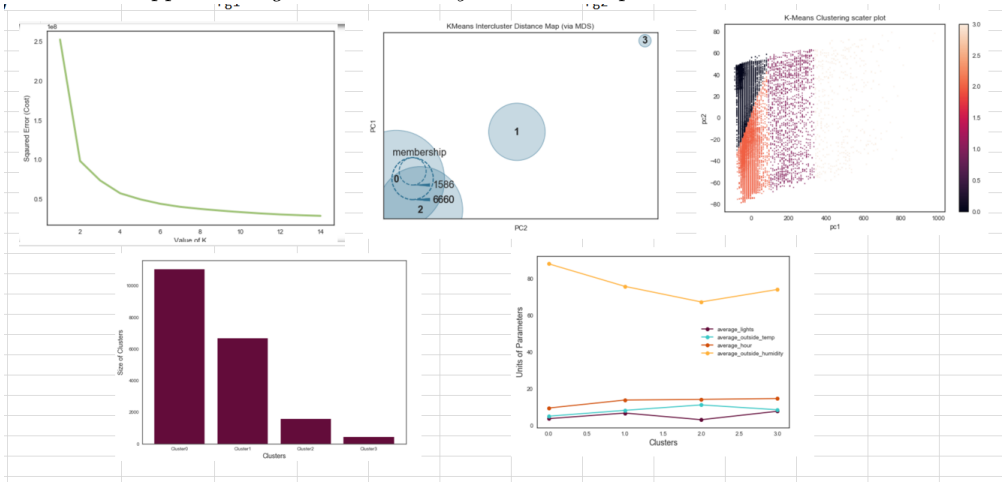
Engineered Variable Name	Description
Duration	Descriptor labelled 1 to 5 representing month
Is Weekend	Flag if the day is a weekend or weekday
Launched year, quarter	Year and seasonality information derived from date of launch
average amount per backer	Average amount each patron has given till now
goal level	Money campaign is targetting divided by mean goal of projects in the category
competition quotient	Total same category projects launched in the same month with given campaign
syllable count	Number of syllables in the campaign name

2 Appliance Energy Consumption

In the subsequent subsections, we deeply explore the two classification techniques and predict spike in appliance energy usage. We see we have 11.6 % of classes marked as positive (2298 out of 19735) where the usage exceeds 150Kwh. **In this section, we first apply kmeans, Gausian Mixture ModelS (GMM/Expectation Maximization). We follow this by implementing four dimensionality reduction / feature transformation algorithms, followed by the two clustering algorithms after each step. We then move on to Neural Networks and apply it at four different stages (illustrated in detail at the end of this section)**

2.1 Appliance Energy Consumption: Pre-Dimensionality Reduction

appliance Fg1: Rudimentary K means implmentation on Raw Features

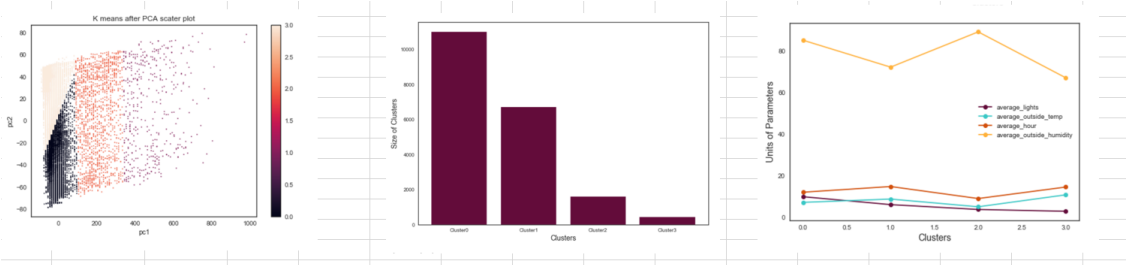


Top left: In clockwise order: elbow curve, intercluster distance plot, cluster viz., independent variables variance in clusters,cluster size distribution

