UNSUPERVISED LEARNING & DIMENSIONALITY REDUCTION

Bosun Anifowoshe

bosunani@gmail.com

Introduction

This paper presents an analysis on the performance of 2 algorithms for unsupervised learning - K-Means and expectation maximization (EM) and 4 algorithms for dimensionality reduction - principal component analysis (PCA), independent component analysis (ICA), random components analysis (RCA), and random forest classifier (RFC). These learning algorithms were tested on two unique classification problems – Magic Telescope[1] and Phishing Websites[2].

Data Overview

Magic Telescope (MT): This data set was generated by a Monte Carlo program to simulate registration of high energy gamma particles in a ground-based atmospheric Cherenkov gamma telescope using the imaging technique. The binary classification task is to discriminate between patterns caused by primary gammas from those caused by hadronic showers using the properties of the images captured by the telescope.

This data consists of 19,000 observations with 10 *continuous* valued attributes and a binary target attribute. One thing to note with this data is that the target attribute is somewhat unbalanced i.e. 65% of the observation belong to the primary gamma class. This could be attributed to the fact that the data set was simulated by a Monte Carlo program causing some of the hadronic showers class to be underestimated. It will be interesting to see how the various algorithms handle this artifact in the data set.

Phishing Websites (PW): This data was collected by a group of researchers to shed light on the important features that have proved to be sound and effective in predicting phishing websites. The binary classification task is to discriminate if a website is legitimate or not using the properties of the website such as URL length, appearance, etc.

This data consists of 11,000 observations with 30 *discrete* valued attributes and a binary target attribute. In contrast to the Magic Telescope data set, this data is relatively balanced i.e. 56% of the observations belong to illegitimate webpage class. One interesting thing to note with this data is that in addition to the researchers gathering the data, they also added some new features which they think will be important in improving the classification of the phishing websites. The upshots of this data gathering approach are (1) since humans are prone to errors, there is a potential that this data will be prone to noise (2) the added features might improve the predictive power of the model. It will be interesting to see how the various algorithms handle this artifact in the data set. Furthermore, several of the attributes in this data set are categorical with the levels {-1,0,1}. The data is preprocessed to using one-hot encoding to create dummy features with level {0,1}. This further increased the number of discrete attributes that will be used by the algorithms to 46 resulting in this data set having quadruple number of attributes compared to the Magic Telescope data set.

Methodology

All implementation of all the algorithms is performed using Python's scikit-learn module. All of the data will be used for clustering and dimensionality reduction. After running dimensionality reduction, the Phishing Website data will be split into two sets – 70% used for training, 30% for testing to rerun neural network learner on the newly projected data. The performance of k-Means will be evaluated using average within-cluster sum of square errors (SSE), and log probability for EM. In addition, silhouette, homogeneity and, F1 score will be used to evaluate both algorithms. Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) are also used to evaluate EM. For the dimensionality reduction algorithms, we utilized the variance ratio, kurtosis, and mean reconstruction correlation to pick the optimal number of components. Finally, the performance of the neural network algorithm will be based on F1 score, accuracy, AUC, precision, and recall.

Part 1: Unsupervised Learning

This section explores both the k-means and EM algorithms for clustering both datasets. In clustering, we try to group observations together such that the observations which belong to one cluster are similar to each other.

k-Means Clustering

k-Means algorithm clusters the dataset by separating the observations into k clusters of equal variances such that the within cluster sum of squares distance or inertia is minimized. Euclidean distance is preferable in k-Means because other distance functions might not converge. k-Means randomly initializes the cluster centroids, assigns each observation to its nearest centroid, and then assigns a new centroid to each cluster based on chosen observation in that cluster. Since k-Means is not guaranteed to find the global optimum due to the random selection of initial cluster centroids, the algorithm was run 5 times with different initializations and the average metrics over the 5 models are reported.

For this analysis, the optimal number of clusters for each dataset is determined by inspecting the elbow on the SSE plot. In addition, the silhouette score measures how well observations are assigned to its own cluster and how far they are from other clusters while the homogeneity score is used to describe how each cluster contains only observation of a single class. Finally, the F1-score is used to evaluate the accuracy of each cluster since we have the labels of each observation in the dataset.

