Unsupervised Learning and Dimensionality Reduction

Ian Dover

November 9, 2023

1 Datasets

1.1 Auction Dataset

Auction verification is a complex topic which has led to billions in lost revenue. Detecting undesirable auction outcomes is of upmost importance when it comes to auctioning policies. My selected dataset comes from the "UC Irvine Machine Learning Repository" (DOI: 10.24432/C52K6N). The auction dataset can be found here: https://archive.ics.uci.edu/dataset/713/.
This dataset is derived by looking at bid history for a given auction, and then evaluating whether the final outcome of the auction was desirable or not. The duplicates rows in this final dataset were then removed to arrive at a final 2043 rows. This problem can be approached as either a classification or a regression problem, but for the purpose of this project, only the classification target was considered.

1.2 Dropout Dataset

Predicting student dropout is essential to the enactment of mitigation efforts to improve retention rates. If we can predict a student dropout ahead of time, an intervention can be made to increase their likelihood of succeeding. This dataset contains information known on a student when they enroll, such as: past academic performance, anticipated academic path, demographics, and socio-enconomic factors.
Each row in this dataset represents an individual student and their eventual outcome: "graduated", "dropout", or "enrolled". For the purpose of this project, the targets were converted to 0, 1, and 2, respectively.

2 Clustering

2.1 Expected Maximization

Initialization: Expected Maximization begins with an initial guess for the parameters of interest. The central idea is to iteratively refine these parameter estimates.

E-Step (Expectation): In the E-step, given the current estimate of the model parameters, the algorithm calculates the expected value of the likelihood function's complete-data log-likelihood. It computes the expectation of the log-likelihood with respect to the missing data, based on the observed data and the current parameter estimates. Essentially, this step involves "filling in" the missing data using the best guesses based on the observed data and current parameter estimates.

M-Step (Maximization): After the E-step, the M-step updates the parameter estimates by maximizing the expected log-likelihood found in the E-step. This means it finds the parameter values that make the filled-in data most probable. Once the parameters are updated, the algorithm returns to the E-step and the process iterates until convergence, meaning the parameter estimates stop changing significantly between iterations.

Gaussian Mixture Model: The Gaussian Mixture Model (GMM) is a probabilistic model that represents data as a combination of several Gaussian distributions. Each of these Gaussian distributions is called a component, and the model aims to identify the parameters of these components as well as their weights. The Expected-Maximization algorithm is frequently employed to estimate the parameters of a GMM when the component responsible for generating each data point is unknown.

GMM using EM operates in a loop: given current estimates of the Gaussian parameters, determine the "responsibility" of each component for each data point (E-step); then, given these responsibilities, re-estimate the Gaussian parameters to better fit the data (M-step). The process ends when the parameters no longer change significantly between iterations, indicating that the algorithm has likely found a local maximum of the likelihood function.

2.2 K-Means

The k-means algorithm is a widely used clustering technique that aims to partition a set of points into k distinct clusters, where each point belongs to the cluster with the nearest mean. The following explanation breaks down the fundamental steps and logic behind the k-means algorithm:

Initialization: The k-means algorithm starts by selecting k initial centroids, which are the centers of the clusters. There are various methods to choose these initial centroids. In the case of this experiment, the initial k cluster centroids are randomly initialized as k samples in the dataset. Once these centroids are chosen, the algorithm enters an iterative process to adjust these centroids and assign data points to clusters.

Update Step: After all data points are assigned to clusters, the next step is to update the centroids. To do this, the algorithm calculates the mean of all the data points assigned to each cluster and sets this mean as the new centroid. Essentially, each centroid moves to the center of its cluster. Once the centroids have been recalculated, the assignment step is executed again with these new centroids, and the process iterates. The algorithm stops when the centroids no longer change significantly between successive iterations or after a set number of iterations.

The k-means algorithm is powerful, but it can often get stuck in local minima. This means that the final clustering result can depend on the initial choice of centroids.

3 Dimensionality Reduction

3.1 PCA

Principal Component Analysis (PCA) is a statistical method used to reduce the dimensionality of data while retaining as much variance as possible. It's often used for data visualization and noise reduction.

Covariance Matrix Computation: Once the data is standardized, the next step is to compute the covariance matrix. The covariance matrix captures the linear relationships between every pair of features.

Eigen-decomposition: The core of PCA involves finding the eigenvectors of the covariance matrix. These eigenvectors represent the vectors of maximum variance in the data. The eigenvector associated with the largest eigenvalue is the direction of maximum variance, called the first principal component. The eigenvector associated with the second largest eigenvalue is the direction that captures the second most variance and is the second principal component. This continues for as many components as there are features. Each of the eigenvectors are orthogonal to one another.

Projection: Once the principal components are determined, the original data can be projected onto these components to reduce its dimensionality. If you want to reduce your data to k-dimensions, you select the first k principal components and multiply your data by these selected components. This step effectively transforms the original dataset into a new coordinate system defined by the principal components.

Explained Variance: The eigenvalues also give valuable information about the amount of variance "captured" by each principal component. By ranking the eigenvalues from highest to lowest, you can determine the proportion of the total variance in the data that is accounted for by each principal component. This helps in deciding how many principal components to retain. For instance, one might decide to retain enough components to capture 95% of the total variance.

In summary, PCA identifies the vectors/principal components in the dataset that maximize variance. When data is projected onto these vectors, it's transformed into a lower-dimensional space that retains a majority of the explained variance.

3.2 ICA

Independent Component Analysis (ICA) is a computational method for separating a multivariate signal into additive sub-components that are maximally independent from each other. It's often used when you have signals that are mixtures of other unknown signals and you want to figure out the original components.

Signal Separation: ICA starts with the assumption that the signals you have are mixtures of independent non-Gaussian sources. For instance, the sound from multiple people talking doesn't get mixed in a way that one person's words physically change the sound of another's—they just add together. ICA uses statistical properties to untangle these mixtures. It tries to find a way to 'unmix' the signals by making the resulting signals as independent (or different) from each other as possible.

Maximizing Independence: To achieve this separation, ICA looks for a statistical property called independence. Two signals are independent if knowing something about one signal doesn't give you any information about the other. ICA finds a way to filter or transform the mixed signals so that the outputs are as independent as possible. It does this by looking at the statistics of the signals, trying to maximize their non-Gaussianity because independent signals contribute to the overall statistics differently than mixed or Gaussian signals.

ICA is primarily designed to separate a multivariate signal into components that are as statistically independent from each other as possible. When performing ICA, you often end up with as many independent components as there are original features. However, not all of these components are equally important. Some may represent noise, others may carry very little information. By selecting only the most significant independent components based on explain variance, you can reduce the number of independent components you consider.

3.3 Randomized Projections

Randomized projection harnesses the foundational principles of the Johnson-Lindenstrauss lemma to adeptly maintain interpoint distances within a dataset, even in the wake of dimensional contraction. This lemma, a cornerstone in the field of mathematics, assures that when a dataset undergoes transformation through the application of a random matrix characterized by a Gaussian distribution, the original spatial relationships between points are largely conserved. This preservation is contingent upon selecting an adequately substantial projection dimension, denoted as k, which should be in approximate correspondence with the logarithm of the dataset's cardinality.

