Data Mining Final Project

Rivyesch Ranjan, 29392004
School of Computing and Communications
MSc Data Science

Data analysis using two sets of different real-life data was performed. Two predictive data mining techniques were used, namely clustering and classification. To achieve reliable and accurate results, pre-processing techniques such as scaling, feature selection and feature extraction had to be performed on both datasets prior to classification and clustering. To gain a more thorough understanding of each technique, two clustering algorithms called K-Means and BIRCH were applied on the climate dataset and two classification algorithms called SVM and RF were applied on the video dataset.

I. Introduction

Data mining refers to the technique and process of finding useful patterns and extracting interesting information from often large, raw data [1]. The objective of any data mining process it to build an efficient predictive or descriptive model that explains a data set and is capable of generalising to new data [2]. The growth and development of a wide range of data mining techniques and algorithms has seen data mining become an essential task in many application domains such as insurance, healthcare and retail [3].

There are two main categories of data mining, namely predictive data mining and descriptive data mining. Predictive data mining is concerned with finding relationships between the independent variables and dependent variables to enable forecasting based on available data. On the other hand, descriptive data mining is more focused on discovering interesting patterns to describe the data [4].

The various underlying techniques for data mining can be grouped into four main categories which are regression, association, clustering and classification. The first two are predictive techniques while the others are descriptive techniques. The focus of this project is on the implementation of clustering and classification techniques and algorithms on two separate datasets [4].

II. Datasets

A. Data Set 1 (Climate)

The first data contains information pertaining to the climate in Basel, Switzerland. There are 18 different meteorological features and 1763 instances of data from the summer and winter seasons recorded during the time period 2010 to 2019. Clustering techniques and algorithms will be developed and applied to the climate data.

B. Data Set 2 (Video)

This data set is derived from a video stream of two moving objects, a car and a motorbike. The data shows the dimensions of the objects frame by frame. In total there are 102 image frames and 3 features. Of the 102 image frames, only the last 86 images have both the car and motorbike in the same frame. The corresponding true labels of the objects seen in each image frame are also provided.

III. Pre-Processing

A. Missing Data

A simple check was conducted to identify if the datasets had any missing values. It was found that both the climate dataset and the video dataset had no missing data instances.

B. Global Anomalies/Outliers

Detection and treating outliers are important in order to build a robust and generalizable machine learning model. An outlier is any data point in the dataset that differs significantly from the other data or observations.

Whilst there are several more complex methods to detect outliers, a simple Z score value is calculated. The threshold for classifying an outlier is set at three standard deviations from the mean. Approximately 99.7% of the features data should fall within three standard deviations of the mean. Hence, any data points that do not fall within this region are very likely to be anomalous or outliers.

For Data Set 1 113 rows were dropped due to one or more features in that row being an outlier. The choice to drop a row with an outlier entirely instead of attempting to impute the outlier value was done because the proportion of data being dropped was below 10%. The percentage of data that was dropped for Data Set 1 was only 6.31%. In the case of Data Set 2 which is relatively small, anomaly detection was not performed.

C. Scaling

Feature scaling is one of the most important pre-processing steps as it is required for correct predictions and results. Features with relatively higher magnitudes will have a higher weight or importance. This results in an incorrect bias towards certain features that may not necessarily be crucial in determining the output. There are two main feature scaling methods which are standardization and normalization. Both eliminates the units from consideration and places the data on exactly the same scale

Standardisation is a preprocessing method used to transform continuous data to make it look normally distributed. This is often necessary since many models assume that the data being trained is normally distributed. This is a requirement for many models in scikit-learn which is used to create the model.

Considering that the features found in Dataset 1 are on different scales, it is useful to transform the features so they have a mean of 0 and variance of 1. In Python this is done using the StandardScaler function. This first and foremost make it easier to linearly compare features. Secondly, it has been proven that the accuracy of models increases significantly when trained with standardized data. Another consideration is that in the subsequent pre-processing step PCA will be used. PCA is affected by scale. It is therefore

important to standardize the data so that the covariances are easily comparable for each pair of features. If the data is not scaled the principal components will be biased towards features with high variance, leading to false results. The formula for standardization is given as follows:

$$z = \frac{x - \mu}{\sigma} \quad (1)$$

where μ is the mean of the given distribution and σ is the standard deviation of the given distribution.