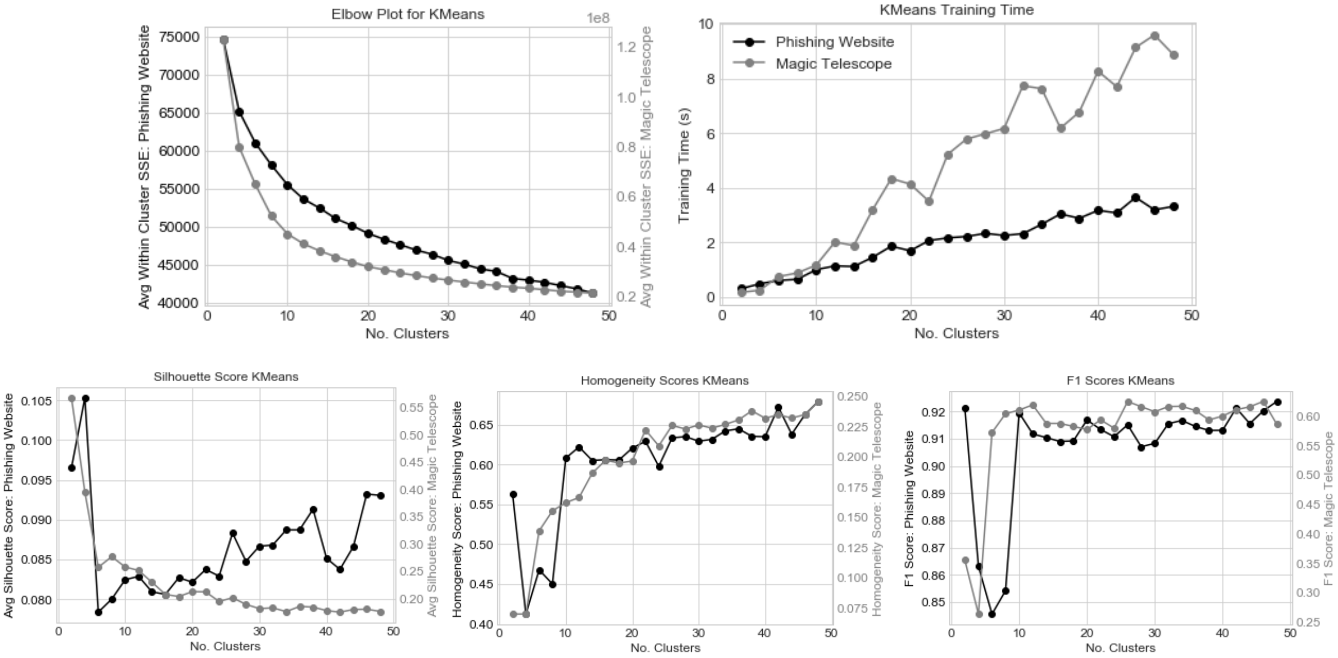
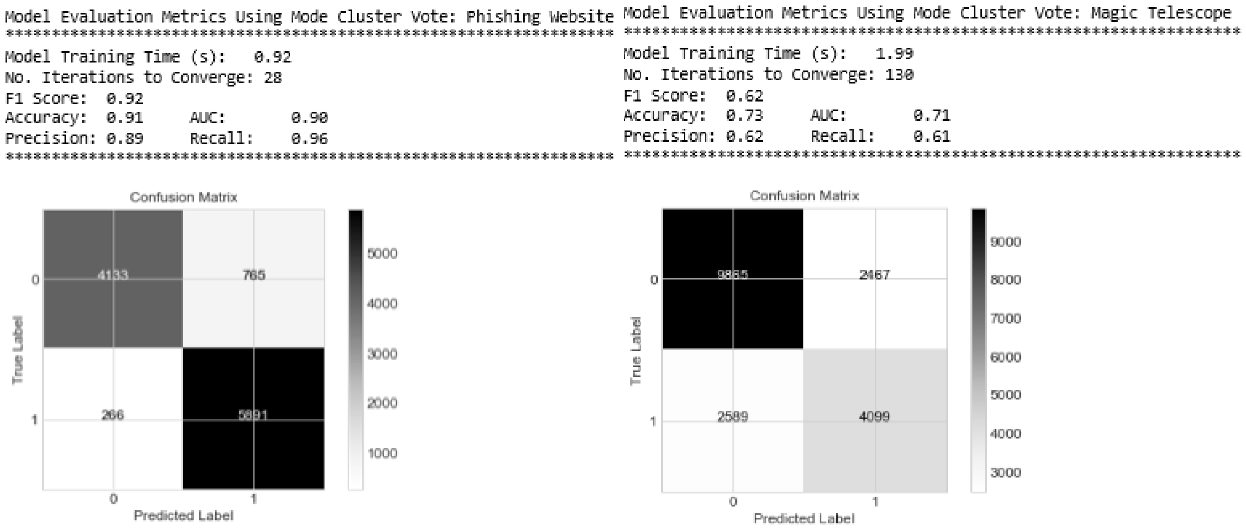


Figure 1: Elbow Plots – Black: Phishing Website Data, Grey: Magic Telescope Data

In figure 1, the black lines show the model complexity curves for the phishing website data while the grey lines represent the magic telescope dataset. As observed, the SSE initially decreases sharply with increase in number of clusters and then the rate of change in SSE decreases as we add more clusters for both datasets. Even though increasing the number of clusters decreases the SSE, the time it takes to train the model increases linearly. If we have a bigger dataset, having more clusters could be very expensive and prone to overfitting. The optimal number of clusters is selected at the elbow of the plot. The elbow is the position where inertia stops decreasing steeply. For the PW data, cluster = 12, and for the MT data, cluster = 10 seems to be the best choice. This observation is corroborated by silhouette, homogeneity, and F1 plots which show the elbow at around the same number of clusters chosen.

Furthermore, the model performance is examined for both data sets by building the model using all dataset. Figure 2 below shows the confusion matrix and summary of the model evaluation metrics.



```
Model Evaluation Metrics Using Mode Cluster Vote: Phishing Website
*********************************************************************
Model Training Time (s):   0.92
No. Iterations to Converge: 28
F1 Score:  0.92
Accuracy:  0.91     AUC:       0.90
Precision: 0.89     Recall:    0.96
*********************************************************************
```

```
Model Evaluation Metrics Using Mode Cluster Vote: Magic Telescope
*********************************************************************
Model Training Time (s):   1.99
No. Iterations to Converge: 130
F1 Score:  0.62
Accuracy:  0.73     AUC:       0.71
Precision: 0.62     Recall:    0.61
*********************************************************************
```

Figure 2: Model Evaluation Metric for – Left: Phishing Website Data, Right: Magic Telescope Data