Creating the Random Projection Matrix:

The generation of the random projection matrix, denoted as R, is a critical step that underpins the dimensionality reduction process. The matrix R is a $k \times d$ matrix, where k represents the new, lower dimension, and d is the dimensionality of the original data.

The importance of using a random matrix stems from:

1. **The Johnson-Lindenstrauss Lemma:** This lemma posits that a random linear projection of high-dimensional data into a lower-dimensional space will, with high probability, preserve the pairwise Euclidean distances between the points.

2. **Distribution of Projections:** By choosing a zero-mean distribution, each new coordinate in the lower-dimensional space is the sum of the original coordinates, each weighted randomly. This sum is a random variable itself, which, by the Central Limit Theorem, approaches a Gaussian distribution as d grows large.

In this transformation, points initially situated in a d-dimensional space are transposed into a k-dimensional space. Due to the projection, the expected distance between any pair of points is scaled by a factor of $\sqrt{d}$. To correct for this scaling, the new projected points are often rescaled by a factor of $\sqrt{1/k}$, ensuring that the expected squared length of the projection of a unit vector remains 1.

This rescaling ensures that the average pairwise distances in the reduced k-dimensional space are faithful representations of the original distances in the d-dimensional space. Hence, the global structure of the dataset, in terms of relative distances, is largely preserved.

3.4 t-SNE

t-SNE (t-distributed Stochastic Neighbor Embedding) is a non-linear technique for dimensionality reduction that is particularly well suited for embedding high-dimensional data into a space of fewer dimensions. t-SNE aims to learn a low-dimensional representation of the data (usually 2D or 3D for visualization purposes) that reflects the similarities in the high-dimensional space.

Pairwise Probability Calculation: For each pair of points in the low-dimensional space, it computes a pairwise probability by using a t-distribution (Cauchy distribution) rather than a Gaussian to compute the probability of neighborhood in the low-dimensional map.

KL divergence: The t-SNE algorithm then tries to make the two probability distributions—over pairs of high-dimensional points and over pairs of low-dimensional points—as similar as possible. This is achieved by minimizing the Kullback–Leibler (KL) divergence between the two distributions using gradient descent. KL divergence is a measure of how one probability distribution diverges from a second, expected probability distribution.

Gradient Descent: Starting with a random initialization, t-SNE moves the points in the low-dimensional space in such a way as to reduce the KL divergence. Points that are similar in the high-dimensional space will get closer in the low-dimensional space, and points that are dissimilar will move apart.

Early Exploitative Optimization: Early in the optimization, the attractive forces between points are artificially magnified to help the space unfold more appropriately. This early exaggeration helps t-SNE to overcome local minima and create a more interpretable map of clusters.

4 Unsupervised Evaluation Metrics

4.1 AIC

The Akaike Information Criterion (AIC) serves as a tool for model selection, particularly after utilizing the Expectation-Maximization (EM) algorithm to estimate model parameters. Given that the EM algorithm can lead to models with a large number of parameters, such as mixture models or models with hidden variables, overfitting can become a concern. Overfitting is characterized by the model capturing noise instead of the underlying data structure.

The AIC addresses this by quantifying the trade-off between the model's complexity and its fit to the data. It is calculated for each model using the formula:

$$AIC = 2k - 2\ln(L) \tag{1}$$

where k denotes the number of estimated parameters within the model, and L represents the model's maximum likelihood function. The term $-2\ln(L)$ decreases as the model better fits the data, which is desired. Conversely, the term $2k$ acts as a penalty for the number of parameters, discouraging unnecessary complexity.

4.2 BIC

The Bayesian Information Criterion (BIC) is a criterion for model selection among a finite set of models, and is particularly useful when applied to models estimated via the Expectation-Maximization (EM) algorithm. The BIC helps to select the model that best balances fit and complexity, especially when comparing models with different numbers of parameters, such as the number of clusters in a mixture model. The BIC penalizes the complexity of the model to prevent overfitting. The formula for calculating BIC is:

$$BIC = -2\ln(\hat{L}) + k\ln(n)$$

where $\ln(\hat{L})$ is the natural logarithm of the likelihood of the model, k is the number of estimated parameters, and n is the number of observations. The model with the lowest BIC is generally preferred.

4.3 Inertia

In unsupervised learning, inertia is a key metric used in clustering algorithms like K-means to quantify the cohesion of clusters by measuring the sum of squared distances between each point and its cluster centroid. The aim is to minimize inertia, thus ensuring that points within a cluster are as close to each other and to their centroid as possible, indicating tight and well-defined clusters. The "elbow method" helps determine the optimal number of clusters by identifying the point at which increasing clusters no longer significantly reduces inertia, suggesting a suitable clustering solution with a balance between cluster tightness and number.

4.4 Average Absolute Kurtosis

Kurtosis measures the "tailedness" of a data distribution, indicating the presence of extreme deviations. For Independent Component Analysis (ICA), which separates a mixed signal into non-Gaussian independent components, kurtosis helps in identifying non-Gaussian features of the distributions. "Average absolute kurtosis" in ICA is the mean of the absolute kurtosis values across all extracted components, ensuring only the magnitude of non-Gaussianity is considered regardless of whether the distribution has heavy (leptokurtic) or light (platykurtic) tails. High average absolute kurtosis suggests effective separation by ICA, as it implies the components have more distinct non-Gaussian characteristics than the original mixed signals.

4.5 Reconstruction Error

Reconstruction error in randomized projection is the discrepancy between the original high-dimensional dataset and its approximation after being projected to a lower-dimensional space and then reconstructed back to the high dimension. During dimensionality reduction, information is lost because fewer dimensions are used to represent data that initially had many dimensions. This process involves using a random matrix to project the data onto a subspace, and then trying to reverse this projection to measure the quality of the dimensionality reduction, with the reconstruction error quantifying the amount of information lost during the process.

4.6 KL Divergence

KL Divergence serves as a statistical measure for quantifying the discrepancy between two probability distributions. Within the framework of t-SNE, these distributions are derived from the pairwise similarities in the original high-dimensional dataset versus those in the reduced low-dimensional representation.

KL Divergence is utilized as a fidelity metric for the low-dimensional mapping, comparing the high-dimensional probability distribution with the one derived from the low-dimensional embedding. If the low-dimensional embedding preserves the neighborhood structure perfectly, the KL Divergence reaches its theoretical minimum value of zero, indicating no information loss between the two distributions.

However, due to the inherent complexities and potential information loss when reducing dimensions, some discrepancies are inevitable. Such discrepancies result in a positive KL Divergence value, providing a quantifiable measure of the loss of neighborhood information. t-SNE employs optimization techniques to adjust the low-dimensional representation, seeking to minimize the KL Divergence and thereby create a representation that is as faithful as possible to the high-dimensional original.

5 Clustering Analysis

Figure (1) presents a compelling visualization of the model selection process using the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC) within the framework of an Expectation-Maximization (EM) clustering algorithm.