Unlike standardization which centres the data, normalisation converts every value of a column into a number between 0 and 1. It is similar to standardization in the sense it brings the data to the same scale. Normalisation is done on Dataset 2 using the MinMax function in Python. The formula for normalization is given as follows:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (2)$$

where x_{max} and x_{min} are the maximum and minimum values of the feature respectively.

D. Dimensionality Reduction

Whilst large and complex data does indeed contain plenty of information, it is difficult to discover the essential characteristics of the data and use it effectively. As long as the effective information is submerged in high-dimensional, complex data the data is worthless [11]. Working with high-dimensional data also requires a lot of computing time and storage space. Furthermore, there are often redundant information in the input data set which creates a well-known challenge called multicollinearity. These problems can be addressed by dimensional reduction [10].

At its core, dimensional reduction maps a data sample from a high dimensional space to a relatively low dimensional space. This mapping will inevitably lead to the loss of some original information. However, the mapping is done under the premise of maintaining the essential characteristics of the original data [11]. Reformulating the data with a smaller number of variables will simplify further processing of data, improve machine learning model's performance and help to visualize the datasets more precisely [10].

Dimensionality reduction techniques can be broadly categorized into feature selection and feature extraction. While many combinations of algorithms exist to reduce the resolution of data, there is no clear right option since each one has its pros and cons. There are different schools of thought where some prefer to extract or select the features whereas other leave the totality of the features intact [13].

Feature Selection is essentially selecting feature subsets. Hence, it causes a reduction in the resolution of the data. It is done on the basis that redundant and related features can be removed without losing much information. It optimized the interpretability of the features since every feature will keep its name and significance. Feature selection simplifies the model, decreases run time and enhances generalization by reducing excessive fitting.

For Dataset 1 two distinct methods of feature selection was carried. The first was low variance filtering which removes features with low variance. Features with low variance have less impact on the target variable. The variance is a statistical measure of the amount of variation in the given variable. If the variability of a feature is low, it does not provide any information to a machine learning model for learning the patterns and hence it can be ignored [19]. The features Snowfall Amount and Precipitation stood out for having a significantly lower variance than the rest of the features in the dataset as seen in Table 1 in the Appendix. Thus, these two features were dropped completely.

Next high correlation filtering was done. A dataset that contains multiple variables that are highly correlated to each other will have a high multicollinearity [10]. Highly correlated features can exaggerate the similarity or dissimilarity between observation and possibly result in the final solution being skewed in the direction of those features. Furthermore, they provide redundant information since an accurate prediction can be made with just one of the redundant variables. Pearson's correlation heatmap was plotted for the remaining features to identify the extent to which any two variables are linearly related. The correlation coefficient ranges from -1 to 1, where a value closer to -1 implies a strong negative correlation and a value closer to 1 implies a strong positive correlation. In contrast a value close to 0 implies a weaker correlation. From the heatmap plotted in Figure 11, the minimum, mean and maximum for temperature had a coefficient greater than 0.9 and this was the case for pressure too. Hence, the minimum and maximum was dropped for both temperature and pressure. Only the mean temperature and mean pressure was kept. Furthermore, Maximum Wind Speed and Maximum Wind Gust were also removed as these features were seen to have a correlation coefficient of 0.85 or more with other features.

Dataset 2 only had three features. All three features were found to be relevant in relation to the target variable. This is depicted in Figure 9 to Figure 11 where the profile distribution for the motorbike and car are different for each feature. However, the dataset can be considered to be imbalanced. It contains more examples of one class than another. The first 16 frames which showed only the motorbike, contained one of the two objects of interest. Hence, the class which represents the motorbike makes up a larger proportion of the dataset and can be referred to as the majority class. Given the prevalence of the majority class, it is likely that a classification algorithm will regress to a prediction of the majority class. This deteriorates the predictive power of the classification model trained. To deal with the imbalance in the dataset, the first 16 rows are removed completely. By doing this, the classification models are provided training data that have the same proportion of both classes.

Feature Extraction creates new features from the original features. Contrary to feature selection, it optimizes the integrity of the features. While it is more efficient a major drawback is that there is a loss of interpretability of the newly generated features.