Comparing both models, the k-Means model performed better on the PW data set compared to the MT data set on all metric – F1 score, accuracy, AUC, precision, and recall even though the PW model has a lower number of clusters. This is potentially due to the fact the PW model uses more discrete attributes and the target attribute is more balanced than the MT data set. In addition, for both data set, the training time is much higher in building the MT model compared to the PW model

Expectation Maximization

In contrast to the k-Means algorithm which assigns each observation to one cluster, the EM algorithm is modeled using a mixture of Gaussian distributions i.e. it enables soft clustering which gives probabilities of an observation belonging to each cluster. EM finds k-distributions of data that maximizes the likelihood of data given the distributions. It alternates between computing an expectation (E) step, which creates a function for the expectation of the log-likelihood evaluated using the current estimate of the parameters, and a maximization (M) step, which computes parameters that maximizes the expected log-likelihood found on the expectation step. The likelihood is often represented as a logarithmic to handle large and small values and may be positive or negative, based on the probability density function. The diagonal covariance type was selected in this implementation.

For this analysis, the optimal number of distributions for each dataset is determined by inspecting the elbow on the log probability plot. In addition, the silhouette score, homogeneity score, F1-score, AIC, and BIC are used to evaluate model performance.

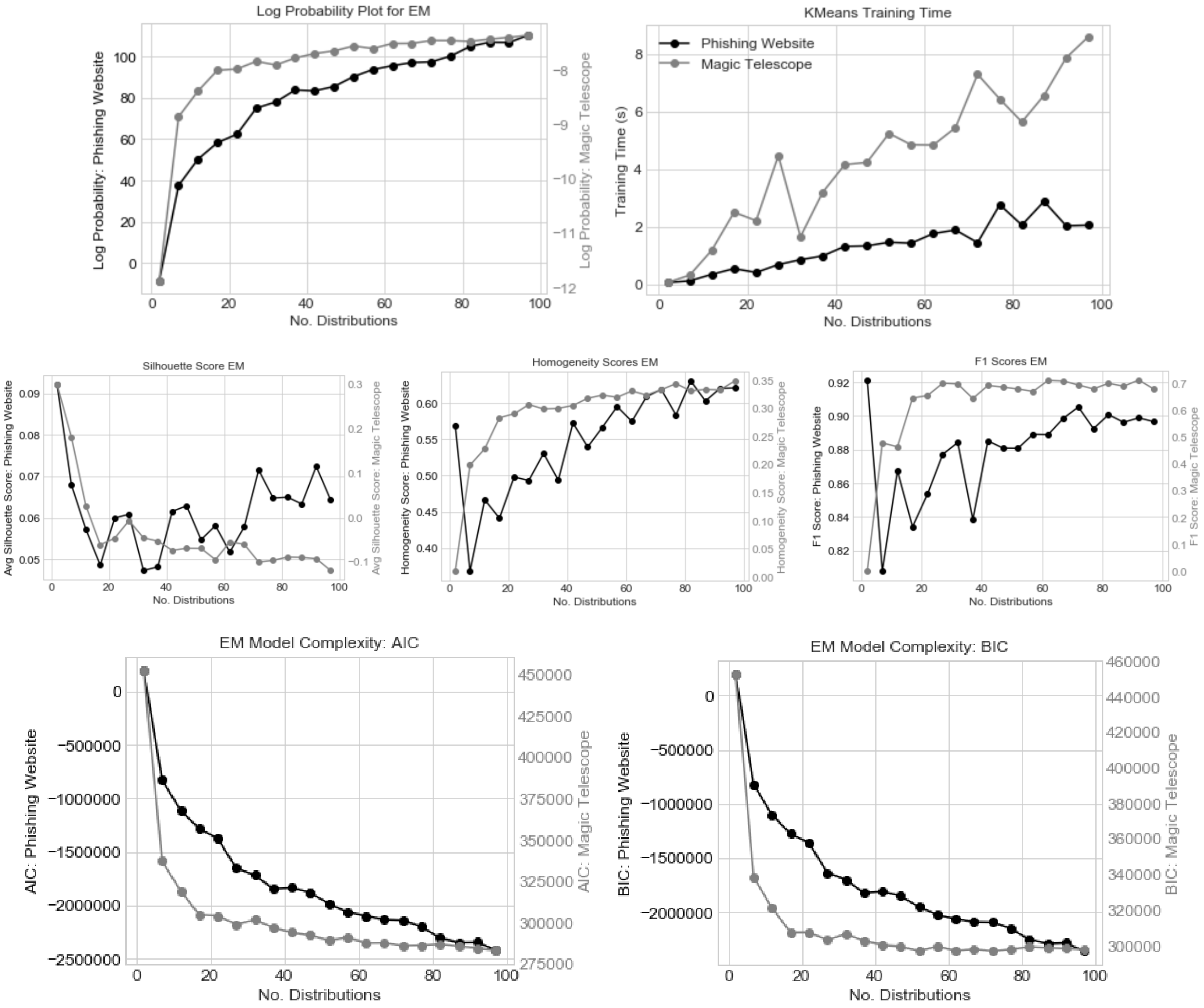
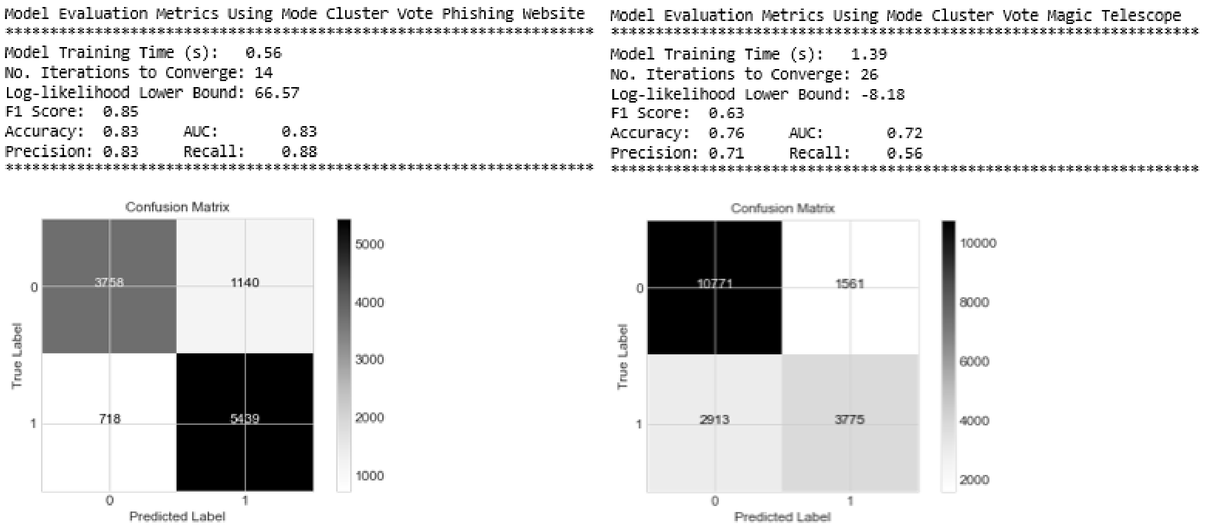


