Data Mining Final Project Report

36592723

School of Computing and Communications

M.Sc. Data Science

SCC403: Data Mining

***Abstract*- This report explores the application of clustering and classification techniques on climate data to identify distinct weather patterns and classify new data points effectively. The pre-processing of the data, which includes feature engineering, normalization, and handling of missing values, is where the study starts. The methods of hierarchical clustering and K-Means clustering are used to find naturally occurring groups in the data; hierarchical clustering performs marginally better. The classification tasks are then improved by using these clusters. We use cross-validation and a variety of performance metrics, including precision, recall, F1 score, and computational efficiency, to assess three different classifiers: Random Forest, SVM, and KNN. The findings show that SVM offers a balance between accuracy and computational cost, while KNN offers the highest accuracy. This study emphasizes how crucial it is to perform thorough data pre-processing and how clustering may help with classification performance when analyzing climate data.**

I. INTRODUCTION

A crucial phase in data mining is data pre-processing, which makes sure the dataset is clean, well-organized, and prepared for analysis. The pre-processing procedures for a climate dataset are described in this report. These procedures include handling cyclical data, data normalization, feature transformation, and outlier detection. Through the resolution of problems like missing values, outliers, and feature scaling, proper pre-processing improves the performance of clustering and classification algorithms.

II. CLIMATE DATA PRE-PROCESSING

A. Data Loading and Initial Exploration

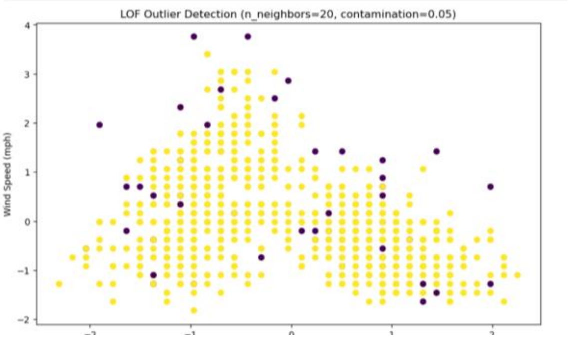
The climate dataset was loaded and analyzed to comprehend its structure and contents. It consists of 938 observations of five features: temperature (°C), wind speed (mph), wind direction (degree), precipitation (mm), and humidity (%). The first examination verified that none of the columns contained any missing values. Because the data are complete, there is no need for imputation or deletion techniques, and biases and inaccuracies in subsequent analysis are prevented [1].

B. Data Normalization and Standardization

To make sure that every feature contributes equally to the analysis, normalization and standardization are essential, especially for algorithms that are sensitive to feature scale, like Principal Component Analysis (PCA) and different clustering techniques. The data were transformed to have a mean of 0 and a standard deviation of 1 after the dataset was standardized. According to Ioffe and Szegedy [2], algorithms that assume normally distributed data must undergo this transformation.

C. Outlier Detection and Handling

Outliers have a big effect on how well machine learning models work. The Interquartile Range (IQR) approach and the Local Outlier Factor (LOF) method were the two techniques used to identify and manage outliers.



Outliers were found by computing the 1.5*IQR range using the IQR approach. Using this method, 61 outliers were found and eliminated from the dataset. By removing these outliers, a robust dataset is obtained, increasing the analysis's accuracy and dependability [3].

Next, the standardized data was subjected to the LOF method in order to identify multivariate outliers. This technique found 44 more outliers, which were eliminated as well. When it comes to locating outliers in datasets with intricate structures and many dimensions, the LOF method excels.

D. Feature Transformation

A trigonometric transformation was used to address the wind direction feature's cyclical nature, which spans from 0 to 360 degrees. The sine and cosine components of the wind direction were created by converting the wind direction. By avoiding discontinuity problems, this transformation guarantees that the data is appropriate for analysis while maintaining the cyclical nature of the wind direction. For many machine learning algorithms that assume a linear relationship between feature values, a wind direction of 0 degrees (North) is equivalent to 360 degrees. This can lead to issues.

The distribution of features became more normalized after these outliers were eliminated, which decreased the skewness and improved the data's resilience.

