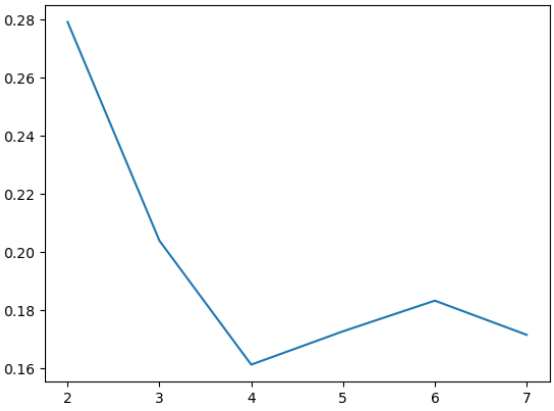
1. Finding the optimal groups using Kmean.

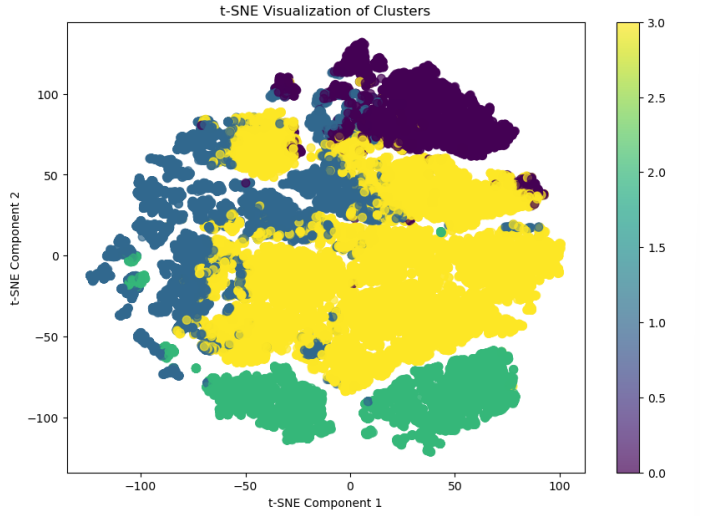
Before I start to find the most optimal groups, we need to preprocess the data. I dropped all the null data, encode a few features and standardize the data. I split our data into X (features) and Y (ground truth).

I use Kmean as the first method because this is the fastest and easiest technique to understand. First, I look for the most optimal `k` cluster in `X `using silhouette score. Here is the graph



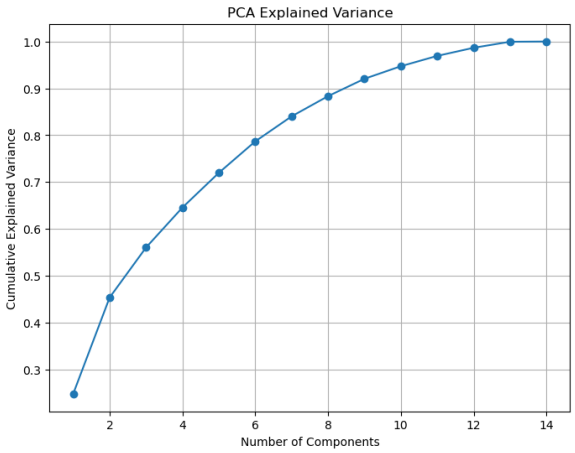
Seems like the optimal number of clusters is 4. Since I have the ground truth of `sensor location`, I will be using that feature to test the performance of our model. I am not using purity score because its bias toward the number of clusters, and each clustering algorithm will not have the exact same number of clusters as the ground truth. I will use Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI). ARI and NMI measure the similarity between 2 clustering while considering the clustering results and ground truth. ARI values range from -1 to 1, with 0 being random clustering, while NMI values range from 0 to 1, with 1 being the highest. The ARI and NMI scores of are 0.18 and 0.347, which show Kmean of X do not align with the ground truth very well.

To visualize the clusters, we can use t-SNE, an algorithm great for visualizing clusters but not preprocessing.