Figure 3: Elbow Plots – Black: Phishing Website Data, Grey: Magic Telescope Data

In figure 3, the black lines show the model complexity curves for the phishing website data while the grey lines represent the magic telescope dataset. As observed, the log probability initially increases sharply with increase in number of distributions and then the rate of change in log probability decreases as we add more distributions for both datasets. Even though increasing the number of clusters increased the log likelihood, the time it takes to train the model increases linearly. If we have a bigger dataset, having more clusters could be very expensive and prone to overfitting. The optimal number of distributions is selected at the elbow of the plot. The elbow is the position where inertia stops increasing steeply. For the PW data, cluster = 24, and for the MT data, cluster = 20 seems to be the best choice. This observation is corroborated by silhouette, homogeneity, F1, AIC, and BIC plots which show the elbow at around the same number of clusters chosen.

Furthermore, the model performance is examined for both data sets by building the model using all dataset. Figure 4 below shows the confusion matrix and summary of the model evaluation metrics.



```
Model Evaluation Metrics Using Mode Cluster Vote Phishing Website
*****************************************************************
Model Training Time (s):   0.56
No. Iterations to Converge: 14
Log-likelihood Lower Bound: 66.57
F1 Score:  0.85
Accuracy:  0.83     AUC:       0.83
Precision: 0.83     Recall:    0.88
*****************************************************************
```

```
Model Evaluation Metrics Using Mode Cluster Vote Magic Telescope
*****************************************************************
Model Training Time (s):   1.39
No. Iterations to Converge: 26
Log-likelihood Lower Bound: -8.18
F1 Score:  0.63
Accuracy:  0.76     AUC:       0.72
Precision: 0.71     Recall:    0.56
*****************************************************************
```

Figure 4: Model Evaluation Metric for – Left: Phishing Website Data, Right: Magic Telescope Data

Comparing both models, the EM model performed better on the PW data set compared to the MT data set on all metric – F1 score, accuracy, AUC, precision, and recall even though the PW model has a lower number of clusters. This is potentially due to the fact the PW model uses more discrete attributes and the target attribute is more balanced than the MT data set. In addition, for both data set, the training time is much higher in building the MT model compared to the PW model.

Part 2: Dimensionality Reduction

This section explores 4 dimensionality reduction techniques to transform the input datasets to fewer dimensions.

Principal Component Analysis

PCA maps the dataset to linear planes and finds the orthogonal eigenvectors that best explain the maximum amount of variance in the data. The principal/first component explains the highest variance and decreases with each successive component. PCA is performed using eigenvalue decomposition which is sensitive to initial relative scaling of the attributes.

Figure 5 show the proportion of total variance in the data explained by each principal component for both datasets. As observed, the cumulative explained variance increases with principal components. Most of the variance is explained by the first few components. Going beyond the 25th and 4th principal component does not yield considerable information in classifying the PW and MT data respectively.

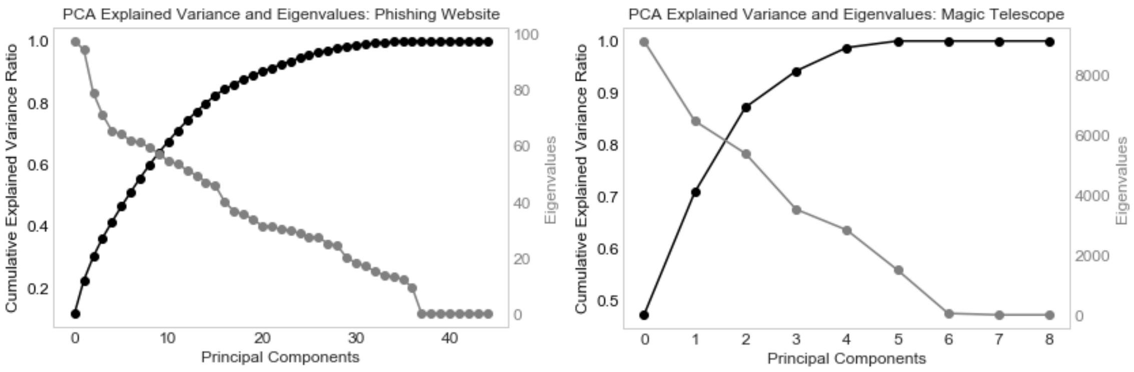


Figure 5: Model Complexity Curve for – Left: Phishing Website Data, Right: Magic Telescope Data