PCA is a useful feature extraction technique that is used to reduce the dimensionality of a dataset. It involves linear transformation to transform a set of correlated features into a set of linearly uncorrelated features. The newly created features are known as principal components. These components are in fact a linear combination of the original variables. Usually, the first principal component has the highest possible variance followed by the second principal component, and so on [13]. The motivation behind PCA is to use the fewest components possible that capture a large

majority of the contrast. Simply, it keeps only the top components and discards the rest. The main benefits of PCA are that it overcomes duplication in features in the data set and speeds up machine learning algorithms by increasing the computational efficiency. It also enables the discovery of valuable information that explains the high contrast providing the best resolution. Overall, research has proven that PCA is an effective method that yields high accuracy when dealing with dimensions of big data [12] PCA has perfect theoretical and practical feasibility. However, its feasibility is dependent on the data being embedded in the global linear or approximately linear low-dimensional space. When performing PCA the focus is on the total explained variance but the variance does not necessarily fully reflect the amount of information [11].

Despite having already dropped eight features in the feature selection stage, there was still a need to use PCA to further reduce the dimensionality of dataset 1. Figure 1 below illustrates that selecting just three principal components explains almost 80.5% of the variance in the dataset.

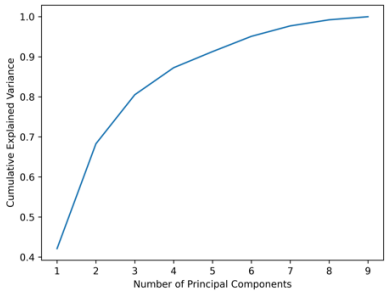

Figure 1: Cumulative Explained Variance as a function of the Number of Components

IV. Clustering

Clustering is one of the most fundamental methods that can be applied to any type of data. The purpose of clustering is to learn and understand the structure or natural grouping in a data set. The data points are segregated such that similar points are grouped into a cluster and the clusters themselves are dissimilar to each other [5]. The process of assigning data points to specific clusters are done by utilizing proximity measures that signify the extent to which two data points are similar or dissimilar. Unlike other data mining methods, the analysis does not require any prior knowledge as it is an unsupervised machine learning method [6].

A. K-Means

The first clustering algorithm applied to the climate dataset was a simple k-means algorithm. It is widely used due to its ease of use, and this is evidenced by the algorithm being listed among the top 10 clustering algorithms for data analysis. K-means is a centroid based clustering technique. The algorithm itself has a high execution efficiency and can perform cluster analysis on a variety of data types. The k-means algorithm depends on the input k, which represents the number of clusters, that must be provided beforehand. Hence, different k values will lead to different clustering results as the algorithm divides the data objects into k clusters [9].

Figure 2 and Figure 3 depict two of the most common methods to determine the optimal number of clusters for the K-means clustering. The elbow method finds the within cluster sum of squares for a range of cluster numbers. At a certain point the addition of more clusters does not significantly improve. In figure 2 it is evident that the WCSS gradient starts to flatten out or is not as steep after three clusters. Hence, the elbow method suggest that the optimal number of clusters is three for the model. The silhouette score measures the similarity of data points in its own cluster compared to the closest neighbour cluster. It also checks the compactness of the cluster. A high silhouette score gives the clusters optimal value. Figure 12 also shows that having three clusters would give good clusters. Hence, the k-means algorithm will be run with k=3 as the input.

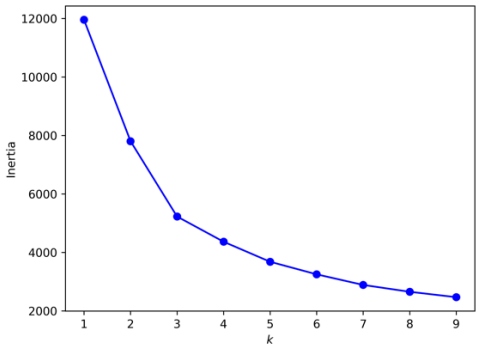

Figure 2: Optimal Number of Clusters using Elbow Method

Despite its popularity, K-means experiences several limitations. The main problem is related to the random initialization of the centroids. This could potentially lead to the algorithm converging to a poor local minima. unexpected convergence. Another concern is that the choice of initial clustering has significant influence on the results [8].

To overcome the problem of centroid initialization, a variant of K-means called K-means++ was used on the data set. The algorithm and principles are exactly the same with the exception that it employs a smart centroid initialization technique. This makes it more likely to converge and faster than running K-means alone.

