CS-7641 Machine Learning:
Assignment 3, Unsupervised Learning and Dimensionality Reduction

Pavel Ponomarev
pavponn@gatech.edu

1 INTRODUCTION

In this project, we [1] study the performance of clustering and dimensionality reduction algorithms on two datasets. We first look into how these methods perform in an unsupervised setting and then how their outputs might support better solutions for supervised problems.

The clustering algorithms we cover in this work are *K-Means* and *Expectation Maximisation (EM)*. The K-Means algorithm is trying to separates data points into k distinct, non-overlapping clusters. It iteratively assigns each data point to the nearest cluster centroid, then recalculates the centroids based on the updated assignments. At the same time, Expectation Maximisation is a soft clustering algorithm that models underlying distribution of data using mixture models (more specifically, we use *Gaussian Mixture Model (GMM)*). It iteratively estimates the parameters of the mixture model, maximizing the likelihood of the observed data. We also use four dimenesionality reduction algorithms: *Principal Component Analysis (PCA)*, *Independent Component Analysis (ICA)*, *Random Projections (RP)*, and *Isomap*. The former three techniques are linear. PCA is aiming to transforms the original features into a new set of orthogonal axes, while retaining the maximum variance in the data. ICA is extracting underlying independent non-Gaussian components that feature space is considered to be constructed from. RP, as its name suggests, uses random projections to reduce data dimensionality. The latter method, Isomap, is a non-linear dimensionality reduction that seeks a lower-dimensional embedding which maintains geodesic distances between all points.

2 PRELIMINARY

2.1 Datasets

The Wine Quality dataset comprises two subsets: one with 1599 samples of red wine and the other with 4898 samples of white wine. Commonly used in a supervised setting, the classification task involves predicting wine quality, which falls into 7 classes ranging from 3 to 9. Notably, class distribution is non-uniform, with more samples falling within the quality range of 5 to 7 than in the ranges of $3-4$ or $8-9$. There are 11 real-valued features and one categorical feature, which represents the *color* of wine. After conducting preliminary data analysis, we were not able to find strongly correlated features. However, we noted that difference in feature distribution for wines with different quality labels is unclear (Figure 1a), while the difference in feature distributions in pairplots for different wine colors is more easy to notice (Figure 1c).

The Breast Cancer is a relatively small dataset, consisting of only 569 samples. At the same time, it has 30 features. The labels usually used in supervised setting are binary, indicating the diagnosis: malignant or benign. 30 real-valued features represent 10 different estimations (like perimeter, smoothness, fractal dimensions) with their 3 statistics: mean, error, worst. Similarly to the former dataset, all features were standardised. After conducting preliminary data analysis, we noticed that there is a sufficient number of correlated (or even dependent) features (please refer to Figure 1c).

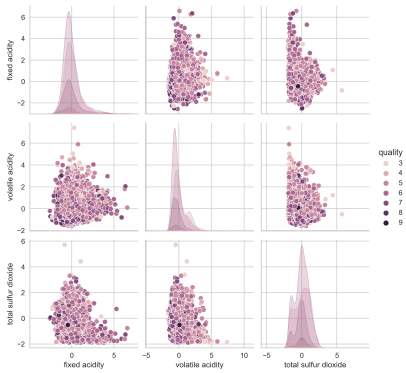
2.2 Hypotheses

For the Wine Quality dataset we hypothesise that clusters found will be far from matching the existing labels given that distribution is non-trivial (Figure 1a) and number of possible target classes. If anything, we believe clusters will more closely align with wine color than with quality label. We believe that dimensionality reduction algorithms should be able to bring down the number of features given (i) the natural relation of some of the features (e.g., fixed acidity/volatile acidity vs pH); (ii) noticeable dominant components in pairplots with much higher variance.

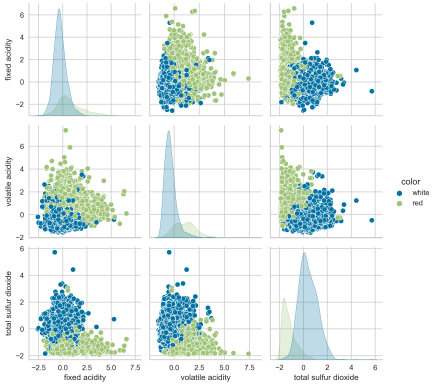
For the Breast Cancer dataset, we expect a better matching of clusters with actual labels than for the Wine Quality dataset, since the number of target classes is only 2. We also expect that dimensionality reduction algorithms to heavily decrease the number of features without a big effect on performance given noticeable high correlation between some of the features, as well as the fact that in fact we only compute 10 actual measures per sample, providing its mean, error and worst estimations results which might be dependent or correlated.

---

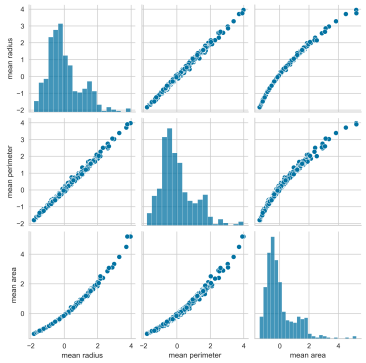
1 The work described in this report is completed fully independently by one sole author (Pavel Ponomarev). The pronounce "we" is used to follow academic writing style.