Independent Component Analysis

ICA separates the dataset into additive sub components that are maximally independent. It tries to restructure the input data by maximizing the separation between each of the components from one another and finds basis vectors that are independent components of the original data.

Figure 6 show the distribution of average kurtosis values by each independent component for both datasets. Kurtosis is representation of 4th moment the degrees of the non-Gaussianity of the data, As observed, for the PW data, the average Kurtosis increases with independent components before dropping sharply around the 37th independent component. For the MT data, the average Kurtosis increases with independent components, however a drop in the Kurtosis value is observed at 4th and 7th independent components. By identifying and eliminating Independent components that had hat had insignificant effect i.e. Kurtosis far greater than 3 (normal distribution) and could be considered as noise, we ended up with 36 and 5 independent components for the PW and MT data respectively.

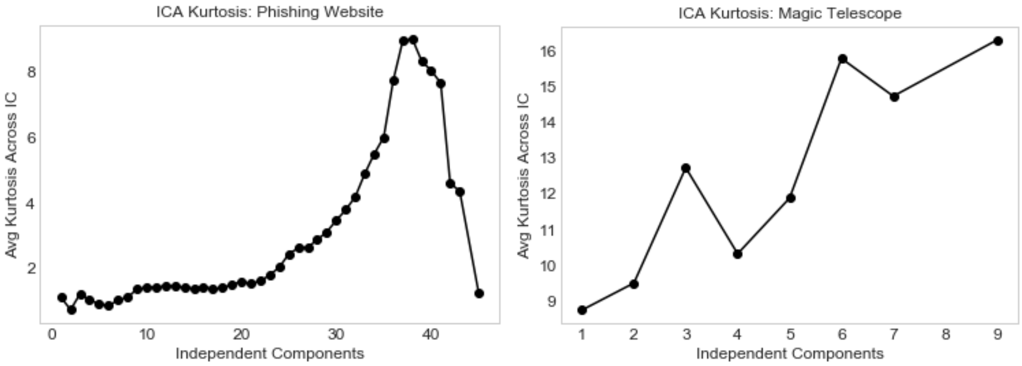


Figure 6: Model Complexity Curve for – Left: Phishing Website Data, Right: Magic Telescope Data

Random Components Analysis

RCA otherwise known as random projections is used to project the original input space "n" on a randomly generated Gaussian matrix "k" such that k << n. Main benefit of using this approach to reduce the number of attributes is to decrease computation cost. However, depending on the random projection, the accuracy of the data may be affected. The optimal number of random components is chosen by inspecting reconstruction error plots in figure 7 below. As observed, the mean reconstruction correlation increases with random components for both datasets. Most of the reconstruction correlation is explained by the first few components. Going beyond the 30th and 7th random component does not yield considerable information in classifying the PW and MT data respectively.

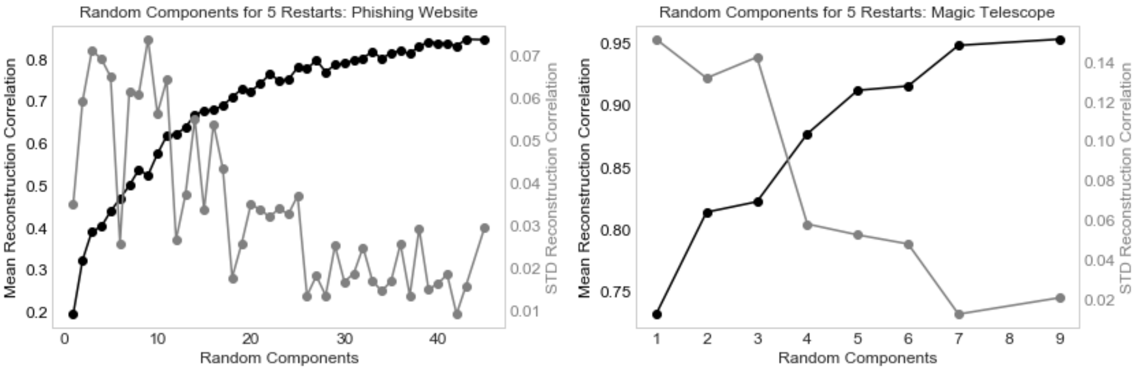


Figure 7: Model Complexity Curve for – Left: Phishing Website Data, Right: Magic Telescope Data

Random Forest Classifier

RCF can be used for feature selection because the tree-based strategies used by random forests naturally ranks features by how well they improve the purity of the node. RCF can be built using information gain based on Gini or entropy impurity measure to split the nodes at the most informative features. Nodes with the greatest decrease in impurity happen at the start of the trees, while nodes with the least decrease in impurity occur at the end of trees. Thus, a subset of the most important features is chosen by pruning trees below a particular node.