Initial Sharp Decrease: The plot illustrates an initial sharp decrease in both AIC and BIC values as the number of clusters increases from 2 to 3. This pronounced decline is indicative of a substantial enhancement in model fit attributable to the introduction of an additional cluster. It suggests that the underlying data are structured into at least two distinct groups, and the EM algorithm's move to bifurcate the data into two clusters allows for a significantly better representation of this intrinsic grouping. The steepness of the drop-off reflects the degree to which a single cluster was insufficient, and it reinforces the notion that the data cannot be reasonably described by a unimodal distribution.

Gradual Decrease After 2 Clusters: Subsequent to this sharp decrease, the plot indicates a more tempered descent in the values of both criteria. This pattern of gradual reduction upon increasing the cluster count beyond two suggests that while additional clusters continue to capture more of the data's complexity, each cluster added beyond the second contributes less to the overall fit of the model. The diminishing returns from additional clusters are manifest in the gentle slope of the curve post the initial decline. Each new cluster represents an incremental gain in capturing the data's variance, yet the relative improvement to the model's explicative ability is less significant.

The search for the "elbow" — the point at which further increases in cluster numbers fail to yield commensurate improvements in the fit — becomes crucial here. It is at this juncture where the increase in model complexity due to additional clusters is not justified by a marked improvement in model fit

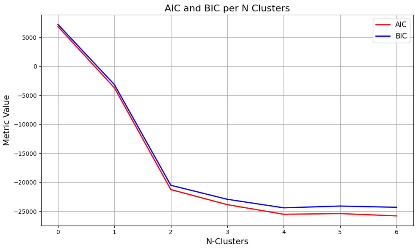


Figure 1: Auction: AIC and BIC per N-Clusters

In Figure (2), we observe the contrasting model selection behaviors exhibited by the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC) as the number of clusters in the dataset increases. Initially, both criteria agree on the improvement of model performance when transitioning from one to two clusters, as evidenced by lower (better) AIC and BIC values. This consensus suggests a substantial gain in the goodness-of-fit that outweighs the penalty for increased model complexity inherent in both criteria.

However, the alignment between AIC and BIC diverges as we progress beyond two clusters. The AIC continues to decline, albeit at a decelerating pace, indicating a preference for models with additional clusters. This pattern suggests that the AIC perceives the incremental improvements in likelihood from additional clusters as sufficient to compensate for the added complexity—thereby advocating for more complex models.

Conversely, the BIC values increase beyond the two-cluster solution, indicating that the incremental improvements in likelihood no longer justify the additional complexity according to the BIC's penalization scheme. The steeper penalty for additional parameters, particularly pronounced due to the larger sample size, favors a simpler model that balances fit and parsimony.

The divergence depicted in Figure (2) exemplifies a fundamental tension in model selection: the trade-off between fitting the data well and maintaining a simpler, more generalizable model. While the AIC may advocate for a more granular clustering, suggesting subtle structures in the data are worth capturing, the BIC cautions against overfitting and points towards a more succinct representation that may generalize better to new data.

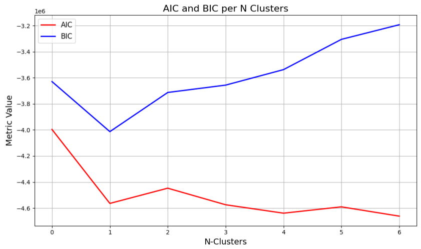


Figure 2: Dropout: AIC and BIC per N-Clusters

Figure (3) illustrates an elbow chart that captures the relationship between the number of clusters and the corresponding inertia, a measure of the internal coherence of the clusters, within a k-means clustering algorithm framework. As depicted, the initial increase in the number of clusters from 1 to 7 results in a steep decrease in inertia, indicating a significant improvement in the homogeneity of the identified clusters. This is attributed to the fact that with few clusters, each encompasses a broad range of the data points, thus the centroids are relatively distant from many of the points in their cluster. Incrementally adding clusters up to 7 allows centroids to align more closely with the data points they represent, dramatically improving the overall fit of the model.

At the critical juncture of 7 clusters, Figure (3) reveals the "elbow," a point where the rate of decrease in inertia markedly slows down. This suggests that the k-means algorithm has reached a level of clustering where adding additional clusters results in a diminishing return with respect to the compactness of the clusters. Prior to this point, the clusters are distinct and the data within each cluster is well-contained, suggesting that the aggregate variance within clusters is high. Past the elbow, additional clusters serve only to marginally enhance the model's accuracy, as they sub-divide the data into increasingly finer groups, thus reducing the inertia slowly.

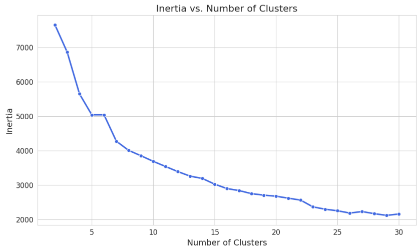


Figure 3: Auction: K-Means Elbow Chart

The above observations about k-means in Figure (3) hold true in Figure (4) as well.

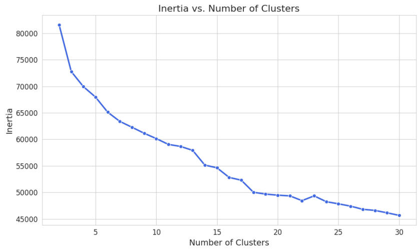


Figure 4: Dropout: K-Means Elbow Chart

Figure (5) illustrates the performance of a binary classification task evaluated by F1 score and Accuracy, using k-means clustering with various distance metrics. Among these, Euclidean distance emerges as the most effective, possibly due to its efficiency in datasets where classes are linearly separable. Manhattan distance, or L1 distance, shows slightly lesser effectiveness, potentially due to its limitations when data points aren't axis-aligned. Cosine similarity, while useful in high-dimensional spaces for its direction-focused measurement, underperforms possibly due to the dataset's nature not aligning with its strengths. Chebyshev distance, focusing on the maximum difference, might not capture subtle but crucial differences across multiple dimensions as effectively in this context.

This variation in metric performance largely depends on how well each aligns with the dataset's characteristics and the distribution of the classes. Euclidean distance's success indicates a dataset with a linear, geometrically interpretable feature space, while the others falter perhaps due to a mismatch with the data's intrinsic qualities. Additionally, it's important to note that the effectiveness of these metrics is also influenced by how they interact with the k-means algorithm's assumptions about the data, such as the shape and distribution of clusters.

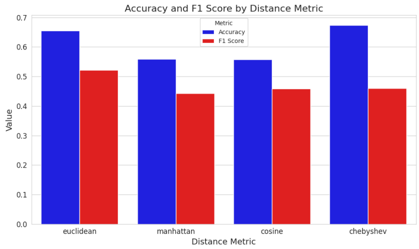


Figure 5: Auction: K-Means Distance Metric VS. Accuracy and F1

In Figure (6), cosine distance outperforming Euclidean distance in terms of F1 score on the Dropout dataset indicates unique characteristics of the dataset and the suitability of the cosine metric. Cosine distance, focusing on the orientation rather than the magnitude of data points, is particularly effective in high-dimensional spaces. This suggests that the dataset may be high-dimensional, with feature patterns (how features are aligned or oriented with each other) being more critical than their absolute magnitudes. The superiority of cosine distance in this context implies that the dataset's features convey more information through their pattern of occurrence rather than through their linear separability, which is what Euclidean distance measures. This characteristic aligns well with the F1 score's sensitivity to correctly identifying true positives and negatives, making cosine distance a more apt choice for this specific dataset and task.