The first step is to implement two clustering algorithms on the raw features, the initial thing is to find the value of appropriate k for K means clustering. We begin by drawing the elbow plot and observe the marginal decrease in the SSE, at a particular point k=4, we see that the marginal change is minimal, hence k=4 is an appropriate value to decide the number of means to initialize for k means clustering. **An important thing to note is that optimal value for k means and expectation maximization can be same. The underlying pricipal of both the algorithms is same and it can be argued that k means is a special case of EM.** To visualize the clusters we use, an interesting package called yellowbrick cluster. It gives a fair idea of scale about how part apart we have the clusters. We visualize the clusters by plotting the results

obtained after applying k means (above) and expectation maximization (below) on raw features. In the excel scrapbook, attached with the assignment, the cluster plot with two variables outside temperature and humidity is also there. But it was more intuitive and easier to visualize the results plotting on two principal components. Hence for the sake of uniformity we stick to this approach throughout the course of this project. We can also see plots of cluster sizes,**For the clusters to be considered good clusters it is important for them to conform to traditional yardsticks like maximizing inter-cluster distance but also for high quality output we need to make sure that we shouldn't have too many small sparse clusters. Later on we introduce another metric, which is the deviation of the cluster sizes, it augments other cluster analysis metrics and gives a more holistic view.** In the last plots we see how our independent variables vary, but analyzing the **clusters centres**. It is evident that the four clusters have a varying degree of light, time of day, outside temperature etc. For instance cluster 1, has higher than average humidity and clustering 2 may very well contain time slices of evening hours. **Even more interesting analysis is the distribution of dependent variable over the created clusters which will be covered later** In the next section we apply dimensionality reduction algorithms and then reapply clustering algorithms.

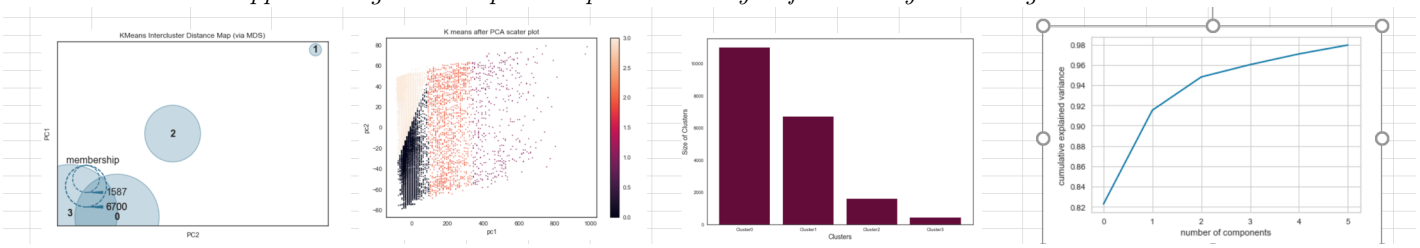
appliance Fg2: Rudimentary Expectation Max implementation on Raw Features



left: cluster viz., cluster size distribution, independent variables variance in clusters,

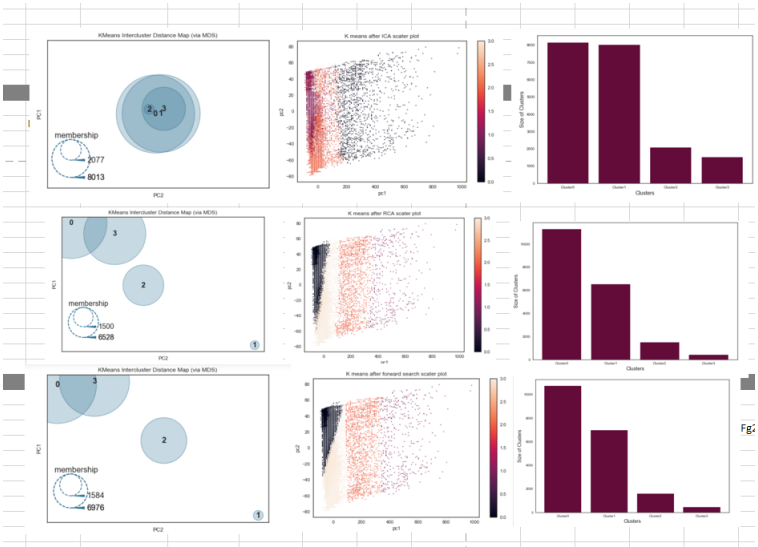
Below is the principal component analysis, a very common variance based dimensionaliy reduction technique that has been throughly studied in machine learning class. It transforms the original components into new orthogonal components with each component having monotonically decreasing variance explained. *As we can see first 3 principal component explain close to 95 percent of variance. Post that the marginal change plateaus.* We use these principal components and have thier outputs as inputs of our clustering algorithm. Looking at the first plot below, inter cluster distance chart we can see which clusters are close and which ones are apart from each other. Like before we visualize the clustering results and the sizes distribution. In the later section when we measure the algorithm performance, we quantify these metrics, **db index, inter cluster distance, size variance factors which gives us an excellent comparison mechanism between various techniques.**