Figure 8 shows the feature importance plot where the features are sorted in order of decreasing relative influence. Only the first 10 most important features are shown for the PW dataset. As observed, only the first few features show the greatest relative importance. Going beyond the 8th and 4th features does not yield considerable information gain in classifying the PW and MT data respectively.

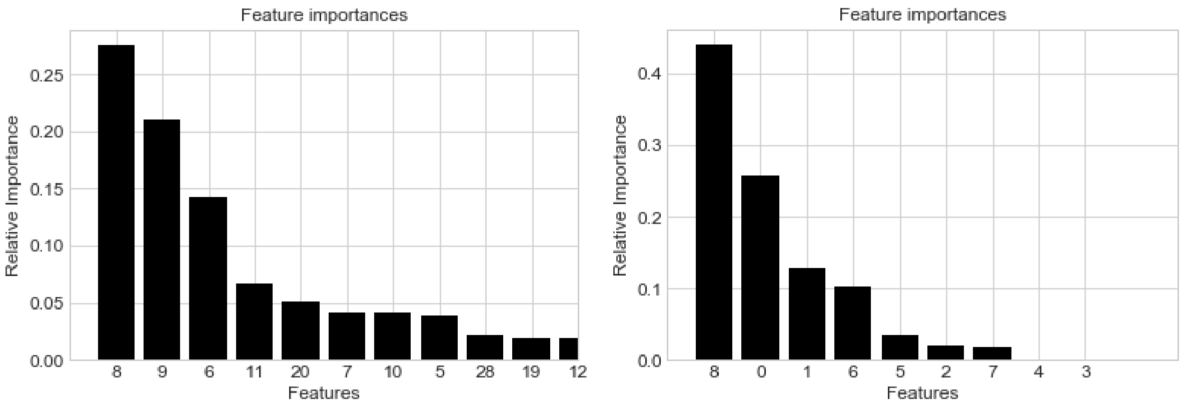
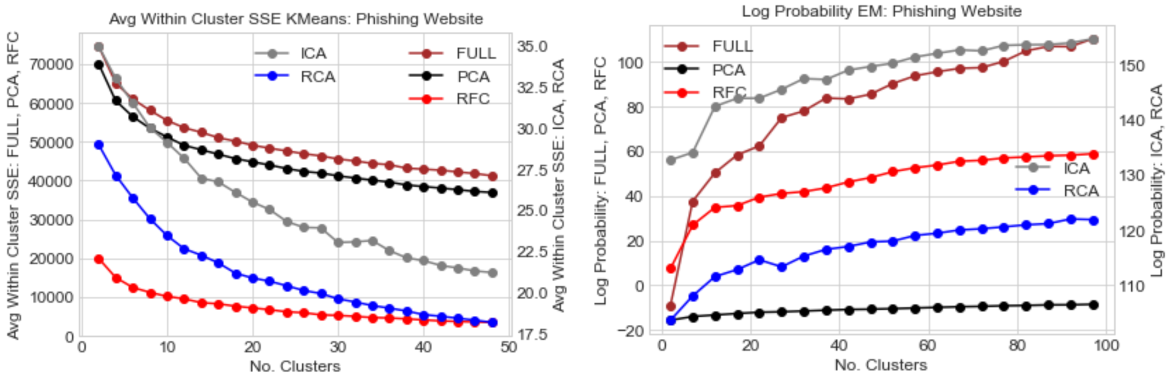


Figure 8: Feature Importance Plots for – Left: Phishing Website Data, Right: Magic Telescope Data

Part 3: Reproduce Clustering Experiments

The k-Means and EM algorithms will then be performed on the reduced dataset and compare its performance against the models with the full data. Figure 9 below shows the performance comparison for the PW dataset. The left plots represent the results for the k-Means algorithm while the right plots are for the EM algorithm. For k-Means, as observed by the average within cluster sum of squared error, SSE initially decreases with increase in number of clusters and then the rate of change in SSE decreases as we add more clusters. The model with the full data has the highest SSE as expected because it uses the highest number of features. ICA shows the lowest SSE followed by RCA, RFC, and PCA in that order. However, RFC shows the lowest computation time while the other algorithms have similar computation times. Based on the combination of accuracy and computation time, RFC will be chosen as the top algorithm for generating features to be used for clustering the PW dataset using k-Means.



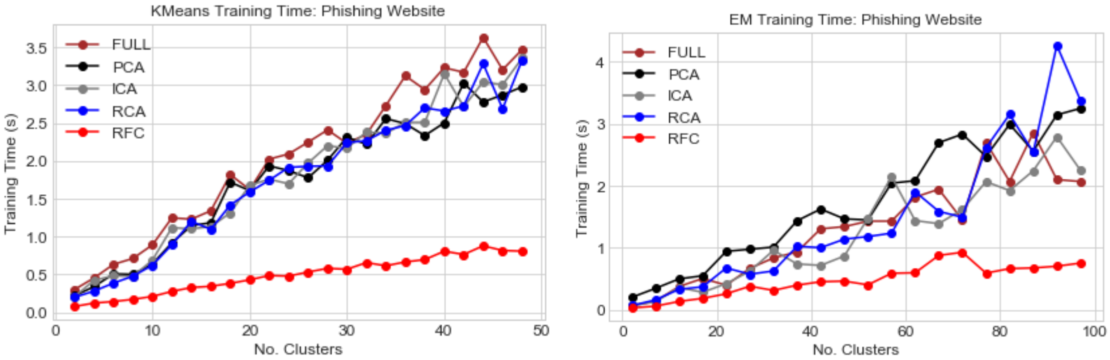


Figure 9: Model Complexity Curve for Phishing Website Data – Left: k-Means, Right: EM

Similarly, ICA shows the highest log probability with increasing clusters followed by RCA, full model, RFC, and PCA respectively. RFC again shows the lowest computation time while the other algorithms have similar computation times. Based on the combination of accuracy and computation time, RFC will be chosen as the top algorithm for generating features to be used for clustering the PW dataset using EM.