Figure 3 shows the results of the K-Means clustering done on the climate dataset. The graph illustrates that the data can be classified into three distinct clusters where the separation between each cluster is clearly visible. It can also be observed that K-means has partitioned the dataset into spherical clusters.

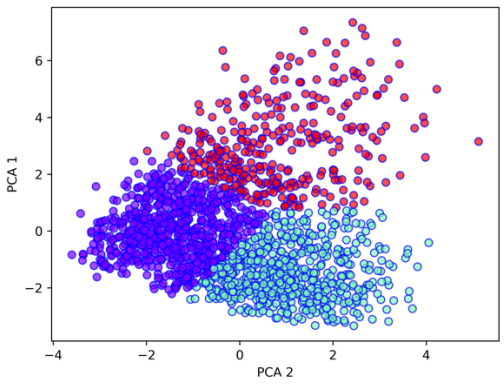

Figure 3: K-Means Clustering

B. *Balanced Iterative Reducing and Clustering Hierarchies*

More commonly known as BIRCH, it is a distance based hierarchical clustering method that can cluster effectively with one scan and handle outliers in the data. The concept is to establish a clustering feature tree stored in memory by scanning the database, and then cluster the lead nodes of the clustering feature tree. The features of hierarchical clusters are stored in the tree. The clustering feature tree has two main parameters which are branching factor and threshold. These two factors affect the size of the resulting tree [14].

The process by which BIRCH performs the clustering begins with the scanning of the entire dataset and construction of the CF tree. The information that is stored in the tree is meant to reflect all the information in the dataset as much as possible. Outliers are removed from the dataset during the formation of the CF tree. By establishing a CF tree, no further input-output operation is required in the subsequent phases which consequently reduces the computation time. Clustering is done to smaller sub-datasets of each sub-clusters in the entries of leaves of the CF. The incremental update and reconstruction of the CF tree is based on the lead nodes of the original tree, which also means that the initial scan does not need to be repeated [15].

While the number of clusters is not a required input for the BIRCH algorithm, simply not providing the number of desired clusters would lead to a plot with many subclusters. This makes it difficult to interpret the results. For hierarchical clustering dendrograms are commonly used to decide the number of clusters appropriate to the dataset. Figure 4 displays the dendrogram representation between the data points in the pre-processed dataset. The dashed line cuts across the region that has the maximum height of the vertical dendrogram line. This cut signifies that the optimal number of clusters is two. This essentially represents the maximum Euclidean distance between the optimal number of clusters.

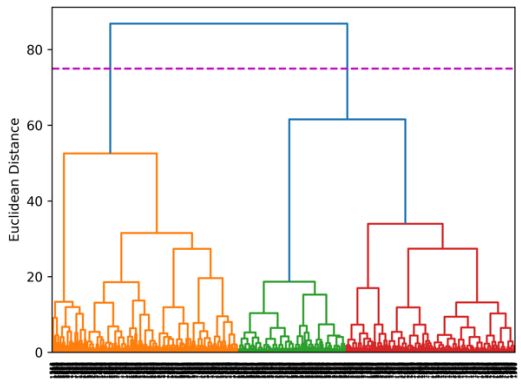

Figure 4: Dendrogram

Figure 5 represents the BIRCH clustering with just two expected clusters. BIRCH manages to create two visibly separable clusters. The separation of the clusters created by BIRCH is not as distinct as the clusters seen in the K-Means method. There is some visible overlap between the two clusters. It can be seen that some data are classified as belonging to the pink cluster when it clearly lies in the purple clusters region. The same can be said for some data being classified as belonging to the purple region when it most likely seems to belong to the pink cluster. However, since no labels were provided for this dataset there is no clear way to validate the results of the clustering.

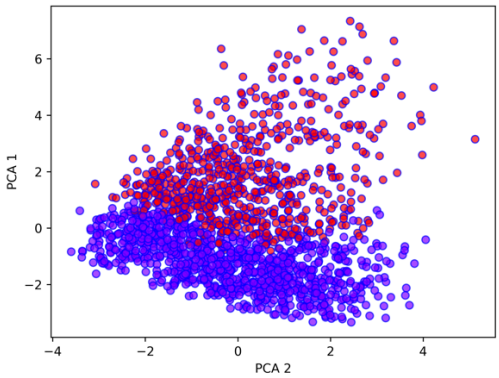

Figure 5: BIRCH Clustering

V. Classification

Classification is a supervised learning approach used to assign each instance of the data into a class. This is done based on either the class attribute or the value of an objective attribute. Each class is characterized based on the qualities of data that has previously been assigned to them. The goal of classification is to classify new data based on the predetermined classes.