appliance Fg3: Principal Components Analysis followed by clustering



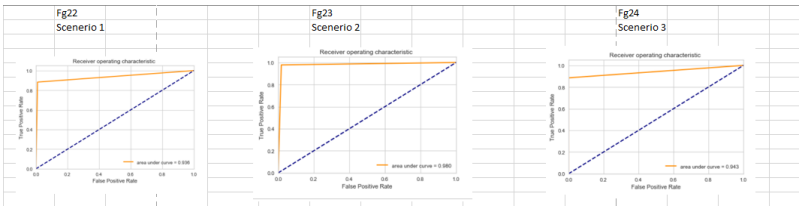
left: inter cluster distance part, cluster viz., cluster size distribution, explained variance plot

In this part we implement three algorithms, two for feature transformation (Independent components, Random Components) and one for feature selection (forward selection algorithms). **It is a crucial aspect to know to decide the number of components to select for these algorithms** for say **best forward search we see in the code that at n=20, the accuracy metric of the wrapper classifier in our implementation it is logistic regression** is optimal, in the python implementation it is visible in the stacke-trace when the wrapper algorithm iteratively runs, similarly ica/rca shows predictive results at 10 components, **This is a important caveat that a learner should have in mind that if the end goal is unsupervised learning then having predictive results as a bench mark may not be the best idea.** Looking at the plots below it is evident that ICA gives poor results with cluster overalapping, RCA on a first glance shows good results in the cluster distance plot. This will be validated again in the later section. Forward search also gives good results as evident from inter cluster plot.



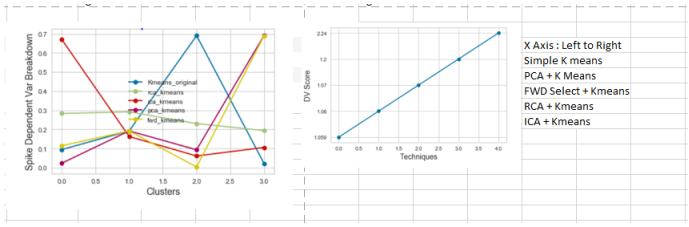
From Top row to bottom: inter cluster distance plot, cluster viz., cluster size distribution, for ica, rca, forward selection respectively

After running the clustering algorithms we run the following, **We already have a benchmarked optimized and tuned version of Neural Networks from as3** As asked in the tasks we implement, NN on the reduced dimensionality data. Like in previous assignments we use Area under the curve as our metric. **We improve the previous best score of 0.92 in assignment 3 quite considerably by reducing the dimensionality. It may suggest that our algorithm was suffering from the curve of dimensionality**. In the second scenario we provide our clustering output (pca+kmeans output)label as one of the variables for prediction along with the three principle components, we see the AUC is maximum under this scenario at very high 0.98. It can be very well inferred that cluster label is treated like new information by the neural network model. In the first scenario we only have our clustering outputs (all implemented versions of kmeans and em along with reduction techniques) as input labels. Even though AUC is good but it does show that labels along with only themselves miss a lot of information that was conveyed by the principle components in scenario2.



From left: AUC curve post PCA and Neural Nets, AUC curve post PCA and Neural Nets and adding cluster label as predictor to PCA, AUC curve after using NN on only cluster results as predictor

Below are two interesting graphs, **we first study the distribution of dependent variables in proportional terms over the four clusters. If a clustering technique divides the depending variable in such a way that it disproportionately falls in to one cluster, it could indicate that its' clustering output could be a good feature variable.** We see all techniques except ICA more or less do this job in this case. Later we look at **an important metric called DB index. There are multiple ways to determine the robustness of clusters; but the underlying intuition behind all clustering algorithms is that the inter cluster distance should be maximized and intra - cluster distance should be minimized**. DB index implements a similar calculation, it is a pair wise sum of a metric that has the sum of intra cluster distance in the numerator and inter cluster distance in the denominator. For better clusters it is low, of the implementations we have shown here simple k means on raw features has the least DB Index whereas **ICA had the highest DB Index** indicating that it is not a good technique to use here.