(a) Feature distribution pairplots with split by quality label in Wine Quality dataset



(b) Feature distribution pairplots with split by wine color in Wine Quality dataset



(c) Example of highly-correlated features in Breast Cancer dataset

Figure 1—Insights from exploratory feature analysis for Wine Quality and Breast Cancer datasets

3 METHODOLOGY

3.1 Metrics

To determine the optimal number of clusters, we utilize metrics such as *distortion score* for K-Means and *Akaike/Bayesian Information Criterion (AIC/BIC)* for Expectation Maximization. The distortion score quantifies the average squared distance between data points and their centroids in K-Means. AIC and BIC strike a balance between clustering likelihood and model complexity in EM. For K-Means, we identify the best cluster count by locating the distortion score's elbow point, while for EM, we select the lowest cluster count where either AIC or BIC is minimised. While we also consider silhouette scores (that measures how well an object fits within its own cluster compared to other clusters), they do not influence our decision-making process.

To evaluate the quality of the obtained clusters, we use metrics like the *Rand Index* (measuring the correctness of pair of sample placements in clusters, similar to classification accuracy), the *Fowlkes–Mallows Index* (combining pairwise precision and recall), and the *V-Measure*, which is harmonic mean of homogeneity (how much clusters consist solely of one class) and completeness (to which extent all data points of a class are in the same cluster).

To assess the results of dimensionality reduction algorithms, we employ metrics such as *explained variance ratio* for PCA, *absolute mean kurtosis* for ICA, and *projection error* for RP and Isomap. We determine the number of components needed for PCA to achieve a 0.9 explained variance threshold, use the elbow method for ICA's absolute mean kurtosis and Isomap's reconstruction error, and select the minimum number of components to reduce reconstruction error below 0.2 for RP.

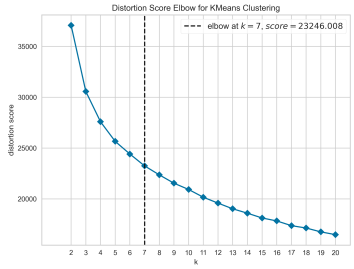
3.2 Experiment Setup

Before starting the experiment, each dataset is split into training and testing sets. Throughout the experiment, we mainly operate on training dataset and only use testing dataset in the last steps to evaluate final performance of the built supervised models. We then perform one-hot encoding of categorical features and apply standard scaling to the features, using mean and variance parameters estimated solely from the training set. This normalization is applied consistently to both the training and testing sets.

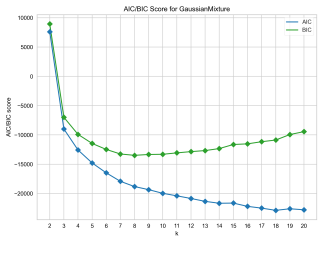
Our formal way to proceed with the experiments as follows. In **Step 1**, we apply K-Means and Expectation Maximisation (GMM) clustering algorithms to the training datasets, choosing the number of clusters and evaluating them via metrics described in Section 3.1. Then, in **Step 2**, we proceed with application of dimensionality reduction techniques. We find the best number of components via metrics covered in Section 3.1. We then evaluate new features using a set of simple classifiers comparing them to original datasets. This is followed by **Step 3**, where we apply clustering algorithms to the outputs of Step 2. We choose and evaluate clusters in the same manner as in Step 1. In **Step 4**, we use features produced by Isomap and PCA methods during Step 2 on Wine Quality dataset and then retrain a Neural Network using, comparing performance with a model trained for Assignment 1. Finally, during **Step 5**, we additionally use cluster features to train a neural network and compare it with the other results.

4 RESULTS OVERVIEW

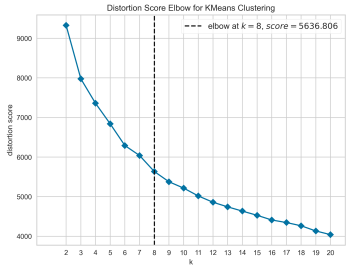
4.1 Step 1: Applying clustering algorithms



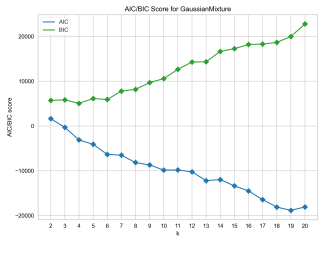
(a) Wine Quality dataset: K-Means, distortion vs number of clusters



(b) Wine Quality dataset: Expectation Maximization, AIC/BIC vs number of clusters



(c) Breast Cancer dataset: K-Means, distortion vs number of clusters



(d) Breast Cancer dataset: Expectation Maximization, AIC/BIC vs number of clusters

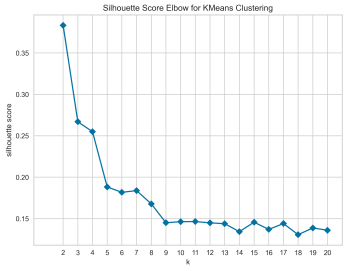
Figure 2—Number of clusters using K-Means and Expectation Maximisation algorithms