In the classification process, the data is split into a training set and testing set. These two sets combined are used to create the classification model. The model is developed in the training phase by allowing it to learn from the attribute values and the corresponding target attribute values in the training dataset. The classification algorithm's prediction accuracy is highlighted in the testing phase when categorizing cases in the test set that were previously withheld during the training [7].

Over recent years, many methods for classifications have been developed. Among the most widely used techniques are Support Vector Machines (SVM), K-Nearest Neighbors (KNN), artificial neural networks (ANN) and Random Forest (RF). Many studies on classification have demonstrated that RF, KNN and SVM produce high accuracy classifiers. Moreover, family of kernel methods such as SVM and RF have emerged as very promising algorithms for classification purposes.

Imbalance in datasets is a significant factor that needs to be considered. The classification dataset was initially imbalances as it consisted of 16 additional frames of the motorbike object. Many normal classification methods have difficulties classifying minority class object in skewed datasets. Algorithms such as SVM face problems determining the optimal separation hyperplane when the dataset is not balanced. Furthermore, the model tends to be biased towards the minority class [17]. The reason for removing the first 16 frames of data in the pre-processing stage was done to avoid this issue and to develop models that give reliable classification results.

Initially the pre-processed data is split up into two sets. The first set is used to train the model and the other is used to access how the model behaves with completely unseen data. For this dataset the split is done such that 75% of the data is used as the training data while the remaining 25% is kept as the holdout set or testing data. Whilst this approach is very

common is supervised learning, there is a possibility of high bias due to underfitting if the dataset is small since some information about the data may not be used during the training phase. Moreover, the accuracy and metrics are highly dependent on how the split was performed, whether the dataset was shuffled, etc. Simply splitting the dataset into training and test sets for classification is not representative of the model's ability to generalize.

The problems mentioned can be avoided by using K-Folds cross validation. It is a technique that results in a less biased model. It does that by ensuring that every observation from the original dataset has the chance of appearing in training and test set. Cross validation is the best approach to implement when there is limited input data. It allows any model developed to be validated with more data. This in turn proves the model's consistency on unseen data.

K-Folds cross validation involves breaking the dataset into k equal parts or folds. The process of training the model and testing it on the holdout set is repeated k times. The uniqueness of this technique is that each time the process is repeated a different fold is used as the holdout set. This means every data point gets an equal opportunity at being included in the test set. The dataset is split into five k folds for the cross validation of both the models discussed below.

There are various metrics that can be used to evaluate the classification models developed. Figure 6 shows a confusion matrix. It is a table that shows the summary of the number of correct and incorrect predictions made by the model. When performing classification there are four possible outcomes that could occur.

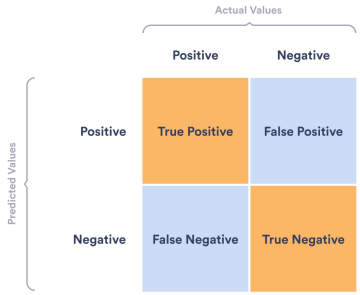

Figure 6: Confusion Matrix

The two models created will be evaluated using four metrics called accuracy, precision, recall and F1-score. These three metrics are determined from the confusion matrix. Accuracy indicate how much of the predictions are right. Precision measures the review precision whereas recall measures completeness. The more false positive the model predicts, the lower the precision. Similarly, the more false negatives the model predicts, the lower the recall. F1-score combines both precision and recall, hence the higher it is the better the performance of the model [18]. The equations of the metrics are given in formulae (3) to (6) in the Appendix.

A. Support Vector Machines (SVM)

SVM is a popular supervised learning model used for classification and regression analysis. This is due to its extraordinary generalization capabilities, along with its optimal solution and its discriminative power. It is particularly powerful for solving binary classification problems. The SVM utilizes a kernel function to map the training set to improve its resemblance to a linearly separable data set. The SVM maps the decision boundary for each class and specifies the hyperplane that separates the different classes during the training stage. The performance of the SVM is dependent on the regularization parameter and kernel parameter selected [16].