Figure 10 below shows the performance comparison for the MT dataset. The left plots represent the results for the k-Means algorithm while the right plots are for the EM algorithm. For k-Means, as observed by the average within cluster sum of squared error, SSE initially decreases with increase in number of clusters and then the rate of change in SSE decreases as we add more clusters. The model with the full data has the highest SSE as expected because it uses the highest number of features. ICA shows the lowest SSE followed by RFC, RCA, and PCA in that order. All the models show similar the computation times. Based on the combination of accuracy and computation time, ICA will be chosen as the top algorithm for generating features to be used for clustering the MT dataset using k-Means.

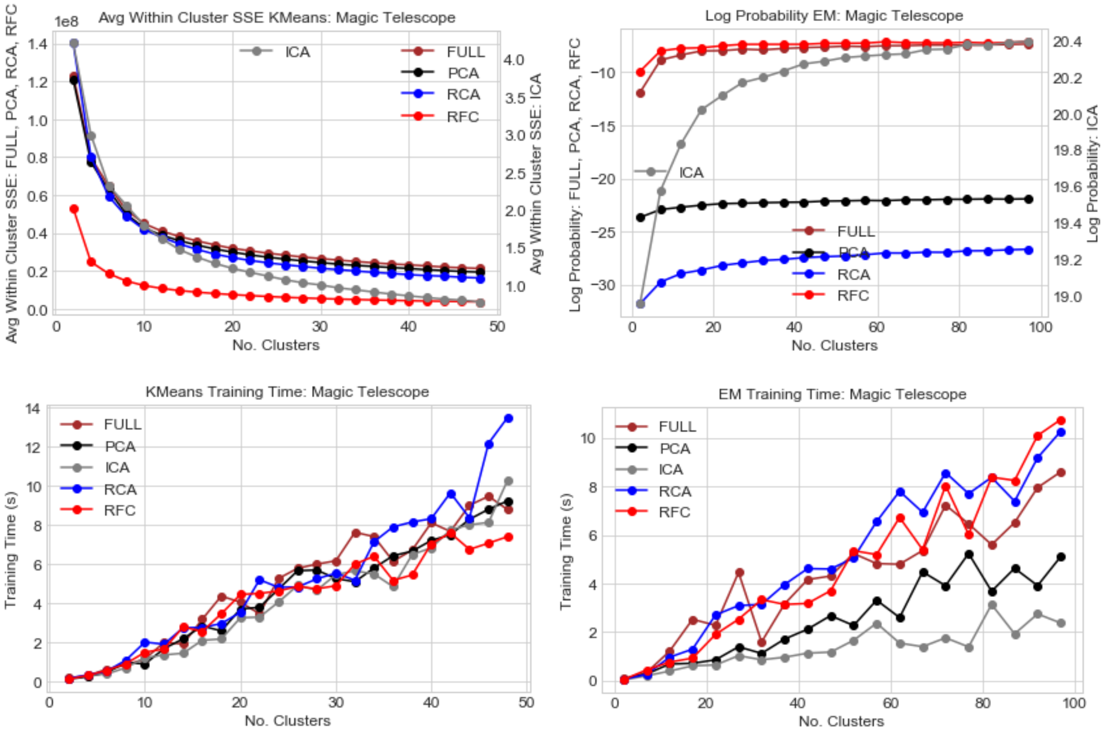


Figure 10: Model Complexity Curve for Magic Telescope Data – Left: k-Means, Right: EM

Similarly, ICA shows the highest log probability with increasing clusters followed by RFC, full model, PCA, and RCA respectively. ICA shows the lowest computation time followed by PCA, while the other algorithms have similar computation times. Based on the combination of accuracy and computation time, ICA will be chosen as the top algorithm for generating features to be used for clustering the MT dataset using EM.

Part 4: Training Neural Network on Projected Data

In this section, the phishing website dataset is used to train a neural network and with four dimensionality reduction algorithms and compare their performance against the models with the full data. The optimal number of components for each algorithm is selected as shown in part 2 above. The multi-layer perceptron back-propagation algorithm with a sigmoid activation function was used to generate the forward feed neural network models for each algorithm. Figure 11 shows the model complexity curves. PCA shows very similar F1 score compared to the full model. ICA and RCA show the next best F1 score while RFC had the worst F1 score although it got better accuracy as the number of training samples increases. In terms of computation time, RFC show the lowest training time initially but as the number of samples increases, the training time increases past ICA and RFC. The full model and PCA show the highest training times however we see that the number of samples increases, the training time of PCA falls below the full model. Table 2 below show the summary of the performance statistics of each model on the untouched testing dataset. Based on the combination of accuracy and computation time, PCA will be chosen as the top algorithm.