Left: Cluster Analysis: Breakdown of Dependent var. over Clusters. Right : DV Ratio for other each dimensionality reduction technique

	Cluster Efficiency Comparison					Neural Net Performance Comparison			
	Technique	Inter-Cluster-Distance	DB Index	Size Variance		Scenerio 1	PCA + NN		
1	Raw Features K Means	300.81	1.509	4889.95		Scenerio - 2	PCA + Cluster Label as feature +NN		
2	Raw Features Expectation Max	190.94	4.453	4639.36		Scenerio - 3	only Cluster Lables as features + NN		
3	PCA + K Means	300.8	1.06	4877.44					
4	ICA + K Means	0.2	1.86	3766		BASELINE METRIC	Scenerio -1	Scenerio - 2	Scenerio - 3
5	RCA + K means	237.3	1.2	5002	Train Error	**taken from asign3**	0.9821	0.982	0.9862
6	Forward Search + K means	299.95	1.07	4802	Test Error		0.9777	0.981	0.9868
7	PCA + Expectation Max	253.7	4.45	4877.44	AUC	**0.92**	0.936	0.98	0.9426
8	ICA + Expectation Max	0.016	2.28	2768					
9	RCA + Expectation Max	83.97	4.8	5002					
10	Forward Search + Expectation Max	10.91	2.24	4802					

Left: Detailed Cluster Analysis, Inter Cluster Distance, DB Index and Cluster Size Variance for different used techniques, Right: Neural Network Performance under various scenarios

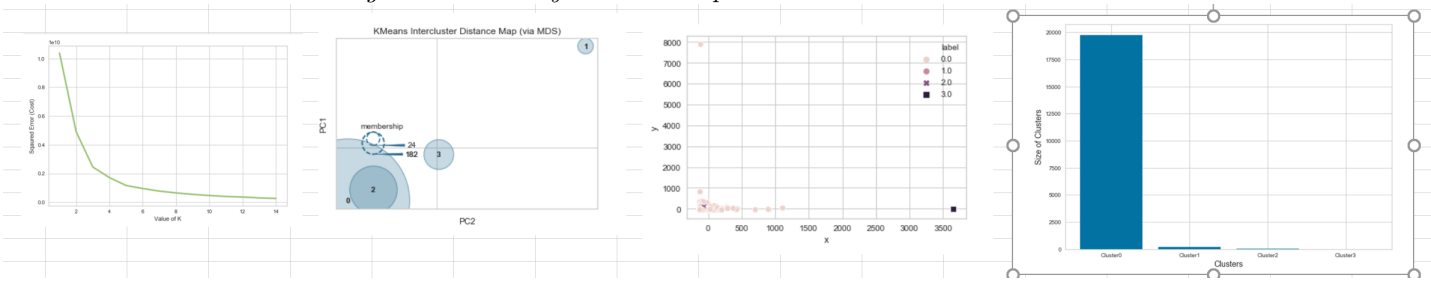
In the right as mentioned, we set up Neural Networks in three scenarios with an optimal benchmark we obtained from assignment 3. This version of neural network, had optimized version of hyperparams like batch size, epochs, activation function, optimizer, number of layers, number of neurons in a layer. **In this case our improvisation with dimensionality reduction and inducing new predictors from clustering output does help us drastically improve out results, however it may not be true for every case, we will see the results in the other dataset and implore on the the possible reason for performance degrade. The table on the left is essentially the summary of the entire project for appliance energy consumption prediction.** Here we can see the output for Inter cluster distance, DB Index, Size Variance for each and every run of dimensionality reduction followed by clustering. **It should be observed that in this particular problem kmeans does better than k modes. The trick is to find the implementation that balances all fronts. From the business standpoint it is usually hard to define small sparse clusters and they dont' usually convey an underlying pattern (hence the need for our third metric). K means on Raw Features, Forward Search and PCA does comparatively better in all the fronts and could be suitably used. Our quantified observations in this table are on par with out plots like inter cluster distance graphs in earlier section.**