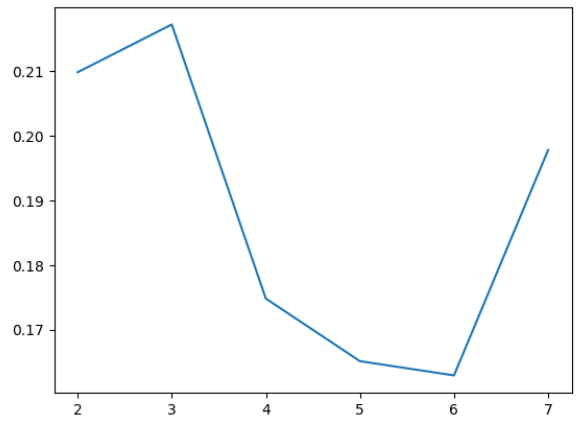


There seems to be some overlap labels, but overall, I can distinguish all 4 classes. We can reduce the number of features using a couple techniques, but I think the most common and easiest to interpret is PCA. PCA is great at capturing the overall structure of our data, like the variance and pairwise distance. I will discuss about another dimension reduction technique in later questions, but for the question 1, 2, 3, we will stick with PCA.

Applying the PCA into `X`, here is the graph on the explained variance.



10 PC (Principal components) can explain 95% of the variance in our features. I called this newly derived features `X_pca`. Applying this to the Kmean algorithm, we get



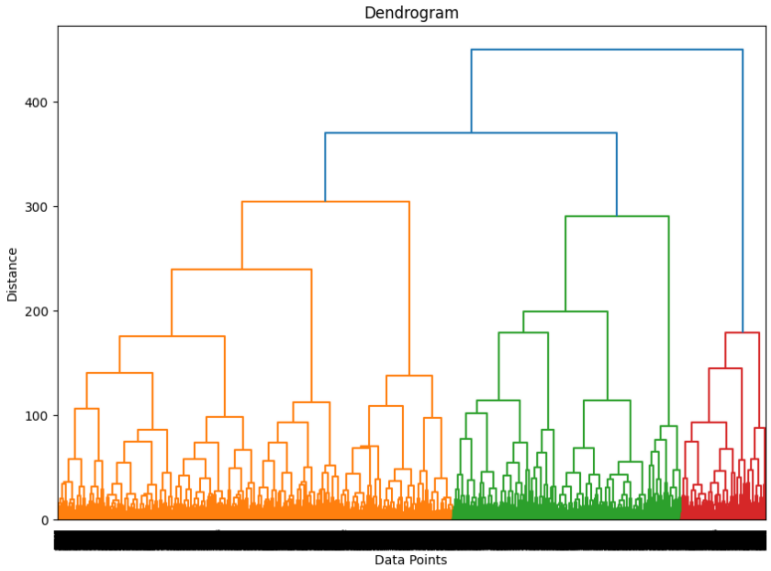
6 as the optimal k clusters. The metrics ARI and NMI are 0.188 and 0.282, a decrease in NMI metric. So, for Kmean, while PCA does reduce the dimension, but it sacrifices the performance of clustering groups. The optimal number of clusters in this case is 4.

2. Implement 2 alternative solutions.

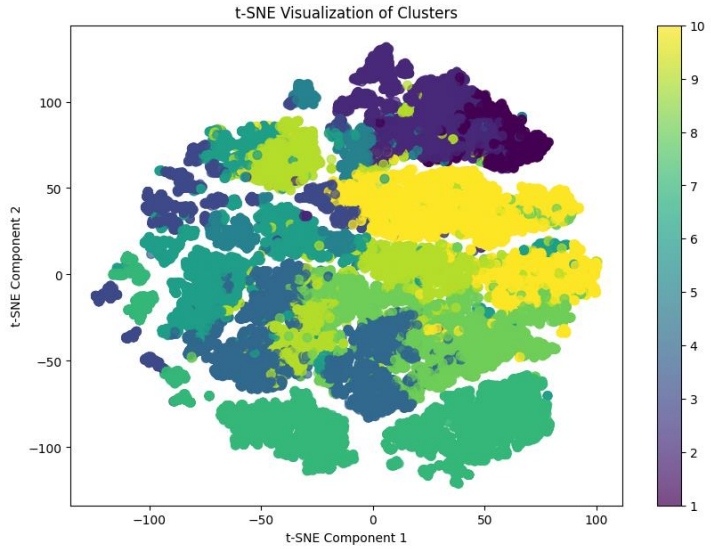
The 2 other clustering algorithms used are Hierarchical clustering and DBSCAN.