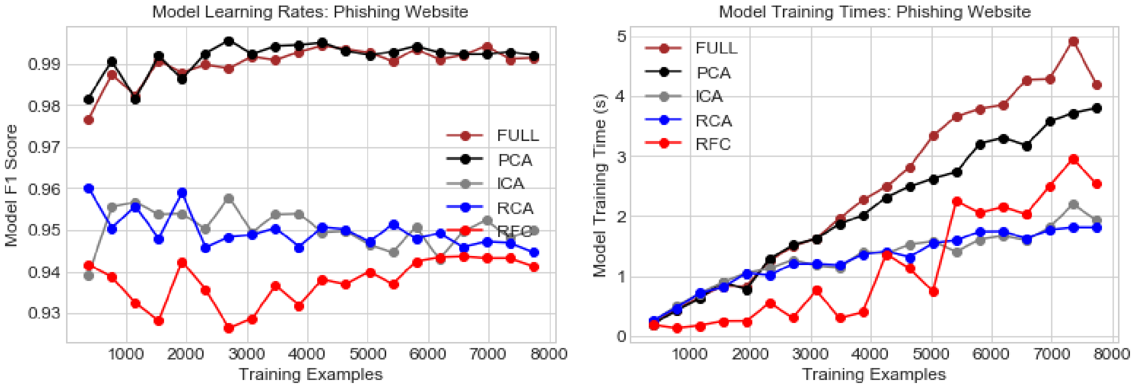


Figure 11: Model Complexity Curve for Phishing Website Data – Left: F1 Score, Right: Computation Time

	FULL	PCA	ICA	RCA	RFC
F1 Score	0.97	0.97	0.94	0.94	0.94
Training Time (s)	2.57676	2.54886	1.00077	1.03656	1.40666
Testing Time (s)	0.00238	0.00287	0.00387	0.00202	0.00275

Table 1: Model Performance Statistics

Part 5: Training Neural Network on Projected Data with Cluster Labels

In this section, we explore how performance of each algorithm changes after adding cluster labels from both k-Means and EM to the reduced datasets. Figure 11 shows the model complexity curves. Overall, this model shows poorer performance in terms of F1 score and computation time compared to the model without the cluster labels as attributes. This is potentially due to the fact that the cluster labels introduced another dimension to the data which made it longer to run and there are some misclassifications in the cluster labels which in turn reduced the F1 score. Overall, the full model showed the best F1 score for this approach. PCA and RFC shows very similar F1 score while the ICA and RCA show the lowest F1 score. In terms of computation time, RFC show the lowest training time while the rest of the models show similar training time. Table 2 below show the summary of the performance statistics of each model on the untouched testing dataset. Based on the combination of accuracy and computation time, RFC will be chosen as the top algorithm.

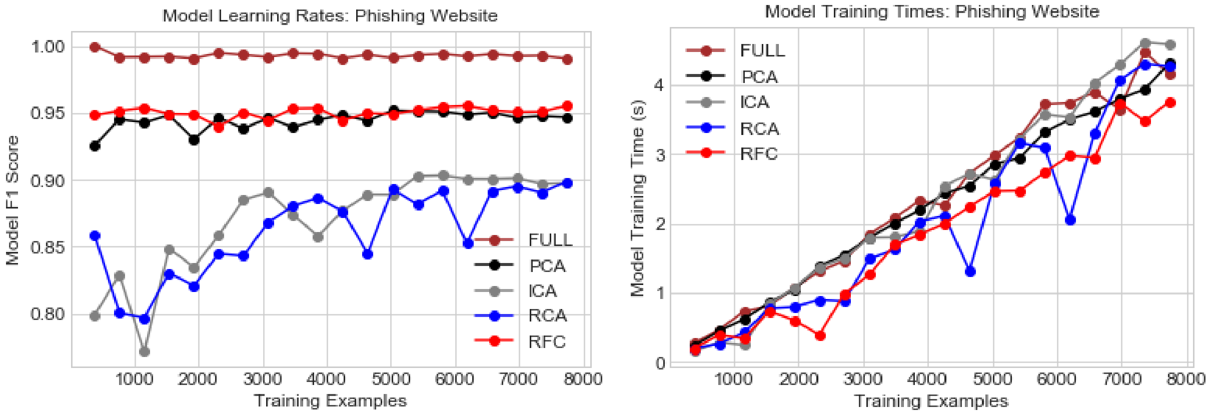


Figure 11: Model Complexity Curve for Phishing Website Data – Left: F1 Score, Right: Computation Time

	FULL	PCA	ICA	RCA	RFC
F1 Score	0.97	0.93	0.9	0.9	0.95
Training Time (s)	3.0158	2.70594	3.01118	2.13749	1.99508
Testing Time (s)	0.00256	0.00199	0.00228	0.00204	0.00281

Table 2: Model Performance Statistics

Conclusion

1. Dimensionality reduction and clustering algorithm to decrease model complexity and improve computation time without impacting the accuracy of the model.

2. For both the PW and MT datasets, it accuracy of the models was rarely improved with the transformations, but there were cases in which accuracy remained virtually the same, while running significantly faster.

3. Based on the combination of accuracy and computation time, RFC performs the best for generating features to be used for clustering the PW dataset using k-Means and EM, and ICA performs the best for generating features to be used for clustering the MT dataset using k-Means and EM.

4. The neural network model on the reduced datasets shows PCA as the top algorithm based on the combination of accuracy and computation time.

5. The neural network model adding cluster labels to the reduced datasets shows poorer performance in terms of F1 score and computation time compared to the model without the cluster labels as attributes.

References

1. Magic Telescope dataset. OpenML Repository
https://www.openml.org/d/1120

2. Phishing Website dataset. OpenML Repository
https://www.openml.org/d/4534

3. Git Repository:
https://github.com/kylewest520/CS-7641---Machine-Learning