Section 3 starts on next page

3 Kickstarter Campaign Success Prediction

We now begin replicating most of the work, for Kickstarter campaign success prediction. It is important to note that there are significant differences in the nature of the datasets and the nature of business problem we intend to solve. These differences will be more clear as we more ahead with our implementations

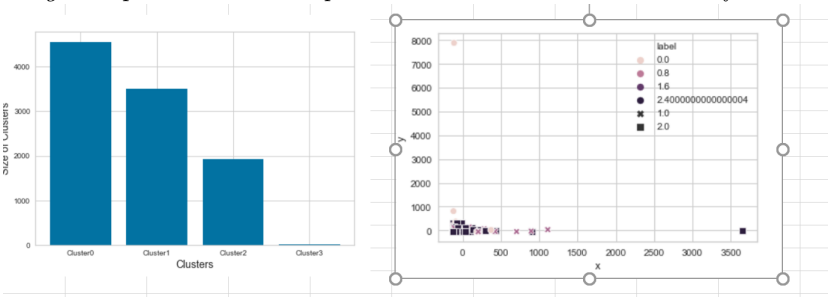
Kickstarter Fg1: Rudimentary K means implementation on Raw Features



left to right: elbow curve, inter-cluster distance plot, cluster viz.,cluster size distribution

Again in the first step we first take the raw features , the initial thing is to find the value of appropriate k for K means clustering. We begin by drawing the elbow plot and observe the marginal decrease in the SSE, at a particular point k=4, we see that the marginal change is minimal, Just like before we can pick k=4 clusters and initialize it for both k-means and expectation minimization algorithm which is a Bayesian approach to k means clustering. The yellow brick visualization tells us the size of the clusters mapped to its scale. We see almost all the data mapped into one cluster and less than 5 percent in the remaining clusters. **Even if this setup minimizes the sum of squared error it might not be the optimal way to cluster** as it leads to sparsity of clusters, with a bare few of data points grouped together without much meaning. **We will observe this phenomenon in some of the other implementations as well, on first thought a possible reason for this to happen is that unlike the previous data set where we were predicting the spike in electricity usage, which happens quite often, this problem has a lot of class imbalance**, with an overwhelming majority of kick starter projects going unfunded, hence the reason for one large cluster. At this stage we haven't looked at any numbers so our guess is speculative at best. We will get the opportunity to make more quantitative observations in the later sections. Correspondingly **Expectation Maximization works better on raw features** this might indicate that it is a better idea to assign probabilities to points and then optimize in this kind of problem. Even in the scatter plot the cluster differences are more profound. We can also see that the distribution of cluster sizes is more desirable when we use expectation maximization approach.

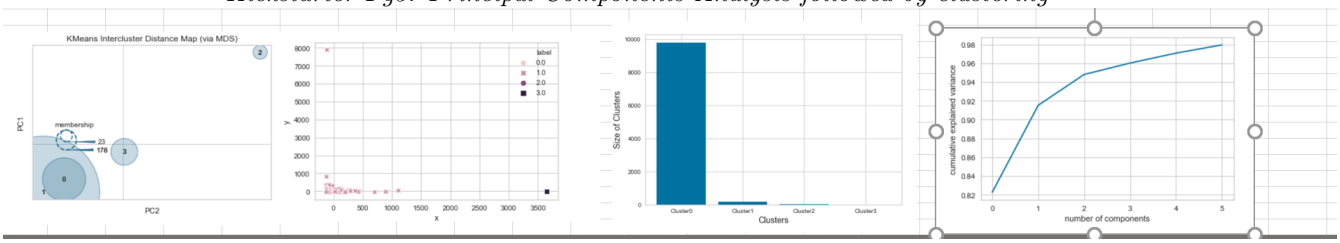
Kickstarter Fg2: Expectation Max implementation on Raw Features : left: cluster size, right cluster viz



left to right: elbow curve, inter-cluster distance plot, cluster viz.,cluster size distribution