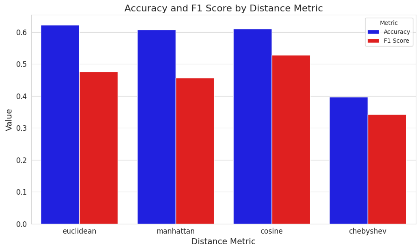


Figure 6: Dropout: K-Means Distance Metric VS. Accuracy and F1

6 Dimensionality Reduction

Figure (7) illustrates the relationship between the number of Principal Component Analysis (PCA) components and the explained variance in a dataset with 6 features. Notably, the graph shows a linear increase in explained variance with the addition of each PCA component, plateauing upon reaching the sixth component. This trend aligns perfectly with the number of features in the dataset, leading to several key observations:

Linear Increase up to Six Components: The initial linear relationship indicates that each of the first six PCA components captures unique variance associated with each of the six features in the dataset. This suggests a significant and independent contribution of each feature to the dataset's overall variance.

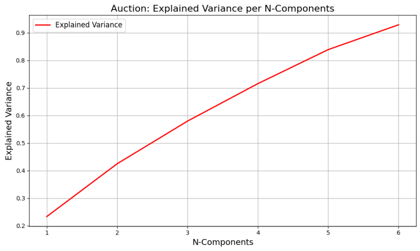


Figure 7: Auction: PCA Explained Variance per N-Components

Figure (8) in our analysis of the dropout dataset reveals a logarithmic relationship between the number of PCA components and the explained variance. This pattern is not only insightful but also indicative of certain intrinsic characteristics of the dataset. In PCA, one usually anticipates a sharp increase in explained variance with the initial components, which then levels off as additional components contribute less and less. This behavior, particularly the logarithmic nature observed, can be interpreted as follows:

Dominance of a Few Features: The initial steep section of the curve highlights the presence of a few dominant features within the dataset. These features are significant in that they account for a substantial portion of the overall variance. The quick ascent in explained variance at the beginning of the curve indicates that the early PCA components are effectively capturing these dominant features. Beyond this point, the incremental gain in explained variance from adding more components begins to diminish.

Incremental Contribution of Remaining Features: The curve's logarithmic profile implies that after accounting for the variance from the primary features, the rest of the features contribute incrementally smaller amounts to the explained variance. This trend is typical in datasets where, beyond several key features, the remainder have progressively less impact on the overall variance.

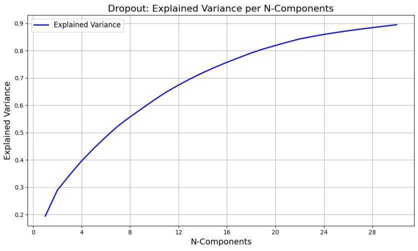


Figure 8: Dropout: PCA Explained Variance per N-Components

In the analysis of Figure (9), the application of Principal Component Analysis (PCA) to project our auction dataset into two dimensions has yielded insightful visual patterns regarding the classification labels. This dimensionality reduction technique, aimed at preserving the most significant variance within the data, reveals distinct distributions for the positive and negative classes in the 2D space. For the positive class, the emergence of clear streaks on the graph is particularly noteworthy. These streaks suggest a strong linear relationship among the features that define this class. In contrast, the negative class predominantly forms a circular distribution with some points interspersed within the streaked areas. This implies a less linearly correlated structure within its defining features, unlike the positive class.

Positive Class with Clear Streaks: The linear patterns observed in the positive class within the PCA plot indicate a significant linear relationship among the dimensions represented. This characteristic suggests that in the multi-dimensional space of the original dataset, certain features that contribute to the positive classification are linearly correlated or follow a specific pattern. This kind of relationship in the data is crucial as it points towards a certain predictability or a set pattern in the variables that lead to a positive auction outcome.

Negative Class Predominately Circular with Some Overlap: The circular distribution pattern observed for the negative class, alongside its partial overlap with the streaked areas of the positive class, suggests a more varied and less predictable set of features defining these data points. This pattern indicates that the features leading to a negative classification might be more independent of each other or follow a less straightforward correlation compared to the positive class. The areas of overlap between the negative and positive classes in the streaks are of particular interest, as they could represent zones of ambiguity or instances where the principal components used in PCA are less effective at distinguishing between the two classes.

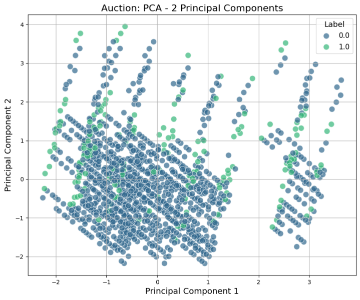


Figure 9: Auction: PCA Scatter Plot

In the analysis of Figure (10), which visualizes the Dropout dataset following a similar methodology used for the Auction dataset, an intriguing pattern emerges from the PCA-based dimensionality reduction. Here, the dataset is effectively projected into two dimensions, revealing a clear "Euclidean" separation between the three classes. This distinct spatial separation in the PCA plot indicates several key characteristics about the underlying structure and nature of the Dropout dataset.

Firstly, the apparent Euclidean separation of the classes suggests a relatively high degree of linear separability among them in the original feature space. In simpler terms, the features that characterize each of the three classes in this dataset exhibit distinct patterns or distributions that PCA is able to capture and differentiate effectively. This level of separation is quite remarkable given that PCA is a linear transformation technique; it implies that the underlying structure of the data is such that the most significant axes of variance (the principal components) align well with the class boundaries.

Additionally, this clear separation indicates that the features relevant to each class are not only distinct but also relatively homogeneous within each class. The homogeneity within classes and heterogeneity between them is a desirable trait for many classification algorithms, as it simplifies the task of distinguishing between classes. For the Dropout dataset, this could mean that students who are likely to drop out have certain characteristics or patterns in their data that are markedly different from those who are not at risk.

Furthermore, the effectiveness of PCA in this scenario suggests that the dimensions with the highest variance in the dataset are closely tied to the class distinctions. This is a valuable insight, as it directs attention to these high-variance dimensions as potentially significant predictors for classifying students based on dropout risk.

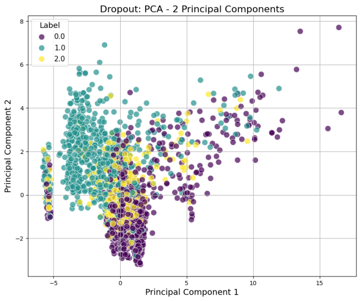


Figure 10: Dropout: PCA Scatter Plot

In Figure (11), the monotonic increase in average absolute kurtosis with more clusters in the Auction dataset suggests a particular type of data structure. High kurtosis typically indicates that there are outliers or a heavy-tailed distribution. In the context of an auction dataset, this could mean that certain independent components (ICs) are capturing extreme bidding behavior or rare but significant auction events.

ICA Behavior: ICA aims to decompose the dataset into components that are as statistically independent from each other as possible. As the number of clusters increases, it appears that ICA is able to increasingly separate out these extreme behaviors or rare events into their own components, thereby increasing the overall kurtosis.