Popular kernel functions include linear, Radial Basis Function (RBF), quadratic, Multilayer Perceptron and Polynomial kernel. SVMs are especially useful when a problem might not be linearly separable due to the use of suitable non-linear kernels such as Radial Basis Function (RBF). When a problem is linearly separable then the linear kernel performs well. An SVM with a linear kernel function is faster to train and is less prone to overfitting than one with a RBF kernel function [16].

The main downside of SVM is the excessive computational cost which is due to the training kernel matrix growing quadratically with the size of the data set. This makes it a very slow process and is not suitable for large data set classification. Besides that, like many other classification algorithms the performance of the SVM suffers when training is done using an imbalanced dataset [17].

The SVM algorithm was implemented using the library Scikit-Learn and applied on Dataset 2. Three main hyperparameters were tuned to find the optimal. This search to find the optimal hyperparameters was done using the GridSearch CV function. The regularization of the error C regulates how soft a margin can be. Typically, the margins are wider when the value of C is smaller. This may lead to more misclassifications. However, simply increasing the value of C does not mean the model performs better. It could very likely lead to overfitting which reduces the model's ability to generalize well to new data. For the given dataset, the optimal regularization error was C=0.1. The type of kernel function used also affects the model significantly. For the SVM model developed for Dataset 2 it was found that the best performing kernel function was the RBF. Lastly, the gamma hyperparameter defines how far the influence of a single training example reaches. A small gamma will lower the bias but increase the variance, and vice versa. The optimal gamma was found to be 1.

The results of the SVM classification are shown below in Figure 7. Each metric shows that the SVM has perfect results and can generalize well to unseen data. The confusion matrix depicted that the SVM trained predicted zero false positive and zero false negatives. Although this is promising, it should be noted that Dataset 2 was limited in terms of size. Using a larger dataset would give a better understanding of the model's ability to generalize.

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	26
1.0	1.00	1.00	1.00	17
accuracy			1.00	43
macro avg	1.00	1.00	1.00	43
weighted avg	1.00	1.00	1.00	43

Figure 7: SVM and RF Classification Results

B. Random Forest (RF)

An RF classifier is an ensemble learning technique that consists of a number of trees. It is an ensemble learning

method because it generates many classifiers and aggregates their results. Each tree is grown using randomization and the lead nodes of each tree are labelled by estimates of the posterior distribution. At every internal node the space of data is split by a test and the leaf distributions are aggregated at the end. RF classifiers construct each tree using a different bootstrap sample of the data. Furthermore, each node is split using the best among a subset of predictors randomly chosen at that node. Hence, how the classification trees are constructed are different each time [16].

RF can be likened to a set of decision trees (DTs) voting on the class assigned to a sample and the majority winning. The nature of RF means that it is robust against overfitting and can handle high dimensional data. Also, it is rather versatile as it can handle categorical features well. The main advantage is that it only has two parameters which are the number of variables in the random subset at each node and the number of trees in the forest. It has been found that the performance is not very sensitive to the values of these parameters [16].

Nevertheless, hyperparameter tuning was still performed on the Random Forest model to obtain the best hyperparameters. Again Scikit-Learn was used to train and implement the model. The model gave optimal results when the number of trees in the forest was 1000 and the number of features provided to each tree was set to "auto". Additionaly, Scikit-Learn has other parameters which can be tuned. For the model developed, the maximum depth of a tree was set to 140.

The results of the RF classification were the same as that of the SVM classification. The results can be viewed in Figure 7. A reasonable guess as to why the results are similar would be the size of the dataset used. Algorithms such as RF and SVM were created to classify much more complex datasets than the one used in this experiment. Hence, it is not surprising that it can classify the dataset perfectly.

VI. Conclusion

The project involved the research and implementation of two fundamental techniques in data mining which is classification and clustering. Prior to implementing either technique, the data contained in the two datasets had to be pre-processed accordingly. For Dataset 1 which had relatively more features than Dataset 2, PCA was used in addition to simple feature selection techniques to reduce the dimensionality of the dataset. The pre-processing of Dataset 2 mainly involved the removal of certain data instances to fix the imbalance in the dataset. The two clustering techniques used were K-Means and BIRCH. Both algorithms proved capable of clustering the Climate data into distinct clusters. However, the clusters formed by the K-Means technique showed better separation between the clusters. The two classification algorithms used to classify Dataset 2 were SVM and RF. Both performed equally well, going so far as classifying the data perfectly. While theoretical differences were found during research, no significant conclusions can be drawn up from the experimental results since the dataset was not complex enough to result in differentiating results.