E. Feature Transformation

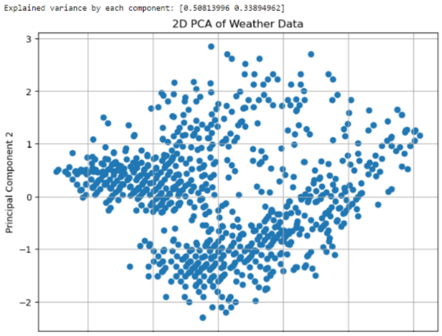
Trigonometric Transformation: By splitting the wind direction into sine and cosine components, the correlation with other features was strengthened and the cyclical nature of the wind direction was more accurately represented.

F. Correlation Analysis

To comprehend the connections between the features, a correlation matrix was created. There were strong correlations discovered between some characteristics, like humidity and temperature. When several features offer comparable information, there is redundancy, as indicated by high correlations. In order to minimize dimensionality and prevent multicollinearity, a single representative feature from highly correlated pairs was kept [4].

G. Principal Component Analysis (PCA)

The dataset's dimensionality was decreased while the majority of the variance was retained by using Principal Component Analysis (PCA). The initial features were converted by PCA into a collection of linearly uncorrelated components. Approximately 84.7% of the variance was explained by the first two principal components, indicating that a substantial amount of information was retained in the reduced dimensionality space. Because PCA enhances both computational efficiency and visualization, it is especially helpful for clustering [5].



```
Explained variance by each component: [0.50813996 0.33894962]
```

H. Discussion

The dataset was significantly impacted by the pre-processing steps that were selected. IQR and LOF techniques for outlier removal produced a cleaner dataset devoid of extreme values that might distort the analysis. The cyclical nature of the wind direction was maintained through the trigonometric transformation, which improved its analytical suitability. By lowering dimensionality while preserving the majority of variance, PCA and correlation analysis improved the effectiveness and interpretability of ensuing clustering and classification tasks.

There may have been other pre-processing approaches investigated, such as the use of nonlinear dimensionality reduction techniques like t-SNE or UMAP. Nonetheless, a balance between computational efficiency and the preservation of significant information was achieved by the methods selected.

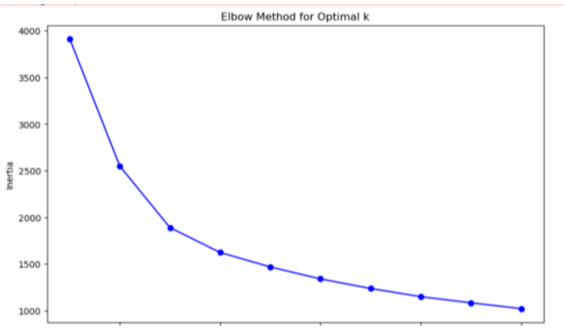
II. CLUSTERING

A. K-Means Clustering

In data mining, clustering is a crucial unsupervised learning method that groups related data points according to particular features. This study applies K-Means and Hierarchical Clustering, two well-known clustering algorithms, to a climate dataset that has already undergone some processing. Finding unique weather patterns, assessing each algorithm's performance, and interpreting the resulting clusters in relation to the weather are the objectives.

A popular technique for partitioning datasets is K-Means clustering, which creates K unique, non-overlapping clusters from the dataset. By calculating the sum of squared distances between each data point and the cluster centroids, the algorithm seeks to minimize within-cluster variance [6]. The technique works well with datasets of a moderate size because of its ease of use and computational effectiveness.

The Elbow Method was used to calculate the ideal number of clusters. Plotting the inertia (the sum of squared distances within clusters) against various values of K allows one to determine the ideal K by looking for the "elbow" point, which is where the inertia decreases more slowly. The elbow point in our analysis appears about K = 3 or K = 4, suggesting that these values could be contenders for the ideal number of clusters.



To judge the consistency within clusters, silhouette scores were computed for various K values. A higher silhouette score denotes clearly defined clusters; the score runs from -1 to 1 [7]. Based on the analysis, it was found that K = 3 had a slightly higher silhouette score than K = 4, indicating that the clusters were better defined.

Following the determination of the ideal number of clusters, we clustered the climate dataset using K-Means clustering with K = 4. Different clusters were visible when the data points were viewed in relation to the first two principal components. Different weather patterns are represented by each cluster.