On the Wine Quality dataset for K-Means and EM (GMM) clustering algorithms we identified number of optimal clusters to be 7 (Figure 2a) and 8 (Figure 2b) respectively using the criteria described in Section 3.1, which matches well the actual number of classes (7), which intuitively is a good thing. Results of quantitative evaluation depicted in Table 2a (*No reduction*), suggest that we cannot easily say which clustering option is better given similar RI, VM and FMI scores. The absolute results of these scores make intuitive sense. One could notice that V-Measure has a very low absolute value, which is justifiable given its definition: as number of classes is relativelybig, it is unlikely that each cluster will contain mostly samples of one class and simultaneously that one class will be in the same cluster. Continuing this discussion, we can see that VM scores are very similar to the ones we used to get for F1-macro score for this dataset in the Assignment 1. Same can be said about RI and F1-micro score, as well as about FMI and weighted F1 score. One could argue that these metrics are analogous in some sense given their similar definition but in different settings, for example F1-Macro score is essentially accuracy and Rand Index is its analogue in unsupervised space.

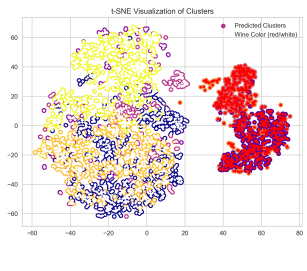
Moreover, upon comparing clusters, it becomes evident that they bear a greater resemblance to each other than to the original labels, as reflected by their RI of 0.891, VM of 0.497, and FMI of 0.482. This observation may be attributed to the inherent similarities between K-Means and GMM clustering algorithms, with the former representing a special case of the latter. What's particularly intriguing is that clusters derived from both K-Means and GMM exhibit a closer affinity to wine color features, as indicated by the higher V-Measure (0.461 and 0.465 respectively) and FMI (0.521 and 0.532 respectively), in comparison to wine quality. This trend may be attributed to the underlying computation of the VM and FMI metrics. It's worth noting that had we opted to utilize silhouette scores for determining the number of clusters, in both cases, the ideal choice would have been $k = 2$ (refer to Figure 3a). This tendency may arise from the highly distinct separation of clusters based on wine colors. This observation gains further support from the visualization of the resulting clusters, depicted using t-SNE alongside their respective wine colors in Figure 3b. This finding aligns with one of the hypotheses posited in Section 2.2.

For the Breast Cancer dataset, both clustering algorithms yield an optimal number of clusters that deviates from the original classes. GMM suggests 4 clusters (Figure 2d), with $k = 2$ BIC value being a close contender, possibly loosing due to random variation. In contrast, K-Means suggests an unexpectedly high number of 8 clusters (Figure 2c). Given our knowledge of the ground truth labels, it appears that employing the elbow method on the distortion score may not be the best approach for this specific problem. In fact, it seems unlikely to yield just 2 clusters due to elbow method. As illustrated in Figure 3c, utilizing silhouette scores would align the number of clusters with the ground truth classes. The clusters closely resemble the original classes, especially with GMM due to the optimal number of clusters aligning with the ground truth classes. Visualization using t-SNE (Figure 3d) shows a preference for more clusters, likely because they are not well-separated, favoring maximizing log-likelihood or minimizing distances to centroids over simplicity.

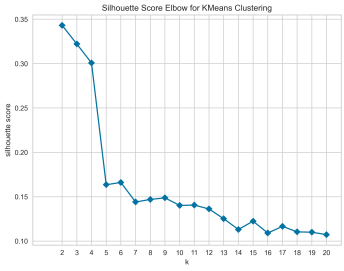
Our hypothesis that clusters would align more closely with the original classes in the second dataset compared to the first is partially confirmed. While this alignment is reflected in the metrics, it's worth noting that the number of clusters generated for the Breast Cancer dataset is notably higher than the number of classes, especially when compared to the Wine Quality dataset.



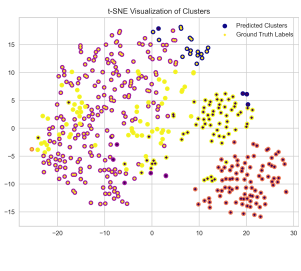
(a) Wine Quality dataset, silhouette scores for K-Means clustering



(b) Wine Quality dataset, identified clusters (via EM) vs colour, t-SNE visualisation

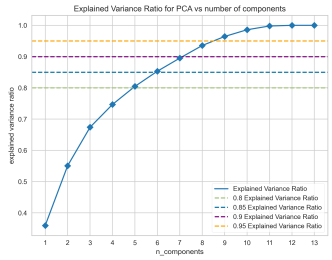


(c) Breast Cancer dataset, silhouette scores for K-Means clustering

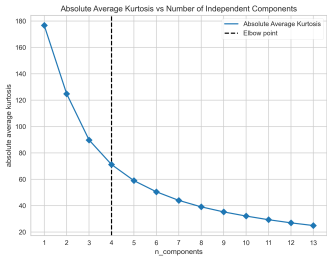


(d) Breast Cancer dataset: identified clusters (via EM) vs GT labels, t-SNE visualisation