Data Interpretation: The monotonic increase in kurtosis might indicate that the dataset has several layers of structure or distinct patterns, which become more discernible as the number of clusters increases. Each additional cluster may be capturing more nuanced or less frequent auction behaviors, leading to higher kurtosis values.

In Figure (12), the Dropout dataset behaves differently. The decrease in average absolute kurtosis after 12 clusters suggests that initially, the ICA is capturing increasingly distinct or extreme patterns (similar to the Auction dataset), but after a certain point (12 clusters), this trend reverses.

ICA Behavior: This reversal could indicate that beyond 12 clusters, the ICA starts to "overfit" or capture noise as if it were signal. This might be dividing the data into too many components, where the additional components are not capturing meaningful independent structures but rather random variations or noise, which tend to have lower kurtosis.

Data Interpretation: The initial increase followed by a decrease in kurtosis suggests that there is a point of diminishing returns in the number of clusters for the Dropout dataset. It implies that the dataset has a certain level of complexity which is adequately captured with a specific number of clusters (in this case, around 12), beyond which the additional clusters do not contribute to meaningful data separation.

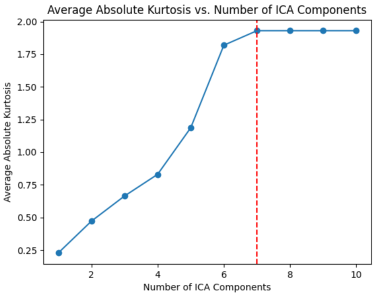


Figure 11: Auction: ICA Explained Variance per N-Components

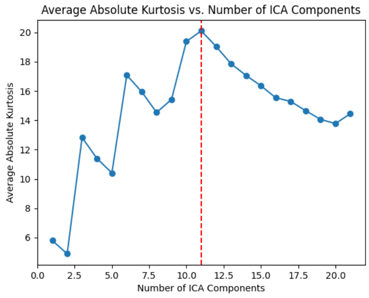


Figure 12: Dropout: ICA Explained Variance per N-Components

In examining the dimensional reductions of the Auction and Dropout datasets, as depicted in Figures (13) and (14) using Independent Component Analysis (ICA), and their counterparts, Figures (9) and (10) using Principal Component Analysis (PCA), a notable observation is the apparent inversion or reflection between the ICA and PCA results. This phenomenon can be primarily attributed to the inherent characteristics of the PCA and ICA algorithms, particularly in how they handle the orientation of the axes in the reduced-dimensional space.

PCA, by design, identifies the directions of maximum variance in the dataset, determining the principal components that capture the most significant structure within the data. These principal components, however, do not have a fixed direction; the sign of the components (positive or negative) can be arbitrary, leading to potential inversions in the plotted dimensions.

Similarly, ICA seeks to decompose the dataset into statistically independent components, a process which inherently allows for the axes in the reduced space to be reflected. The goal of ICA is not to preserve the variance structure but to identify components that are as statistically independent from each other as possible.

When Independent Component Analysis (ICA) and Principal Component Analysis (PCA) yield similar values for the first two components during dimensional reduction of a dataset, it can be indicative of specific characteristics of the data and the alignment of objectives between these two methods. This phenomenon may occur in cases where the directions of maximum variance, which PCA identifies, also coincide with the maximally independent and non-Gaussian components targeted by ICA. This overlap can be a result of the inherent linear independence and non-Gaussian nature of the dominant features within the dataset.

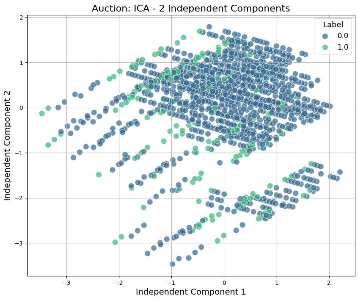


Figure 13: Auction: ICA Scatter Plot

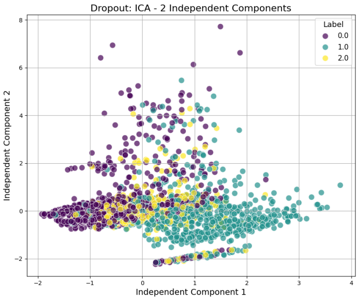


Figure 14: Dropout: ICA Scatter Plot

In the case of the Auction dataset, as observed in Figure (15), Randomized Projection demonstrates an efficient dimensionality reduction with a linearly decreasing reconstruction error. The key observation here is that the reconstruction error reaches 0.0, which is an ideal scenario. This suggests a few points about the Auction dataset:

Inherent Low-Dimensionality: The Auction dataset might inherently reside in a lower-dimensional space despite its higher original dimensionality. This characteristic allows RP to effectively capture the essential information with a relatively small number of components.

Data Structure Compatibility: The linear decrease in reconstruction error and the ability to reach an error of 0.0 indicate that the structure of the Auction dataset is highly compatible with the RP technique. This might imply that the dataset's features have a linear relationship or are otherwise well-suited to linear dimensionality reduction methods.

Effective Random Projection: Achieving a reconstruction error of 0.0 means that the lower-dimensional representation from RP can perfectly reconstruct the original data. This is an ideal outcome, showing that for this particular dataset, the loss of information is negligible even after dimensionality reduction.

In contrast, the Dropout dataset presents a different scenario. According to Figure (16), while the reconstruction error decreases linearly with an increasing number of components, it remains high (around 0.8), which points to several considerations:

Complex or Non-Linear Data Structure: The high reconstruction error suggests that the Dropout dataset has a more complex or non-linear structure that RP struggles to capture. This complexity might be due to intricate relationships between features that are not preserved well under linear transformations.

Dimensionality Reduction Limits: Even with a number of components greater than the original dataset, if the reconstruction error remains high, it indicates a fundamental limit of what linear dimensionality reduction (such as RP) can achieve for this dataset. This could hint at the necessity of exploring non-linear dimensionality reduction techniques or more sophisticated methods that can better handle the data's complexity.

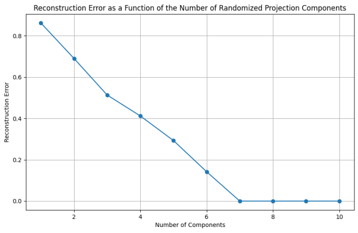


Figure 15: Auction: RP Explained Variance per N-Components

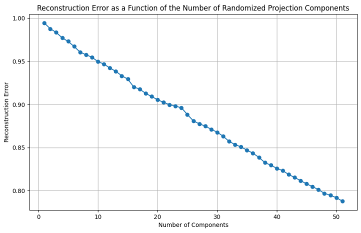


Figure 16: Dropout: RP Explained Variance per N-Components

In our analysis of Figures 17 and 18, we observed distinct behaviors of Randomized Projection (RP), Principal Component Analysis (PCA), and Independent Component Analysis (ICA) in terms of class separation in a two-dimensional feature space. RP's performance was notably inferior in delineating clear class boundaries. This can be attributed to its inherent randomness and lack of optimization for specific data structures, particularly for class separation. In contrast, PCA, by maximizing variance, demonstrated a robust ability to distinguish between classes. This suggests that in our dataset, significant class distinctions are closely aligned with the directions of maximum variance. ICA, focusing on statistical independence rather than just variance, also showed a strong performance in class separation. This implies that the underlying classes in our dataset may be influenced by independent sources. These observations underscore the importance of choosing the appropriate dimensionality reduction technique based on the specific characteristics and requirements of the dataset. While PCA and ICA are particularly effective when class separability aligns with their respective objectives of variance maximization and independence, RP's utility might be limited in scenarios where class differentiation is a primary concern.