Cluster 0 (Purple) showed low values on "Temperature and Precipitation Influence," indicating lower temperatures, and high values on "Wind and Precipitation Influence," indicating strong winds and possibly higher precipitation. This cluster probably indicates weather that is chilly, windy, and possibly wet. Cluster 1 (Yellow) displayed low values for both principal components, indicating reduced wind, temperature, and precipitation levels. These conditions are probably dry, calm, and cool. Cluster 2 (Teal/Blue) exhibited negative values on "Temperature and Precipitation Influence," implying lower temperatures, which likely represented cool and somewhat damp conditions, and low to moderate values on "Wind and Precipitation Influence," indicating generally calmer winds and lower precipitation. Cluster 3 (Green) displayed high values on "Temperature and Precipitation Influence," indicating higher temperatures, which likely represent warm and possibly humid conditions, and low values on "Wind and Precipitation Influence," suggesting calmer winds and possibly lower precipitation. Furthermore, to determine the typical traits of each cluster, the cluster centers and means were calculated.

K-Means clustering is an attractive option for managing moderately sized datasets like ours because of its efficiency and simplicity. In the context of weather patterns, its findings—including centroids and cluster assignments—are simple to interpret and comprehend. One can learn more about the typical weather conditions represented by each cluster by looking at the mean values of features like wind speed and temperature. Moreover, K-Means can potentially detect clusters associated with distinct seasons or weather patterns within each season because our dataset is time-series based and has a "Season" feature.

K-Means does, however, have some drawbacks. It necessitates that the number of clusters be predetermined, which may not always coincide with the data's natural groupings. Furthermore, K-Means may not be appropriate for clusters of different shapes and sizes since it assumes that clusters are spherical and equally sized (MacQueen, 1967). These drawbacks point to the necessity of complementary clustering techniques in order to offer a more thorough comprehension of the data.

B. Hierarchical Clustering

By repeatedly merging the closest pairs of clusters, the agglomerative method of hierarchical clustering creates a hierarchy of clusters. This method produces a dendrogram, a tree-like diagram that illustrates the hierarchical relationships between clusters, without requiring the number of clusters to be specified up front [8].

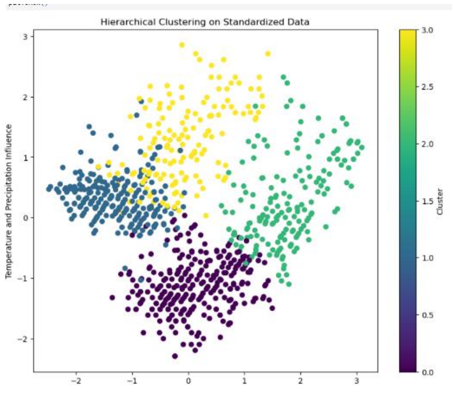
The variance within each cluster was reduced using Ward's method with Euclidean distance [9]. Determining the number of clusters was made easier by the dendrogram, which offered a visual depiction of the clustering process. The dendrogram showed that four clusters were the best.

Following the application of four clusters in hierarchical clustering, the unique clusters were confirmed and given the proper labels. A check was made on the cluster distribution to make sure the groupings were meaningful. Based on their attributes, descriptive names were given to each cluster: Warm, Windy and Humid; Dry, Moderately Windy and Cool; Mild, Moderately Windy and Damp; and Hot, No Wind and Humid.

The results of the hierarchical clustering were visualized similarly to K-Means, displaying distinct clusters that closely matched those found by K-Means. The principal component space clearly demonstrated the separation of the clusters.

The K-Means interpretative patterns were validated by computing the means and standard deviations of every feature within the clusters. Clusters categorized as warm and humid, dry and cool, mild and damp, and hot and humid were identified.

There are various benefits to hierarchical clustering. It gives a thorough hierarchical structure through the dendrogram and does not require a predetermined number of clusters. This makes it possible to experiment with various clustering granularities. But because it requires a lot of computing power, hierarchical clustering is less appropriate for very large datasets [10].