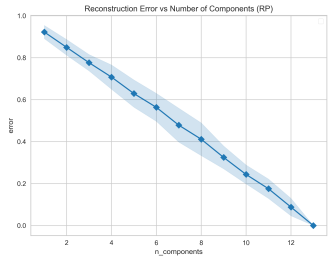
Figure 3—Step 1: other insights



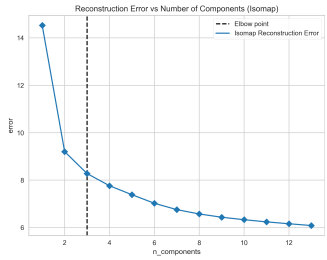
(a) PCA: Explained variance ratio



(b) ICA, Absolute average kurtosis



(c) RP, Reconstruction error



(d) Isomap, Reconstruction error

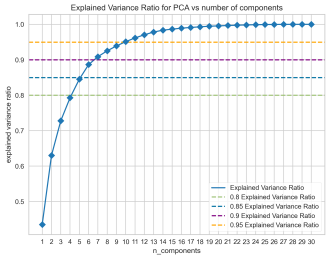
Figure 4—Wine Quality dataset: Number of components for dimensionality reduction algorithms

4.2 Step 2: Applying dimensionality reduction techniques

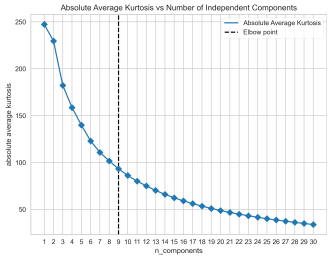
We summarise choice of the optimal number of components based on metrics described in Section 3.1 for Wine Quality dataset in Figure 4 and for Breast Cancer dataset in Figure 5.

After applying PCA to both datasets, we can conclude that we would still be able to explain more than 90% of the variance in the datasets by only relying on 8 out of 13 and 7 out of 30 features for the Wine Quality and the Breast Cancer datasets respectively. Looking at the Figure 4a, we can clearly notice that the last 3 components do not account for the explained variance at all. These 3 components correspond to very small eigenvalues of 0.156, 0.0277 and 2.75×10^{-33}. This, in fact, suggests a high collinearity of the 3 features and, even more, that one of them is completely useless as the variance it explains is almost 0. The latter makes sense given that we have one-hot encoded feature (color) that is essentially represented by two, but one can always be reconstructed from the other one. Four more components have eigenvalues smaller than 0.5, suggesting some collinearity in data. From Figure 5a, it is clear that in Breast Cancer dataset about 9 components do not contribute to the explained variance at all and in fact their eigenvalues are smaller than $1.5 * 10^{-2}$. Even more, another 16 components account only for 10% of the variance. This follows our observation about amount of highly correlated and linearly dependent features that we pointed out in Section 2 in this dataset. Simultaneously, our evaluation summarised in Table 1 shows that performance of the classifiers trained on reduced feature space are almost identical (or sometimes even better) to the baseline for the Breast Cancer dataset and at most 4% worse for the Wine Quality.

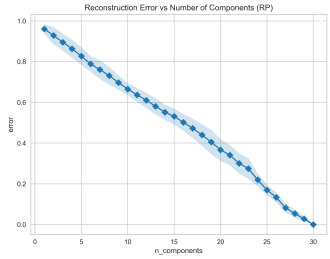
For ICA, we ran FastICA in sklearn with N_COMPONENTS set as number of original features, calculated kurtosis for every component, sorted and plotted absolute mean for each number of components. We note that kurtosis for components distributed almost exponentially for both datasets, i.e., the absolute values are 176.7, 72.9, 19.7, 15.4, 10.1, ... for the Wine Quality dataset and 247.4, 211.9, 87.8, 87.4, 65.3, 38.6, ... for the Breast Cancer dataset. To visualise this, one can look at absolute average kurtosis charts (Figure 4b and Figure 5b). The plots align with theory: more components we have, smaller the absolute mean kurtosis, higher chance the underlying independent components are Gaussian. Based on these plots we chose number of independent components as 4 and 9 for the first and second dataset respectively. Note that in both cases, it essentially means reducing number of features about 3 times. It should not be surprising for the Breast Cancer given that we noted obvious dependency and correlation between some of the features. At the same time, this confirms our previous hypothesis that some of the features in the Wine Quality dataset are dependent, even though it was not obvious from the pairplot analysis. From the conducted evaluation summarised in Table 1, we can



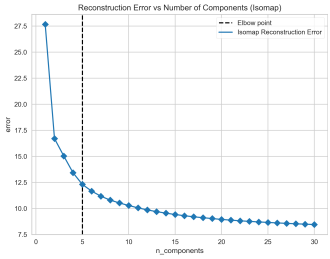
(a) PCA, Explained variance ratio



(b) ICA, Absolute average kurtosis



(c) RP: Reconstruction error



(d) Isomap, Reconstruction error

Figure 5—Breast Cancer dataset: Number of components for dimensionality reduction algorithms

notice that while features extracted by ICA do make sense: indeed, results are still high enough, they are essentially worse than ones built by PCA. At the same time, for the Wine Quality dataset, we can argue that relative reduction in feature number is much higher, so this might give us some other pluses: like training time speed up, etc.