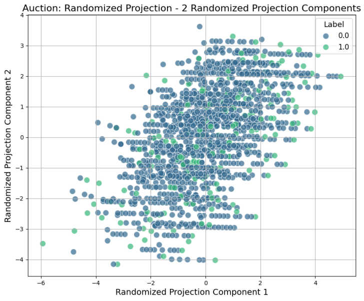


Figure 17: Auction: RP Scatter Plot

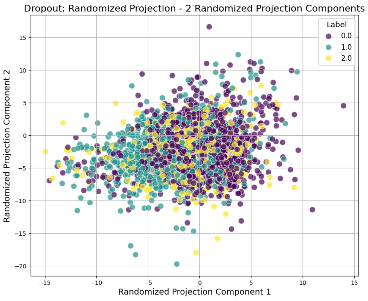


Figure 18: Dropout: RP Scatter Plot

The comparison across Figures (19) to (22) offers a striking illustration of how different dimensionality reduction techniques can vary in their effectiveness depending on the dataset and analysis objectives. t-SNE, as seen in these figures, excels in revealing complex, local patterns within high-dimensional datasets. However, its performance in visualizing clear class separation using only 2 components appears less effective compared to PCA and ICA. This discrepancy arises from t-SNE's focus on non-linear, local relationships and its sensitivity to hyperparameters, which may compromise its ability to accurately represent broader data structures.

In contrast, PCA and ICA, being linear methods, demonstrate a stronger capability in these figures to preserve global structures. Their effectiveness is especially apparent in scenarios where class distinctions are linearly separable and aligned with the principal or independent components of the data.

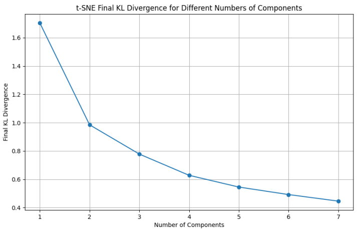


Figure 19: Auction: t-SNE Explained Variance per N-Components

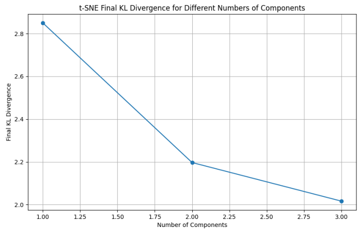


Figure 20: Dropout: t-SNE Explained Variance per N-Components

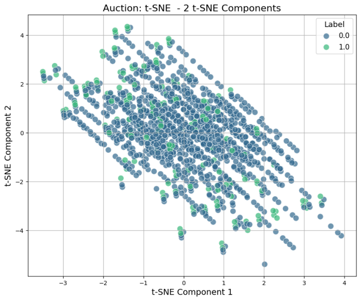


Figure 21: Auction: t-SNE Scatter Plot

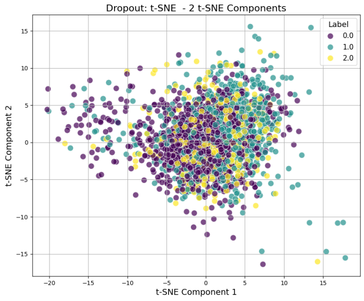


Figure 22: Dropout: t-SNE Scatter Plot

7 Combined Dimensionality Reduction and Clustering

7.1 K-Means and PCA

In Figures (23)-(26), the application of Principal Component Analysis (PCA) before clustering, such as K-means, is a common practice in data analysis. PCA is a dimensionality reduction technique that transforms the data into a new coordinate system with the axes (principal components) representing the directions of maximum variance in the data. Ideally, this transformation can enhance clustering performance by reducing noise and highlighting structures in the data. However, there are several reasons why PCA might not lead to significant improvements in clustering performance:

Inherent Data Structure: If the original data already has well-separated clusters in the original space, and these clusters are aligned along the original axes rather than the principal components, PCA might not only fail to enhance clustering performance but could potentially degrade it.

Loss of Relevant Information: PCA aims to keep the components that explain the most variance, which does not always equate to retaining the components most relevant for clustering. In reducing dimensions, PCA might discard information that is crucial for effectively distinguishing between clusters.

The figures presented here offer a fascinating insight into the dynamics of applying K-means clustering to the Auction dataset, particularly after the implementation of Principal Component Analysis (PCA) for dimensionality reduction. They provide a visual and quantitative narrative of how the process influences clustering outcomes, reflected in various performance metrics.

Contrasting Metric Responses: Another striking aspect of the figures is the apparent increase in average performance across all distance metrics despite the noted decline in F1 and accuracy. This implies that while the PCA transformation may not align perfectly with the actual class boundaries, it does create a data environment where standard distance metrics used in K-means clustering (such as Euclidean distance) operate more effectively. The enhanced average performance could be reflecting improved cluster cohesion or separation, an outcome that doesn't necessarily translate to accurate class label assignments.

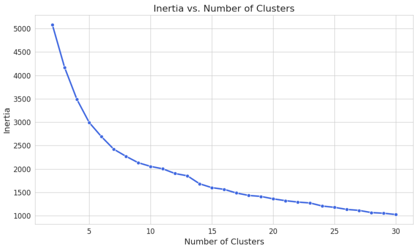


Figure 23: Auction: K-Means Elbow Graph after PCA

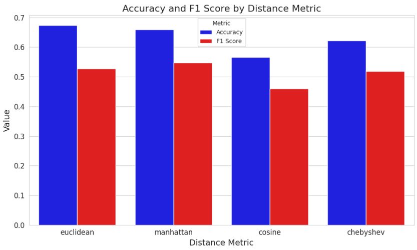


Figure 24: Auction: K-Means Distance Metric VS. Accuracy and F1 Score

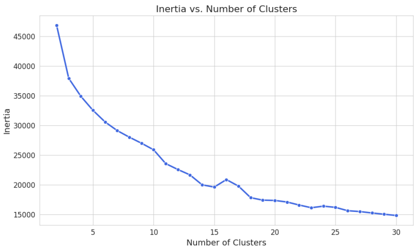


Figure 25: Dropout: K-Means Elbow Graph after PCA

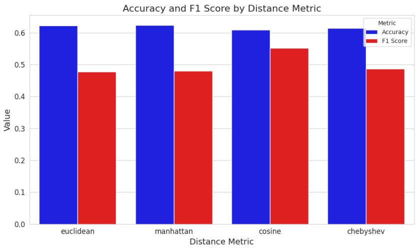


Figure 26: Auction: K-Means Distance Metric VS. Accuracy and F1 Score

7.2 K-Means and t-SNE

Applying t-SNE (t-Distributed Stochastic Neighbor Embedding) before K-means clustering, as demonstrated in Figures (27) and (28) using the Dropout dataset, led to an interesting shift in performance metrics and cluster inertia. t-SNE, renowned for its capability in capturing complex local structures in high-dimensional data, often at the expense of preserving global relationships, reduces the Dropout dataset to three dimensions. This transformation, while visually insightful, can significantly alter the data's landscape. In the context of K-means clustering, which heavily relies on distance measures like Euclidean distance, the reshaped data space may no longer align well with the original class labels. The decreased performance in accuracy and F1 score against the true class labels, as shown in the figures, suggests that the clusters formed post-t-SNE transformation do not correspond as closely to the actual classes as they did in the original high-dimensional space. This misalignment is a potential outcome of t-SNE's focus on local data structures, which, while excellent for visualization and identifying subgroups within data, can distort inter-class distances and boundaries crucial for effective K-means clustering.