Hierarchical clustering

This algorithm groups similar instances via a tree structure. Each data point is considered to be a cluster initially and will be merged into groups later. This picture shows the dendrogram between distance and data points.

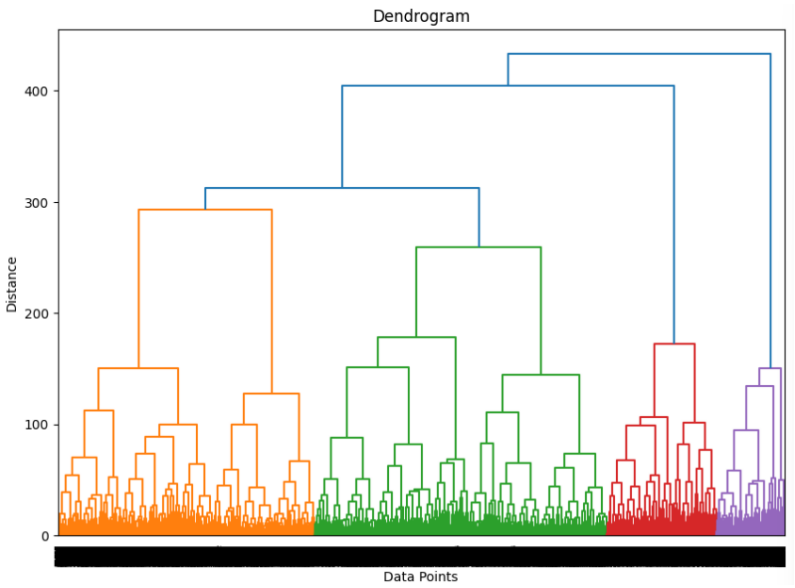


I will test a couple of distances to choose the best cut of the Dendrogram. I found that distance 150 and 175 are the best one with the same ARI and NMI scores of 0.32 and 0.41. These scores suggest the clusters formed by this distance are better aligned with the ground truth labels compare to those with Kmean. There are a total of 10 clusters found, and we can plot using t-SNE.



I can see some dominant clusters, but it is quite hard to visualize 10 clusters on a 2D graph. I will use PCA on the Hierarchical clustering as well. The process is similar to that of Kmean, with 10 components being 95% of the variance.

Here is the dendrogram after reducing the dimension.



I will not look for the best distance corresponding to with the metrics. 150 have the best performance with 0.30 , 0.42 on ARI and NMI. There are 11 labels in our case. There is a slight difference in ARI in PCA, but it is neglectable, and the trade-off in terms of time optimizing is good, so I would say that we can use the results from PCA as the optimal number of groups.

Like I said earlier, PCA is easier to intepret than other dimension reduction algorithms. Each component is the linear combination of all the features, so we can look for the most important features summing the contribution of features in the conponents. I summed the 5 most important features in each components, and found that `AverageWindDirection`, `Longitude`, `MinimumWindDirection`, `MaximumWindDirection`,` AtmosphericPressure` and `Noise` are the most important features in this case.

DBSCAN

DBSCAN or Density-Based Spatial Clustering of Applications with Noise find core samples in high density regions and group them together. If data point is not in those groups, it is considers as noise. I have performed a grid search to find the optimal number of hyperparameters in respect to the NMI score, but the score is only 0.038, very bad. I then applied PCA to see if there is any improvement, but there is no improvement.

3. Evaluate quality of the groupings.

I will start from worst to best. DBSCAN performs the worst and took the longest time to run. I didn't record the time, but I have to use the RAPIDS framework to make use of the GPU resources to compile the model and it’s still very slow. I have used the NMI metric, and it only returns 0.038 as the best performance.