In case of principal component analysis, we see that we obtain the marginal increase in the explained variance in the beginning itself, with top 3 principal components explaining close to 96 percent variance. It should also be noted that for this dataset we have sampled 10,000 project from the library of close to 330,000 projects due to computational memory and speed constraints, specially related to neural networks. In the cluster graph below it is evident that it is not a good clustering setup with 3 clusters overlapping in properties and similar features and one farthest. **Is it possible that that cluster have disproportionate number of approved projects. It is an interesting question, we will find out when we do the distribution of target variable over the clusters in the later section. This exercise also highlights the importance of cluster analysis from purely a business point of view as we do from a algorithmic point of view by benchmarking against metrics.**

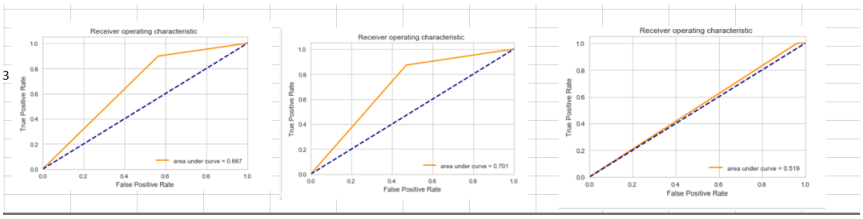
Kickstarter Fg3: Principal Components Analysis followed by clustering



left: inter cluster distance part, cluster viz., cluster size distribution, explained variance plot

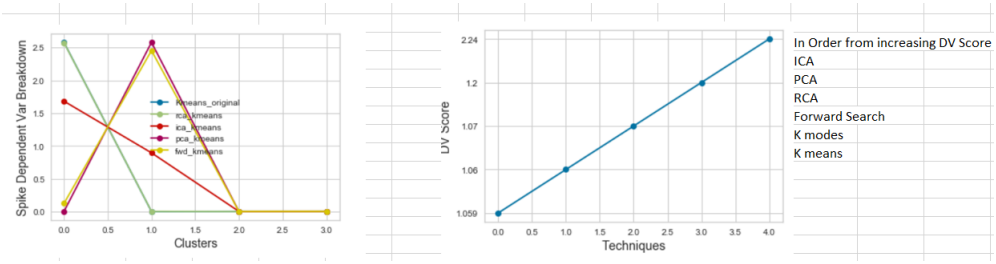
In this section, as we see in the figure below, we compare the three implementations we have been studying, as in the previous dataset. Although we do have the metrics calculated for RB Index, Inter Cluster Distance and Size Variance, at this stage we only make visual inspections looking at the plots. We can clearly observe that ICA and RCA are not a good implementation. Forward search could possible be and we will be able to make effective conclusions later. The overlap in the inter cluster graph directly corresponds to very low inter cluster error which leads to poor clustering. The three implementation show poor clustering output. The methedology for closing the number of parameters for forward search, rca and ica remains the same as before. We stop at the points where predictive ability ceases to increase substantially upon increasing number of predictors, it is visible in the stack trace of running forward search in python module *SequentialFeatureSelector.*

From Top row to bottom: inter cluster distance plot, cluster viz., cluster size distribution, for ica, rca, forward selection respectively



From left: AUC curve post PCA and Neural Nets, AUC curve post PCA and Neural Nets and adding cluster label as predictor to PCA, AUC curve after using NN on only cluster results as predictor

Moving ahead we implement the three scenarios discussed previously. In our last example adding a clustering label had a positive impact and improved the overall result. **However here we see the opposite effect**. In scenario 1 where we are implementing the Neural Network on reduced dimensionality using principal components, our Area under the curve metric drops, **we have the benchmark metric for this experiment from assignment 3** which is 0.81. Furthur when we introduce a cluster label as one of the predictor it drops furthur,and upon using only cluster outputs as predictors it drops even more. **the reason for this is the fact that the learner like neural network is very sensitive to it hyper params compared to most other predictors, our current configuration of neural networks is optimized for features in the original plane, the transformed features are not in coherence with the parameter setup, hence in this case for the sake of consistency it will be hard to make a comparison. This problem is even amplified by the nature of this problem having a high class imbalance.**



Left: Cluster Analysis: Breakdown of Dependent var. over Clusters. Right : DB Ratio for other each dimensionality reduction technique