When working with randomised projections, we noticed that reconstruction error linearly decrease with the increase of the number of components used (Figures 4c and 5c). One can notice a small variance in the result, but overall we can note that it did not affect the overall direction of the reconstruction error: according to the plot the variance is not big. Reasonably, we can see that the variance depends a lot on the initial size of the feature space: larger feature spaces have smaller variance. Also, from the charts one can see thatfor both datasets most of the variation happen when number of random components the space is projected contains from $\frac{1}{3}$ to $\frac{2}{3}$ of original features. Generally, this seems reasonable given that components the feature space is projected on are chosen randomly and assuming uniform choice of the components and lack of any other "smart" selection of components, we believe that linearity of the projected error and higher variance for medium number of features is reasonable. Indeed, when you can project on more components the chance leftover features can affect the result is lower and similarly when you want to project on low number of components, it is very unlikely that choice of features used can seriously decrease or increase error. Using plots, we identified the number of components required to maintain reconstruction error lower than 0.2 as 10 and 25 for the Wine Quality dataset and the Breast Cancer. According to Table 2a, the classifiers trained on data transformed by these RPs do comparatively well to PCA(n_comp=8) in case of the Wine Quality dataset (even though we chose more components than in PCA, they are random and not necessarily with the highest variance, which we could see as equalizing factor). Interestingly though, applying RP(n_comp=25) yields better performance than any other reduction method (including "No reduction") on the Breast Cancer dataset. Potentially, lower number of features/components that are chosen randomly helped to reduce overfitting and thus improve generalisation.

Finally, we tried to reduce number of features using Isomap. For both dataset, using this dimensionality reduction technique allowed us to decrease the feature space the most: i.e., to 3 components in the Wine Quality dataset and to 5 in the Breast Cancer dataset. To choose these numbers we relied on elbow method on reconstruction error. Interestingly, we see very different results when evaluating classifiers trained on top new features: in the Wine Quality dataset classifiers trained on top of Isomap output show the worst performance, while in the case of the Breast Cancer dataset this is almost completely the opposite. This might be a clue that non-linear methods do not work as well on the former dataset, suggesting potential overfitting due to model's complexity.

Dim. Reduction	Micro F1 Score, CV					
	DT	RF	BAG	SVC	SGD	LR
No reduction	0.506	0.558	0.516	0.540	0.487	0.551
PCA(n_comp=8)	0.487	0.546	0.504	0.519	0.460	0.531
ICA(n_comp=4)	0.489	0.521	0.492	0.496	0.428	0.442
RP(n_comp=10)	0.481	0.537	0.512	0.510	0.457	0.514
Isomap(n_comp=3)	0.472	0.503	0.460	0.487	0.394	0.485

(a) Wine Quality dataset

Dim. Reduction	F1 Score, CV					
	DT	RF	BAG	SVC	SGD	LR
No reduction	0.930	0.966	0.950	0.977	0.962	0.979
PCA(n_comp=7)	0.941	0.962	0.953	0.970	0.957	0.977
ICA(n_comp=9)	0.878	0.946	0.893	0.945	0.940	0.786
RP(n_comp=25)	0.947	0.967	0.955	0.979	0.975	0.977
Isomap(n_comp=5)	0.941	0.967	0.954	0.975	0.958	0.974

(b) Breast Cancer dataset

Table 1—Evaluation of dimensionality reduction algorithms using simple classifiers: Decision Tree (DT), Random Forest (RF), Bagging (BAG), Support Vector Machine (SVC), Support Vector Machine with SGD (SGD), Logistic Regression (LR).

4.3 Step 3: Applying clustering to dimensionality reduction outputs

We now take the outputs of the dimensionality reduction algorithms (with the optimal number of components identified on the previous step) and apply K-Means and GMM clustering techniques on top of these results. We summarise evaluation of resulting clusters (with optimal number of clusters as defined in Section 3.1) for all combinations of clustering and dimensionality reduction algorithms in Table 2.

Clustering	Dim. Reduction	RI	VM	FMI
KMeans(k=7)	No reduction	0.625	0.060	0.262
KMeans(k=6)	PCA(n_comp=8)	0.624	0.058	0.264
KMeans(k=6)	ICA(n_comp=4)	0.622	0.054	0.272
KMeans(k=7)	RP(n_comp=10)	0.619	0.037	0.253
KMeans(k=5)	Isomap(n_comp=3)	0.606	0.055	0.291
GMM(k=8)	No reduction	0.625	0.063	0.269
GMM(k=14)	PCA(n_comp=8)	0.648	0.076	0.217
GMM(k=6)	ICA(n_comp=4)	0.600	0.064	0.315
GMM(k=11)	RP(n_comp=10)	0.641	0.065	0.229
GMM(k=16)	Isomap(n_comp=3)	0.646	0.062	0.192