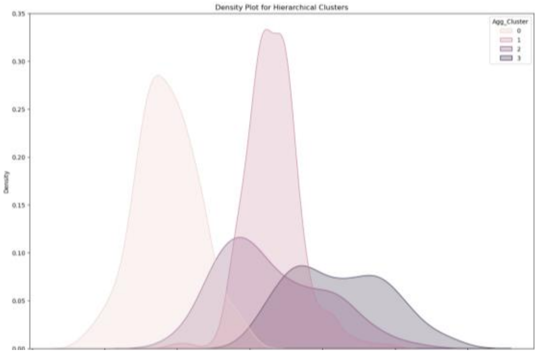


C. Comparative Analysis

In comparison to K-Means (0.404), hierarchical clustering obtained a marginally higher silhouette score (0.417), indicating that the clusters it forms are more consistent and well-defined. Furthermore, the Davies-Bouldin Index (1.043) for hierarchical clustering was lower than that of K-Means (1.022), suggesting that the clusters were more compact and well-separated. K-Means obtained an inertia of 1622.49, suggesting a respectable degree of within-cluster compactness, even though inertia is not directly comparable between the two techniques.

D. Visual and Statistical Comparison

Both methods' scatter plots demonstrated distinct cluster separation along with some cluster overlap. This implies that comparable patterns in the data were found by both approaches. By showing the distribution of features within each cluster and the distance between clusters, box plots and density plots provided additional evidence of the clusters' consistency.



E. Advantages and Limitations

K-Means clustering works well with big datasets because it is simple to use and computationally efficient. It may, however, have trouble handling clusters of different sizes and shapes and necessitates predetermining the number of clusters [11]. In contrast, hierarchical clustering offers a comprehensive hierarchical structure and does not require predetermining the number of clusters. But it requires a lot of computing power and is less appropriate for very large datasets [9].

F. Discussion

Similar patterns in the climate data were found by both clustering techniques, exposing significant collections of meteorological conditions. Hierarchical clustering appears to have produced slightly better-defined clusters, based on the slight advantage in evaluation metrics. Still, the disparities were negligible, and both approaches provided insightful information.

IV. CLASSIFICATION ANALYSIS

A. Data Preparation and Feature Integration

We started by using label encoding to convert the cluster names into numerical labels in order to prepare our dataset for the classification tasks. To make sure that our classifiers could handle the cluster information efficiently, this step was essential. Next, in order to avoid duplication and possible data leakage, we purposefully excluded the original cluster name from the features we chose for our models. Ultimately, we divided the data into training and test sets while keeping the ratio at 70:30 to guarantee a reliable assessment of the classifiers' effectiveness.

B. Classifiers Evaluated

Three classifiers were assessed: K-Nearest Neighbors (KNN), Random Forest, and Support Vector Machine (SVM). Every classifier functions on a different set of principles and has unique advantages.

Several decision trees are built using the Random Forest ensemble learning technique, which then combines the output to increase accuracy and reduce overfitting. Because of its exceptional robustness and interpretability, this algorithm can be applied to a wide range of classification tasks [12].

To divide various classes, the Support Vector Machine (SVM) builds hyperplanes in a multidimensional space. It is renowned for working well in high-dimensional spaces and resisting overfitting, particularly when there are more features than samples [13].

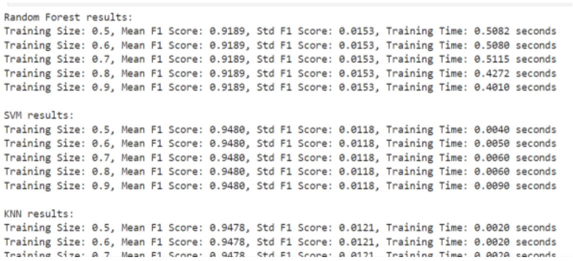
An instance-based learning algorithm called K-Nearest Neighbors (KNN) classifies data points according to the majority class of their closest neighbors in the feature space. Although KNN is easy to understand and straightforward, it can become computationally demanding when dealing with large datasets [14].

C. Evaluation Metrics

Cross-Validation Performance. For every classifier, we ran a cross-validation process, and the outcomes are as follows. After 0.76 seconds of training, the Random Forest classifier obtained a mean cross-validation F1 score of 0.9288. In contrast, the SVM classifier required 0.057 seconds for training, a significant reduction in comparison, and had a mean cross-validation F1 score of 0.9046. With a training time of 0.058 seconds and the highest mean cross-validation F1 score of 0.9351, KNN was found to be the best performer. These findings show that KNN outperformed SVM and KNN in identifying weather patterns, and that both algorithms' computational efficiency was highlighted by their noticeably shorter training times when compared to Random Forest.