This is a very interesting graph above, we see the breakdown of the dependent variable, that is the successful state over the clusters for different clustering implementations, **In forward kmeans and pca kmeans we see close to 25 percent of the successful projects fall in one cluster (the class proportion of success is much less), this is a good indicator that output of these two algorithm could be correlated with the dependent variable.** On the right we see the similar plot, about DB score which as explained before maximizes the inter cluster distance in the denominator and minimizes the intra cluster distances, we can see that ica and pca implementation followed by k means result in low DB score. **This metric is a part of sklearn metrics and is widely used to estimate cluster robustness. link added for reference : https://scikit-learn.org/stable/modules/generated/sklearn.metrics.davies$_b$ouldin_score.html**

Cluster Efficiency Comparison

	Technique	Inter-Cluster-Distance	DB Index	Size Variance
1	Raw Features K Means	11393	0.4737	4859
2	Raw Features Expectation Max	10533	3.47	1985
3	PCA + K Means	11405	0.47	4862
4	ICA + K Means	0.43	7.53	3138
5	RCA + K means	9705	0.74	4861
6	Forward Search + K means	5234	1.07	4602
7	PCA + Expectation Max	10633	3.47	4862
8	ICA + Expectation Max	0.03	1.63	3138
9	RCA + Expectation Max	9144	3.474	4602
10	Forward Search + Expectation Max	5270	2.7	40.6122

Neural Net Performance Comparison

	Scenerio 1	PCA + NN		
	Scenerio - 2	PCA + Cluster Label as feature +NN		
	Scenerio - 3	only Cluster Lables as features + NN		
	BASELINE METRIC	Scenerio -1	Scenerio - 2	Scenerio - 3
Train Error	**taken from asign3**	0.711	0.748	0.62
Test Error		0.0698	0.724	0.585
AUC	**0.82**	0.667	0.701	0.519

Left: Detailed Cluster Analysis, Inter Cluster Distance, DB Index and Cluster Size Variance for different used techniques, Right: Neural Network Performance under various scenarios

Above is like before the summary of the entire implementation of both clustering results as well as neural network results, we have previously discussed the low Area under the curve score after dimensionality reduction. In this section we will focus on interpreting the efficiency of clustering, the table is an assortment of all the results of k-means and expectation maximization after dimensionality reduction. We see two important outputs first is the raw feature implementation of k means, it has a very healthy db index but we have seen this plot earlier it results in sparse clusters, on the other hand we can see, forward search followed by expectation maximization has a perfect balance of all the parameters, it has very low cluster size variance, meaning there is little sparsity in clusters, and its other benchmark metrics like Inter Cluster Distance and DB Index are in a healthy range as well. Hence it can be concluded that for predicting kick-starter campaign success, if we want to divide our vector space into clustering partitions to aid in prediction of kickstarter campaign success Forward Search plus expectation maximization could be preferred over other algorithms.

Section 4 starts on next page

4 Key Learning Outcomes

- **This project gave the opportunity to explore the dimensionality reduction and unsupervised learning techniques, it is fascinating to learn how unsupervised learning can aid in finding the patterns in the data, even though it is agnostic to the dependent variable. This project also briefly touched the space between supervised and unsupervised realms when we use the cluster outputs are one of the predictors and it drastically improved the performance in form of AUC for energy appliance prediction problem set. It was completely a new phenomenon and it gives a new novel lead to start tackling a supervised learning prediction problem (though it not always guaranteed to work! we understand the caveat but is a thought provoking technique never the less!)**

- **We learnt the importance of choosing the metric in evaluating the cluster results, for instance only selecting inter cluster distance as the metric may give us results that may be allegorically robust but not make a business sense, this project gave us the opportunity to discover more holistic and more comprehensive an broad approach in evaluating the cluster output**

- **Another important takeaway is that not all problems may be designed to be solved using unsupervised learning, for instance, kick starter campaign success prediction is a prediction problem in its true essence with high class imbalance, also reducing dimensionality in this case reduced the predicting power as we probably lost information in reducing dimensionality. The reduction typically aids in improving predictive power but a learner must be aware of the trade offs which can go either way. This was again an important learning takeaway from this project. As we conclude the course we observe most of the nuances of machine learning can be seen and understood from the spectrum of bias-variance trade-off. This phenomenon was even more profoundly evident in this last project!**