Acknowledgment

Several Python libraries were used in the code such as pandas, scipy, matplotlib, numpy and sklearn. Some chunks of code were taken from Stack Overflow and blogs on Towards Data Science and Medium.

References

[1] M. Chakarverti, N. Sharma, and R. R. Divivedi, "Prediction Analysis Techniques of Data Mining: A Review," *SSRN Electronic Journal*, 2019.

[2] S. Nalawade and A. Joshi, "A Literature Review: Data Mining Techniques, Applications & Issues," 2021.

[3] M. K. Gupta and P. Chandra, "A comprehensive survey of data mining," *International Journal of Information Technology*, Feb. 2020.

[4] K. Sumiran, "An overview of data mining techniques and their application in industrial engineering," *Asian Journal of Applied Science and Technology,* vol. 2, no. 2, pp. 947-953, 2018.

[5] M. Priyadharshini, G. Harini, and E. Subarna, "Clustering Techniques in Data Mining," 2022.

[6] V. Mehta, S. Bawa, and J. Singh, "Analytical review of clustering techniques and proximity measures," *Artificial Intelligence Review*, May 2020.

[7] A. Jain, D. Somwanshi, K. Joshi, and S. S. Bhatt, "A Review: Data Mining Classification Techniques," in *2022 3rd International Conference on Intelligent Engineering and Management (ICIEM)*, 2022: IEEE, pp. 636-642.

[8] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means Algorithm: A Comprehensive Survey and Performance Evaluation," *Electronics*, vol. 9, no. 8, p. 1295, Aug. 2020.

[9] H. Zou, "Clustering Algorithm and Its Application in Data Mining," *Wireless Personal Communications*, vol. 110, no. 1, pp. 21–30, Aug. 2019.

[10] R. Rani, M. Khurana, A. Kumar, and N. Kumar, "Big data dimensionality reduction techniques in IoT: review, applications and open research challenges," *Cluster Computing,* pp. 1-23, 2022.

[11] W. Jia, M. Sun, J. Lian, and S. Hou, "Feature dimensionality reduction: a review," *Complex & Intelligent Systems,* pp. 1-31, 2022.

[12] B. M. S. Hasan and A. M. Abdulazeez, "A review of principal component analysis algorithm for dimensionality reduction," *Journal of Soft Computing and Data Mining,* vol. 2, no. 1, pp. 20-30, 2021.

[13] J.-S. Dessureault and D. Massicotte, "Feature selection or extraction decision process for clustering using PCA and FRSD," *arXiv preprint arXiv:2111.10492,* 2021.

[14] Y. Cai, "Comparison of different clustering methods applied to omics datasets," in *2022 7th International Conference on Machine Learning Technologies (ICMLT)*, 2022, pp. 105-111.

[15] A. E. Ezugwu *et al.*, "A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects," *Engineering Applications of Artificial Intelligence,* vol. 110, p. 104743, 2022.

[16] E. Y. Boateng, J. Otoo, and D. A. Abaye, "Basic tenets of classification algorithms K-nearest-neighbor, support vector machine, random forest and neural network: a review," *Journal of Data Analysis and Information Processing,* vol. 8, no. 4, pp. 341-357, 2020.

[17] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing,* vol. 408, pp. 189-215, 2020.

[18] B. Abdualgalil and S. Abraham, "Applications of machine learning algorithms and performance comparison: a review," in *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, 2020: IEEE, pp. 1-6.

[19] S. Das and U. M. Cakmak, *Hands-On Automated Machine Learning: A beginner's guide to building automated machine learning systems using AutoML and Python*. Packt Publishing Ltd, 2018.