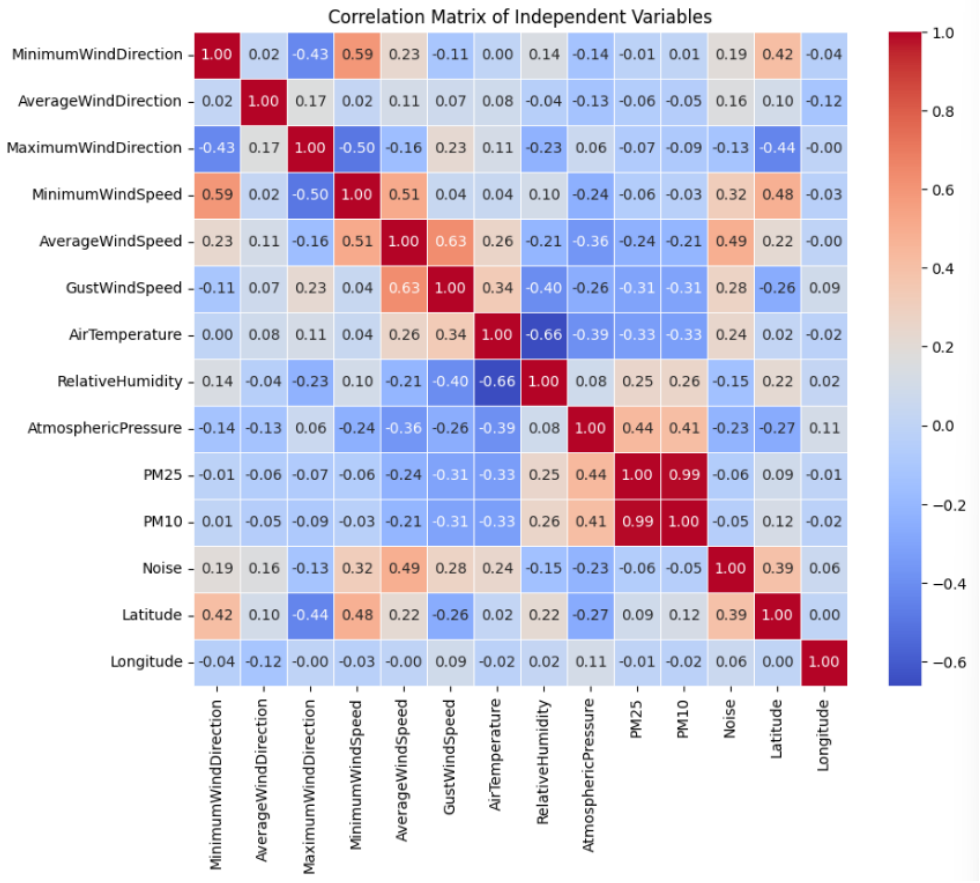
Kmean performance is not bad, while PCA does reduce the dimension, it sacrifices the performance of clustering Kmean groups. Therefore, the optimal number of clusters in this case is 4.

The best cut of Dendrogram based on distance before PCA is 175 and 150 with 0.319737 and 0.414277 in the ARI and NMI metrics. After PCA, the distance of 150 with 0.303090 and 0.418161 in the ARI and NMI metrics is the closest cluster effectiveness to the ground truth. There is a slight decrease in ARI in PCA, but it is neglectable, and the trade-off in terms of time optimizing is good, so I would say that we can use the results from PCA as the optimal number of groups, which is 11.

4. Quantifying Relationships Among Independent Variables and Visualizing Collective Variables.

Quantifying Relationships Among Independent Variables

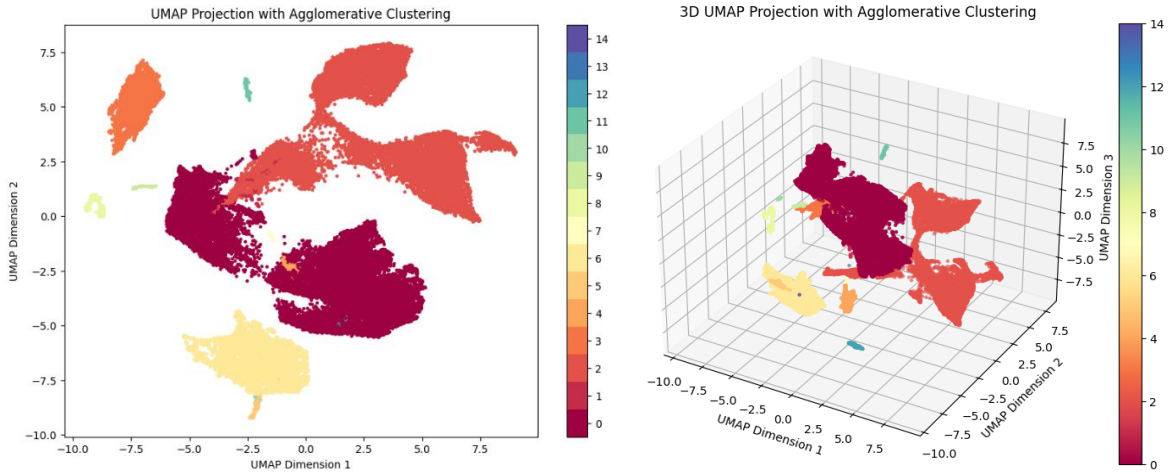
The most straight forward way to quantify is plot a correlation matrix.