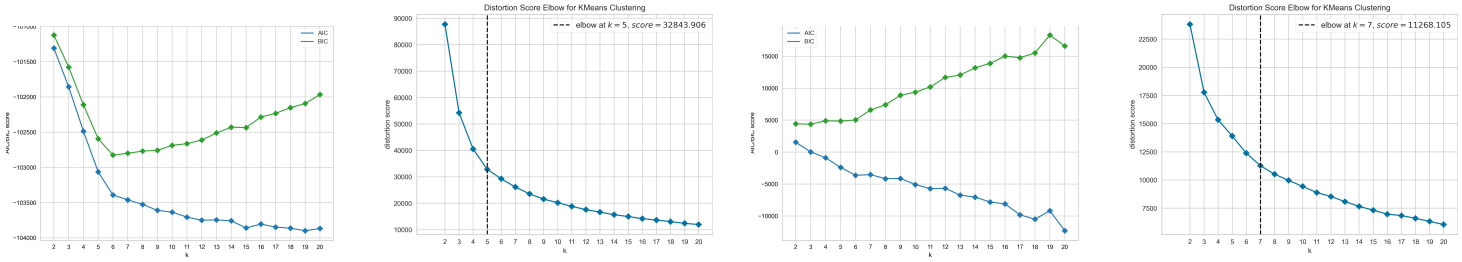
(a) Wine Quality dataset

Clustering	Dim. Reduction	RI	VM	FMI
KMeans(k=8)	No reduction	0.606	0.369	0.514
KMeans(k=8)	PCA(n_comp=7)	0.607	0.366	0.516
KMeans(k=11)	ICA(n_comp=9)	0.554	0.246	0.413
KMeans(k=9)	RP(n_comp=25)	0.593	0.332	0.491
KMeans(k=7)	Isomap(n_comp=5)	0.609	0.392	0.519
GMM(k=4)	No reduction	0.697	0.390	0.669
GMM(k=5)	PCA(n_comp=7)	0.730	0.487	0.707
GMM(k=3)	ICA(n_comp=9)	0.681	0.327	0.668
GMM(k=2)	RP(n_comp=25)	0.817	0.510	0.828
GMM(k=5)	Isomap(n_comp=5)	0.675	0.421	0.627

(b) Breast Cancer dataset

Table 2—Evaluation of clusters obtained via clustering techniques on the output of the dimensionality reduction algorithms using Rand Index (RI), V-Measure (VM), and Fowlkes-Mallows Index (FMI).

Let us have a look in depth at the results produced by GMM on top of RP for the Breast Cancer dataset as it scores the highest across all combinations for all metrics. For the Wine Quality dataset we choose combination of GMM and ICA as it is the only combination that scores more than 0.3 for FMI. Additionally, for each dataset we explore Isomap (as it is non-linear and thus differs from RP and ICA) with K-Means clustering (as opposed to GMM used with linear methods).



(a) EM & ICA(n_comp=4) applied to Wine Quality dataset

(b) K-Means & Isomap(n_comp=3) applied to Wine Quality dataset

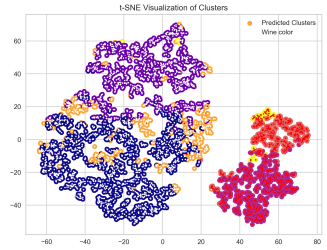
(c) EM & RP(n_comp=25) applied to Breast Cancer dataset

(d) K-Means & Isomap(n_comp=5) applied to Breast Cancer dataset

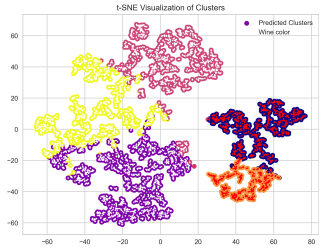
Figure 6—Step 3: Optimal number of clusters

Choosing the optimal number of clusters for EM (GMM) used on top of output of ICA for the Wine Quality dataset was done based on Figure 6a. BIC reached its minimum value at k = 6 and thus we used this number of clusters. This generally matches the number of resulting classes well. At the same time, with the K-Means ran on top of outputs of Isomap we decided 5 as optimal number of clusters (see Figure 6b). Interestingly, despite difference in clustering algorithms and dimensionality reduction algorithms that reduced feature space, we can see that general structure of the clustering is similar (see Figure 7a and Figure 7b) and even more repeats what we saw in Step 1 where clusters matched wine colours more than quality labels. Indeed, we can see that in both cases there are 2 clusters that majorly red wine and 3 clusters that majorly cover white wine. In case of the GMM-based clustering, there is one cluster that seems to cover both types of wine (thus might be representing quality), but unfortunately it is too small.