APPENDIX

A. Graphs

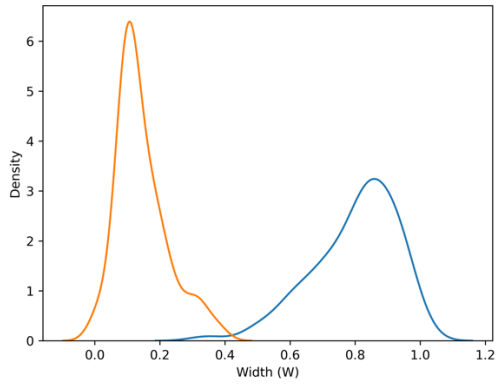

Figure 8: Data Distribution of the feature Width

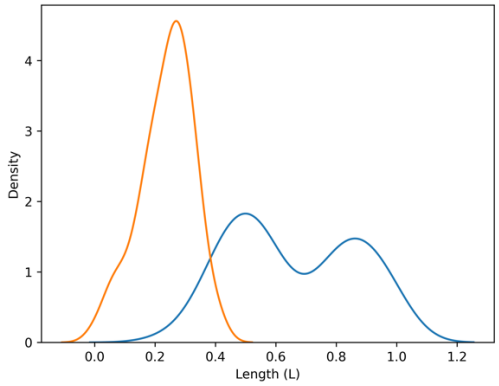

Figure 9: Data Distribution of the feature Length

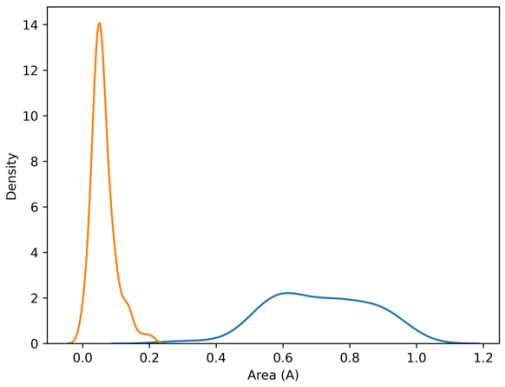

Figure 10: Data Distribution of the feature Area

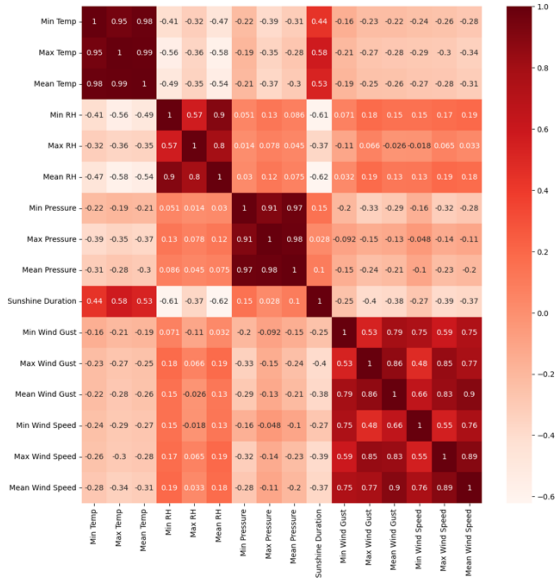

Figure 11: Pearson's Correlation Heatmap

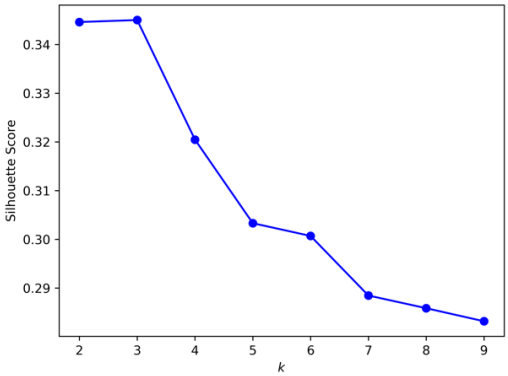

Figure 12: Optimal Number of Clusters using Silhouette Score

B. Equations

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (6)$$

C. Tables

Table 1: Variance of Each Feature

Min Temp	0.041428
Max Temp	0.055129
Mean Temp	0.053340
Min RH	0.030624
Max RH	0.028196
Mean RH	0.030944
Min Pressure	0.017183

Max Pressure	0.013688
Mean Pressure	0.014557
Precipitation	0.008741
Snowfall Amount	0.001375
Sunshine Duration	0.093765
Min Wind Gust	0.020123
Max Wind Gust	0.016374
Mean Wind Gust	0.017078
Min Wind Speed	0.016932
Max Wind Speed	0.019539
Mean Wind Speed	0.022292