We can see that there are some correlations between the wind variables, the two variables PM25 and PM10 has almost perfect correlation.

Collective Variables

These variables or features are derived from the original features. The two collective variables I will be examining are PCA and UMAP. I have already covered PCA quite carefully in the previous questions, so I will cover UMAP in this section. Before diving into UMAP, we need to understand that although both algorithms reduce our data dimensions, PCA focuses on keeping the global structure of our data, like the overall variance and the outer shape of the data. UMAP preserves more local structures where it covers the inner aspects and relationships between data points, like clusters and manifolds. The interesting thing about UMAP is that we need a scoring system for UAMP to preserve the local structures. I have used Hierarchical clustering and ARI score for the scoring system. After random searching for the best hyperparameters, the optimal parameters give an ARI score of 0.579, which outperforms the Hierarchical clustering with PCA. UAMP clusters into 14 groups and reduces `X` to 3 features. It is quite hard to interpret the derived features, but we can visualize the clusters quite nicely. Here are the 2D and 3D plots.



Looking at these 2 plots, we can clearly see distinct clusters. Although some clusters are more prominent than others, like cluster 0, 2, 6, 3, 4, 8 are the most distinguishable.

5. Identify loss of information after dimension reduction.

PCA (specifically)

We can look at the reconstruction error, the lower the reconstruction error, the less information lost. Using mean square error, we have a reconstruction error of 0.052 indicates the error is quite low, and the PCA did a good job in reducing the dimension. We can also look at the Explained Variance Ratio, but we already know this value since we use that to choose the number of components to use to fit PCA.

UAMP (specifically)

The metric called `trustworthiness` can measure how well the UMAP preserves topological structure. In our case, the `trustworthiness` score is 0.99, which shows UMAP preserves the local structure very well.

Both PCA and UMAP

Pairwise Distance Preservation

We compute pairwise distance after dimension reduction and use correlation to see how well that technique preserves the distance. This metric is leaning towards PCA since preserving pairwise is preserving global structure, but it and be tested on both algorithms. I have only sampled 30% of the original dataset because it is too computationally expensive to test the full dataset.

The correlation between original and PCA distance is 1, indicating a perfect linear relationship, so it preserved the original distances extremely well.

The correlation between original and UMAP distance is 0.33, indicating a weaker relationship and has not preserved the original distance as well as PCA. But if we think about it, UAMP is designed for capturing more complex structures like clusters and manifolds, rather than strictly preserving distance. It is better at uncovering underlying structures like clusters, not the pairwise distance like PCA, so it is understandable how it performs worse.

Silhouette Score

Silhouette Score is used for cluster structure preservation, or to see how well clustering structure of the data is preserved after dimension reduction. This metric is leaning toward local structures, but both algorithms can still be tested. I have tested on original space, PCA and UMAP. The final results show UMAP has a score of 0.567, outperforming the others by 40%.

The term loss of information from the question is quite vague, I will clarify it as what kind of structure you are willing to sacrifice. PCA and UMAP both did a good job in preserving global or local structures. We also need to take into consideration the ARI metrics and interpretation. Comparing the clusters derived from UMAP and PCA using Hierarchical clustering, we see that UAMP have better results, but in terms of interpretation, PCA is better since we know the contribution of each feature in each PC.