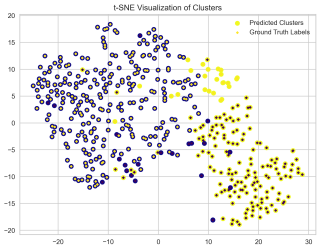
Interestingly, for Breast Cancer dataset, we were able to identify the "right" number of clusters with GMM on top of the features generated by random projection on 25 components (Figure 6c). Note that we had a problem with it in Step 1. Unfortunately, choosing the optimal number of cluster for K-Means still had its problems even in reduced feature space due to the chosen approach based on elbow method (Figure 6d). Looking at the visualization using t-SNE, we can notice that the actual geometry of the space is different: i.e., points location in Figure 7b is different from Figure 7a and what we saw in step 1 (one can indeed identify at least 5 different clusters on the plot). At the same time, Figure 7a demonstrates similar distribution of the points on the plane compared to one we saw in Figure 3d. This might be happening due to the non-linear transformations performed by Isomap that actually change distances between points.



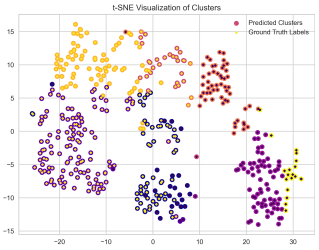
(a) GMM(k=6) & ICA(n_comp=4) applied to Wine Quality dataset



(b) K-Means(k=5) & Isomap(n_comp=3) applied to Wine Quality dataset



(c) GMM(k=2) & RP(n_comp=25) applied to Breast Cancer dataset



(d) K-Means(k=7) & Isomap(n_comp=5) applied to Breast Cancer dataset

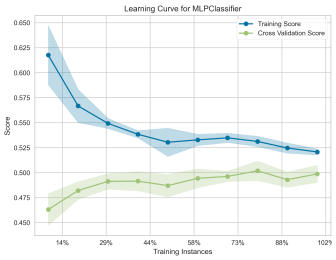
Figure 7—Step 3: Visualization of clusters using t-SNE

4.4 Step 4: Training Neural Network using dimensionality reduction techniques

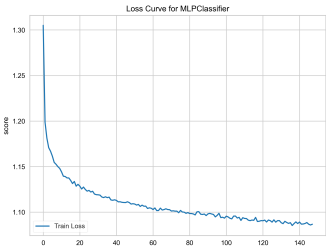
In step 4, we trained the neural network for wine quality dataset using features transformed via one linear dimensionality reduction method, PCA(n_comp=8), and one non-linear, Isomap(n_comp=3). We present the results for the models with the best hyperparameters in Table 3. First, both models achieved micro F1-score on the test set that is comparable to the baseline model. In fact, model trained on the dataset transformed via PCA even outperforms baseline model from the Assignment 1 by half percent. It is worth noting that the performance of Isomap-based NN on both training and testing set is worse than baseline and PCA-based NN. We believe that this might be related to extreme small number of features that we derived previously. Potentially, using reconstruction with some threshold value and thus choosing a bit higher number of components would improve the results here.

Second, using smaller feature space, we were able to lower the model complexity by bringing the height of the two hidden layers down from 50 to 30 for both models. We note that the lower complexity input and model allowed us to improve model generalisation: from Figures 8c and 8a we can see that the gap between training and cross validation learning curves decreases almost up to the point that two curves meet, which suggests that there are no signs of overfitting. Additionally, this is evident from the difference between micro-F1 scores on training and testing datasets for the models trained on reduced features: they generalise better on the data unseen before.

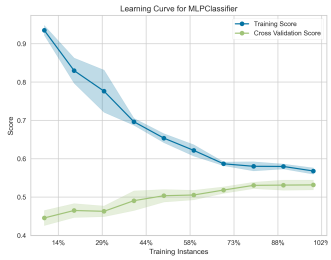
Finally, new models demonstrate improvement in terms of learning time compared to the baseline model. One of the things that definitely playing role here is simpler models: NN with less parameters generally require less training time and computation resources. At the same time, we should note that even though both models demonstrate an improvement compared to the baseline, PCA-based NN requires only 1.65 seconds while Isomap-based needs 8.7 seconds. After looking at Figure 8d and Figure 8b one can conclude that Isomap-based model requires more iterations to converge. Potentially this is happenning due to a low number of input features that are not enough to represent well the problem (i.e., every time we back-propagate a very different error for similar input). The long training time does not cause overfitting: indeed, training micro F1 score of this model is on par with testing score that helps us to exclude this from the list of potential issues.



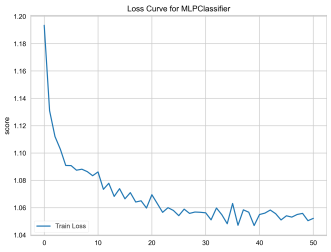
(a) Isomap, Learning curves



(b) Isomap, Loss curve



(c) PCA, Learning curves



(d) PCA, Loss curve

Figure 8—Learning and loss curves of a Neural Network trained on Wine Quality dataset with reduced dimensionality using Isomap(n_comp=3) and PCA(n_comp=8).