Furthermore, the noted drastic reduction in inertia - a measure of within-cluster variance - post-t-SNE application is revealing. Inertia is inherently sensitive to the spatial distribution of data points within clusters. Since t-SNE effectively contracts dense regions and expands sparser ones to preserve local structures, it naturally leads to tighter, more compact clusters in the transformed space. This compactness is reflected in the reduced inertia, indicating closer proximity of data points within each cluster. However, this does not necessarily translate to an improved alignment with true class labels. The lower inertia signifies a strong internal cluster cohesion but doesn't account for the accuracy of these clusters in representing the actual classes. In essence, while t-SNE helps K-means form tighter clusters, it does not guarantee that these clusters accurately capture the inherent class divisions of the original data.

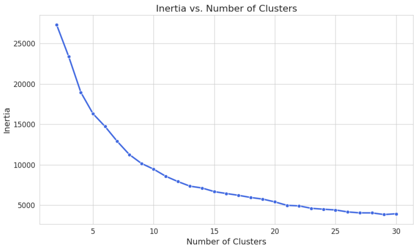


Figure 27: Auction: K-Means Elbow Graph after t_sne

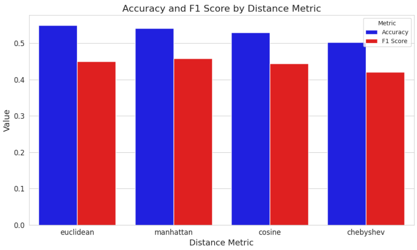


Figure 28: Auction: K-Means Distance Metric VS. Accuracy and F1 Score

8 Neural Network with Dimensionality Reduction Pre-Processing

Figure (29) presents an intriguing case where our neural network exhibits slightly superior performance on the validation dataset compared to the training dataset. This pattern is unusual in typical machine learning scenarios and suggests a few key considerations:

Underfitting in Training: The network seems to underfit the training data. It's plausible that due to either its architectural simplicity or insufficient training epochs, the model fails to capture the complexities within the training set, leading to better generalization on the validation set.

Effect of Randomized Projection: The application of randomized projection, reducing data to four components, likely introduces a regularization effect. This dimensionality reduction might be smoothing out training data noise, inadvertently enhancing model generalization, as reflected in the validation performance.

Dataset Characteristics: Differences in complexity or size between the training and validation sets could also contribute to this phenomenon. If the validation set is less complex or not perfectly representative of the training data, it might inherently be easier to predict.

Random Variance: Considering the stochastic elements in both the randomized projection and neural network training, the observed difference could partly be due to random variance, particularly if the validation dataset size is relatively smaller.

However, in appears that in Figure (30), the training loss slightly outperforms the validation loss, suggesting that the model may be slightly overfitting the training set or the random variance of the dataset permits that the training sample outperforms the validation sample (sampling error).

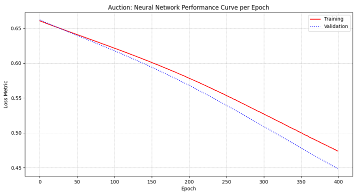


Figure 29: Auction: Randomized Projection Neural Network Performance Curves

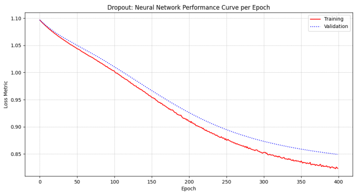


Figure 30: Dropout: Randomized Projection Neural Network Performance Curves

In examining Figure (31), a notable observation is the consistently lower neural network loss on the Auction validation set as compared to the training set. This is an unusual pattern, as standard expectations lean towards better performance on the training set due to model tuning specifically on this data. Several hypotheses and considerations arise from this observation:

Sampling Bias: A primary consideration is the potential for sampling bias. This occurs when the validation set does not accurately reflect the complexity or diversity of the training set. If the validation set includes simpler or less varied examples, the model may erroneously appear to perform better on it. This can result from a non-random or skewed division of data into training and validation subsets.

Size of the Validation Set: The size disparity between the training and validation sets could also be a factor. A smaller validation set might not adequately represent the dataset's diversity, leading to an artificially low loss. This is particularly crucial when the validation set is substantially smaller than the training set.

In Figure (32), the observed overfitting in the neural network post 100 epochs can be partly attributed to the preprocessing of the dataset with t-SNE before training. Here are the key considerations:

t-SNE's Impact on Data Structure: t-SNE, as a nonlinear dimensionality reduction technique, focuses on local structures. This might lead to the loss of critical information and global data patterns, making the neural network susceptible to overfitting on these local, potentially noise-heavy structures.

Model Learning Artificial Structures: The network might be overfitting to the specific structures and distributions created by t-SNE, which do not necessarily represent the real-world data accurately.

Challenges in Generalization: Given t-SNE's emphasis on local neighbor relationships, the neural network could struggle to learn global patterns, contributing to overfitting.

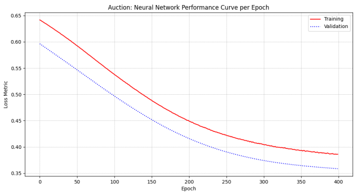


Figure 31: Auction: t-SNE Neural Network Performance Curves

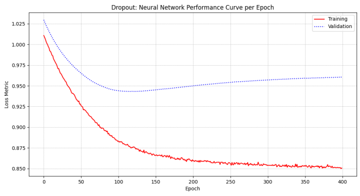
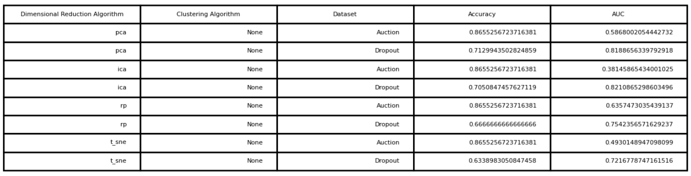


Figure 32: Dropout: t-SNE Neural Network Performance Curves



Dimensional Reduction Algorithm	Clustering Algorithm	Dataset	Accuracy	AUC
pca	None	Auction	0.8655256723716381	0.5868002054442732
pca	None	Dropout	0.7129943502824859	0.8188656339792918
ica	None	Auction	0.8655256723716381	0.38145865434001025
ica	None	Dropout	0.7050847457627119	0.8210865298603496
rp	None	Auction	0.8655256723716381	0.6357473035439137
rp	None	Dropout	0.6666666666666666	0.7542356571629237
t_sne	None	Auction	0.8655256723716381	0.4930148947098099
t_sne	None	Dropout	0.6338983050847458	0.7216778747161516

Figure 33: Dimensionality Reduction Algorithm VS. Performance Table

9 Neural Network with Clustering Pre-Processing

Figure (34) presents a fascinating case study of the consequences of dimensionality reduction through k-means preprocessing in machine learning, specifically when dealing with complex datasets. The figure illustrates a notable shift in the model's performance trajectory post the 100th epoch, signaling a marked onset of overfitting. This outcome is particularly intriguing as it underscores the nuanced balance between dimensionality reduction for computational efficiency and the preservation of essential data characteristics for effective learning.