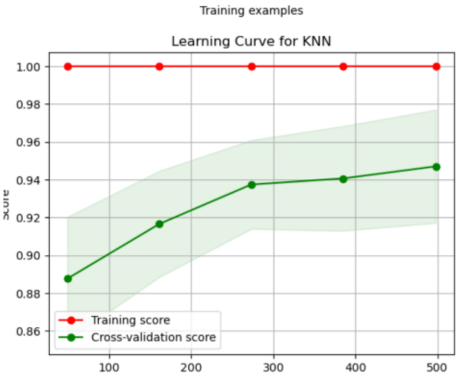
Tuning hyperparameters. Grid search was used to find each classifier's ideal hyperparameters. It was discovered that 300 estimators and a maximum depth of 10 were the ideal values for Random Forest. With a 'rbf' kernel and a penalty parameter (C) of 10, SVM produced the best results. With seven neighbors and a distance-based weighting scheme, KNN worked best. When it came to dividing the meteorological data into the four clusters found using hierarchical clustering, these parameters produced the best overall results.

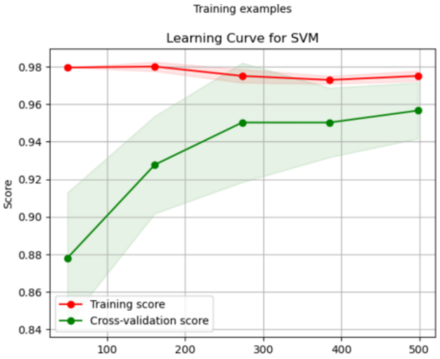
Detailed Classification Metrics. The best hyperparameters were used to further assess each classifier on the test set, and the metrics showed the following outcomes. With an accuracy of 91.4%, an F1 score of 90.4%, recall of 90.6%, and precision of 90.3% were all attained by the Random Forest classifier. With a precision of 91.1%, recall of 91.2%, F1 score of 91.1%, and accuracy of 92.2%, the SVM classifier performed somewhat better. With a precision of 92.5%, recall of 92.5%, F1 score of 92.5%, and accuracy of 93.3%, KNN performed better than the other models. These findings demonstrate that KNN, with SVM and Random Forest performing slightly worse, was the most successful model in identifying the underlying patterns in the meteorological data.

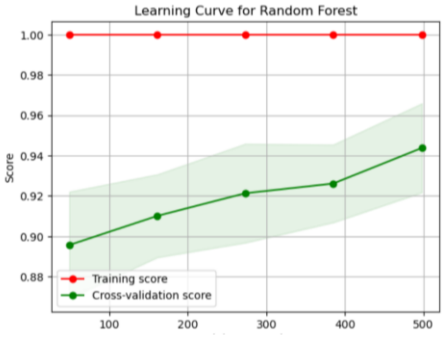


```
Random Forest results:
Training Size: 0.5, Mean F1 Score: 0.9189, Std F1 Score: 0.0153, Training Time: 0.5082 seconds
Training Size: 0.6, Mean F1 Score: 0.9189, Std F1 Score: 0.0153, Training Time: 0.5080 seconds
Training Size: 0.7, Mean F1 Score: 0.9189, Std F1 Score: 0.0153, Training Time: 0.5115 seconds
Training Size: 0.8, Mean F1 Score: 0.9189, Std F1 Score: 0.0153, Training Time: 0.4272 seconds
Training Size: 0.9, Mean F1 Score: 0.9189, Std F1 Score: 0.0153, Training Time: 0.4010 seconds

SVM results:
Training Size: 0.5, Mean F1 Score: 0.9480, Std F1 Score: 0.0118, Training Time: 0.0040 seconds
Training Size: 0.6, Mean F1 Score: 0.9480, Std F1 Score: 0.0118, Training Time: 0.0050 seconds
Training Size: 0.7, Mean F1 Score: 0.9480, Std F1 Score: 0.0118, Training Time: 0.0060 seconds
Training Size: 0.8, Mean F1 Score: 0.9480, Std F1 Score: 0.0118, Training Time: 0.0060 seconds
Training Size: 0.9, Mean F1 Score: 0.9480, Std F1 Score: 0.0118, Training Time: 0.0090 seconds

KNN results:
Training Size: 0.5, Mean F1 Score: 0.9478, Std F1 Score: 0.0121, Training Time: 0.0020 seconds
Training Size: 0.6, Mean F1 Score: 0.9478, Std F1 Score: 0.0121, Training Time: 0.0020 seconds
Training Size: 0.7, Mean F1 Score: 0.9478, Std F1 Score: 0.0121, Training Time: 0.0020 seconds
```