4.5 Step 5: Training Neural Network using cluster features

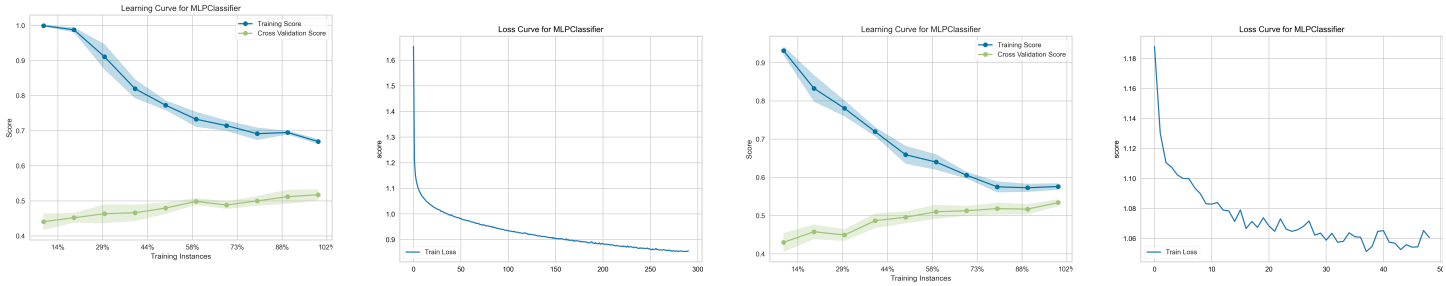
Now, we additionally use cluster features obtained on the Step 1 to train Neural Network. We explore cluster features from K-Means and EM (GMM) algorithms separately. In both cases, we decided to use features obtained after dimen-

From	Dim. Reduction	Cluster Features	Micro F1 Score		Train Time, sec	Inference Time, sec	Best Hyperparameters
			Train	Test			
Assignment 1	No reduction	No	0.7357	0.5105	20.6169	0.0079	height=50, width=2, batch_size=64, alpha=0.001, lr_init=0.01
Step 4	Isomap(n_comp=3)	No	0.5122	0.4981	8.7700	0.0014	height=30, width=2, batch_size=16, alpha=10^{-4}, lr_init=10^{-3}
Step 4	PCA(n_comp=8)	No	0.5444	0.5141	1.6529	0.0021	height=30, width=2, batch_size=32, alpha=10^{-2}, lr_init=0.0215
Step 5	PCA(n_comp=8)	K-Means clusters from Step 1 & their silhouette coefficients	0.6532	0.5320	8.7984	0.0016	height=30, width=2, batch_size=64, alpha=10^{-3}, lr_init=10^{-3}
Step 5	PCA(n_comp=8)	EM (GMM) cluster probabilities from Step 1	0.5620	0.5254	1.6556	0.0012	height=35, width=2, batch_size=32, alpha=10^{-2}, lr_init=0.0215

Table 3—Performance on Neural Networks trained on Wine Quality Dataset

sionality reduction using PCA(n_comp=8), as we saw some good results with it during Step 4. For K-Means we added 2 additional features for every sample in training and testing set: (i) cluster assigned by the algorithm to the sample and (ii) its silhouette coefficient. While it was obvious to use assigned clusters as features, we believed it would be important to also use something that could tell us how similar is a sample to its cluster. Silhouette coefficient is exactly this, as it measures of how similar an object is to its own cluster (cohesion) compared to other clusters using mean intra-cluster distance and mean inter-cluster distance. For EM (GMM), we decided that we could directly benefit from the fact that this is a soft clustering technique and thus use probabilities as features. This way, we added 8 new features (probabilities corresponding to each cluster) to other 8 that come from PCA transformation.

We provide the results in Table 3. Notably, judging by the micro F1 scores obtained on test dataset, both these models superior previously trained models in Step 4, as well as model from Assignment 1. Resulting train scores are also higher than ones we achieved in Step 4, but lower than one for the baseline NN, which is obviously overfitted. While K-Means model performed best on the test set, we can clearly see from the difference in cross validation and training scores depicted in Figure 9a and difference in resulting micro F1 scores for test and train set that it shows signs of overfitting: discrepancy between training and cross-validation score is high compared to other models. As we can see both from the table above and Figure 9c, EM-based model does not have this issue. We can notice that model with K-Means cluster features requires about 5 times more training time than one that uses probability outputs from EM. From Figure 9b and Figure 9d, we can see that this is caused due to a higher number of iterations that K-Means-based NN does before convergence. Potentially, we can limit the number of iterations for this model by 200 and it will strike a good balance between learning enough concepts from the training dataset and generalising to unseen data. Another difference that one can notice is that model with GMM probabilities required more complex model to reach its best results, i.e., height of the hidden layer being 35 instead of 30 like for the other models. Intuitively, this makes sense given that the number of input features for this model is sufficiently higher than for other models.



(a) PCA & K-Means clusters, Learning curves

(b) PCA & K-Means clusters, Loss curve

(c) PCA & EM clusters, Learning curves

(d) PCA & EM clusters, Loss curve

Figure 9—Learning and loss curves of a Neural Network trained on Wine Quality dataset with cluster features from Step 1.

Summarising this section, we can make a conclusion that features obtained as a result of unsupervised learning algorithms (like clusters or they probabilities) might be useful in a supervised setting. Additionally, one can improve performance of their supervised models by reducing the input feature space saving themselves from problems that might arise from the curse of dimensionality.