The k-means preprocessing step adopted in this scenario reduced the dataset to a single dimension represented by 10 cluster values. This transformation, while streamlining the data, seems to have extracted a significant amount of variance and complexity inherent in the original dataset. K-means, by its nature, tends to group data into clusters based on proximity, often leading to a loss of subtler, yet critical, data variations. These nuances, while possibly perceived as 'noise' in some contexts, can be vital for a model to generalize well to unseen data.

In the case of Figure 34, the initial epochs show promising learning trends. However, post the 100th epoch, the model's increasing overfitting indicates that it has begun to learn the noise in the training set – a direct consequence of the reduced complexity of the data. This scenario is a classic example of a model latching onto spurious correlations that do not generalize outside the training set, a common risk in overfitting scenarios.

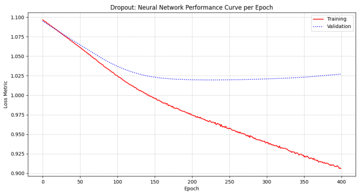


Figure 34: Dropout: K-Means Neural Network Performance

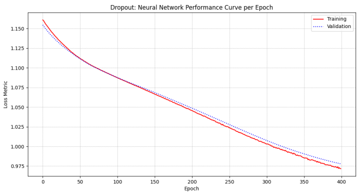
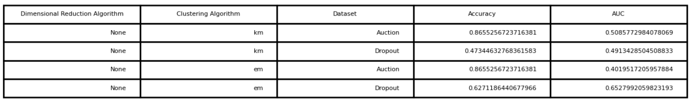


Figure 35: Dropout: Expected Maximization Neural Network Performance



Dimensional Reduction Algorithm	Clustering Algorithm	Dataset	Accuracy	AUC
None	km	Auction	0.8655256723716381	0.5085772984078069
None	km	Dropout	0.47344632768361583	0.4913428504508833
None	em	Auction	0.8655256723716381	0.4019517205957884
None	em	Dropout	0.6271186440677966	0.6527992059823193

Figure 36: Clustering Algorithm VS. Performance Table

10 Conclusion

Figure (1) demonstrates the model selection process in clustering analysis using the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC) applied to an Expectation-Maximization (EM) algorithm. Initially, a sharp decrease in both AIC and BIC values is observed when increasing the number of clusters from 2 to 3, indicating a significant improvement in model fit with the addition of an extra cluster. This suggests that the data inherently consists of at least two distinct groups, and the EM algorithm's ability to partition the data into two clusters results in a much better representation of this inherent structure. Following this initial decrease, a more gradual decline in the values of both criteria occurs as more clusters are added. This suggests that while additional clusters continue to provide more detailed data representation, the marginal improvement in model fit decreases with each additional cluster. This phase emphasizes the importance of identifying the "elbow point," where adding more clusters no longer yields substantial improvements in model fit.

Figures (2) and (3) offer insights into different aspects of model selection and clustering performance. In Figure (2), the AIC and BIC criteria initially concur on the improvement in model performance with an increase to two clusters but later diverge. AIC continues to favor models with more clusters, suggesting that the additional complexity is offset by improvements in likelihood. In contrast, BIC suggests a simpler model, increasing beyond the two-cluster solution, implying that the added complexity is not justified by the likelihood improvement. This divergence highlights the trade-off between data fit and model simplicity. Figure (3) shows an elbow chart for a k-means clustering algorithm. A significant decrease in inertia, indicating improved cluster homogeneity, is observed with an increase in clusters up to 7. Beyond this "elbow point," the rate of decrease in inertia slows, suggesting diminishing returns from additional clusters. This indicates an optimal clustering level where further subdivision of data into more clusters only marginally enhances the model's accuracy.

This study delves into the intricate structures of the Auction and Dropout datasets using advanced dimensionality reduction techniques such as Principal Component Analysis (PCA), Independent Component Analysis (ICA), and Randomized Projection (RP). These methods have enabled a deeper comprehension of the datasets, showcasing their strengths and limitations in extracting meaningful information from complex data. PCA's application to the Dropout dataset, for instance, reveals a distinct Euclidean separation among classes, highlighting its efficacy in identifying linear separability and aligning principal components with class boundaries. This clarity not only offers academic intrigue but also practical benefits for developing more effective classification algorithms in predictive models.

Conversely, the analysis of the Auction dataset via ICA and RP paints a different picture. It exhibits a complex structure with outliers and heavy-tailed distributions, as indicated by the monotonic increase in kurtosis and the dataset's adaptability to lower-dimensional spaces efficiently captured by RP. However, when the Dropout dataset is subjected to similar analysis, it shows a contrasting behavior, with an initial increase in kurtosis followed by a decrease, suggesting overfitting and resistance to linear dimensionality reduction. The persistent high reconstruction error with RP further underscores this complexity. Additionally, the varying results from ICA and PCA across both datasets underscore the significance of choosing the appropriate method based on dataset characteristics. This study, therefore, not only illuminates the practical use of statistical techniques in unraveling the structures within complex datasets but also highlights the crucial aspect of method selection, providing valuable insights for data analysis across diverse fields.

In our analysis, the behavior of Randomized Projection (RP), Principal Component Analysis (PCA), and Independent Component Analysis (ICA) in two-dimensional feature space showed notable differences, especially in class separation. RP's performance in class delineation was relatively inferior, attributed to its randomness and non-optimization for specific data structures. In contrast, PCA, by focusing on maximizing variance, demonstrated a robust capacity for class distinction, indicating that significant class separations in our dataset aligned with the directions of maximum variance. ICA, with its focus on statistical independence, also performed strongly in class separation, suggesting the influence of independent sources on the underlying classes. This highlights the importance of selecting the right dimensionality reduction technique based on dataset characteristics and requirements, especially when class differentiation is a priority.

The application of PCA before K-means clustering, commonly practiced in data analysis, can have mixed results. While PCA is effective in reducing noise and highlighting data structures by transforming data along the directions of maximum variance, it may not always improve clustering performance. This limitation arises if the original data has well-separated clusters not aligned with the principal components or if PCA removes information critical for distinguishing clusters. Figures (23)-(26) show that although PCA can enhance average performance across distance metrics used in K-means clustering, it doesn't necessarily translate to accurate class label assignments. Furthermore, using t-SNE before K-means clustering, as shown in Figures (27) and (28), can significantly change data landscape, often reducing cluster inertia but not guaranteeing alignment with true class labels due to t-SNE's focus on local structures. These insights emphasize the nuanced relationship between dimensionality reduction techniques, data structure, and clustering performance.