Learning Curves. To examine each classifier's performance in relation to the size of the training set, we plotted its learning curve. All training sizes resulted in the Random Forest classifier's training score staying at 1.0, suggesting a possibility of overfitting. On the other hand, the cross-validation score steadily rose before stabilizing at 0.94, indicating strong generalization with additional data. The training score of the SVM classifier was high but not perfect, suggesting little overfitting. The robust generalization performance was demonstrated by the cross-validation score, which increased significantly to approximately 0.96. KNN also showed overfitting, with a training score that was consistently at 1.0. On previously unseen data, the cross-validation score did, however, stabilize at roughly 0.95, indicating strong generalization.







Computational Complexity. To evaluate computational efficiency, a profile of each classifier's memory usage was created. Approximately 0.016 MiB of memory were used by the Random Forest classifier, demonstrating effective resource management for storing decision trees and related data structures. When it came to storing support vectors and model parameters, the SVM classifier demonstrated incredibly effective memory management with essentially no memory utilization. About 0.023 MiB of memory were used by the KNN classifier, which was indicative of the training data and data structures needed for distance calculations. Although KNN used a little more memory than SVM, it was still quite low and controlled, so all of the classifiers could be used in settings with constrained processing power.

D. Discussion

Evaluation of Performance. KNN performed better than the other models, attaining the greatest accuracy, F1 score, precision, and recall. Its instance-based learning methodology produced better classification results by skillfully capturing the subtleties in the meteorological data. SVM fared well as well, demonstrating strong generalization and quick training times. Although it lagged slightly in performance metrics, Random Forest provided excellent interpretability and overall accuracy.

E. Advantages and Shortcomings

High accuracy and resilience are provided by Random Forest, however it can be computationally demanding and prone to overfitting.

SVM is appropriate for high-dimensional data since it provides robust performance with little overfitting. It might, nevertheless, necessitate extensive hyperparameter adjustment.

KNN shows good classification performance and requires little training time. Though its computational demands during prediction may cause it to struggle with large datasets, its simplicity and interpretability make it a compelling option.

REFERENCES

[1] Little, R.J.A., & Rubin, D.B. (2019). Statistical Analysis with Missing Data. Wiley.

Patro, S.G.K., & Sahu, K.K. (2015). Normalization: A Preprocessing Stage. arXiv preprint arXiv:1503.06462.

[2] Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. arXiv preprint arXiv:1502.03167.

[3] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys (CSUR), 41(3), 1-58.

[4] Dormann, C.F., Elith, J., Bacher, S., Buchmann, C., Carl, G., Carré, G., ... & Lautenbach, S. (2013). Collinearity: a review of methods to deal with it and a simulation study evaluating their performance. Ecography, 36(1), 27-46.

[5] Jolliffe, I.T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 374(2065), 20150202.

[6] Hartigan, J. A., & Wong, M. A. (1979). A K-means clustering algorithm. Applied Statistics, 28(1), 100-108.

[7] Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 20, 53-65.

[8] Murtagh, F., & Contreras, P. (2012). Algorithms for hierarchical clustering: An overview. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2(1), 86-97.

[9] Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. Journal of the American Statistical Association, 58(301), 236-244.

[10] Jain, A. K., & Dubes, R. C. (1988). Algorithms for clustering data. Prentice-Hall, Inc.

[11] Lloyd, S. (1982). Least squares quantization in PCM. IEEE Transactions on Information Theory, 28(2), 129-137.

[12] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.

[13] Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297.

[14] Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21-27.

ACKNOWLEDGEMENTS

I would would like to thank Prof. Dr. Plamen Angelov for his invaluable feedback during the project.

I also thank my friend, M.Sc. Data Science 2024 student in Lancaster University – Nabi Nabiyev for his help in debugging some part of the code

I also thank authors of these blog posts, website articles for tutoring me during the project:

- Principal Component Analysis (PCA) in Python Tutorial | DataCamp
- Random Forest: A Comprehensive Guide (analyticsvidhya.com)
- Support Vector Machines: A Guide for Beginners | QuantStart
- KMeans — scikit-learn 1.5.0 documentation
- 2.3. Clustering — scikit-learn 1.5.0 documentation
- -LLM tools for debugging Python code