Name: Arun Kumar

Student No: 21079279

GitHub link: https://github.com/ArunKumar1611/assignment_3.git

Dataset link: https://drive.google.com/file/d/1JvD4Ss2yS3d9X36YkWqmqZXLamNWLSFJ/view?usp=sharing

Title: Weather Data Analysis: Uncovering Patterns and Predicting Future Trends

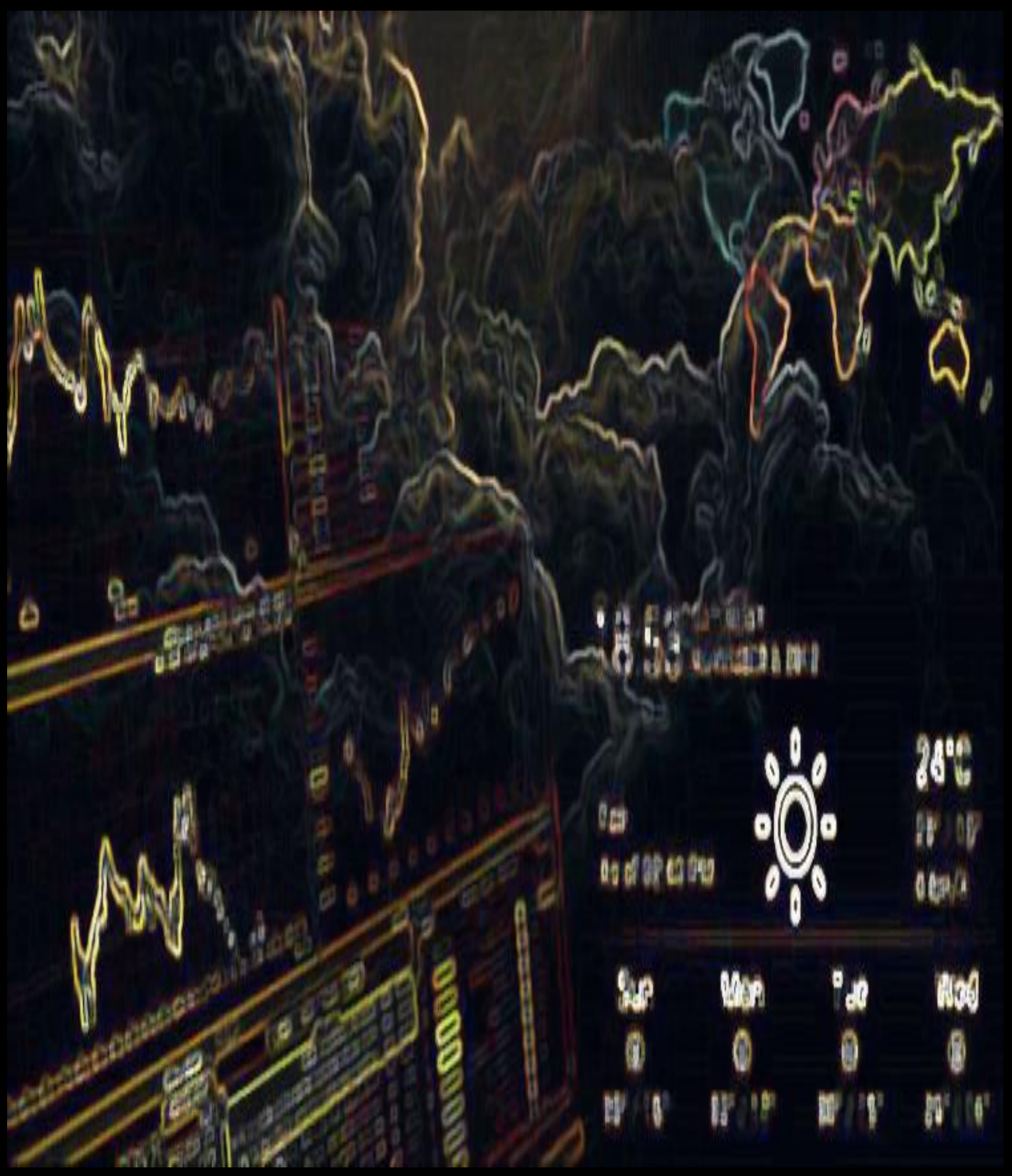
CLUSTERING & FITTING

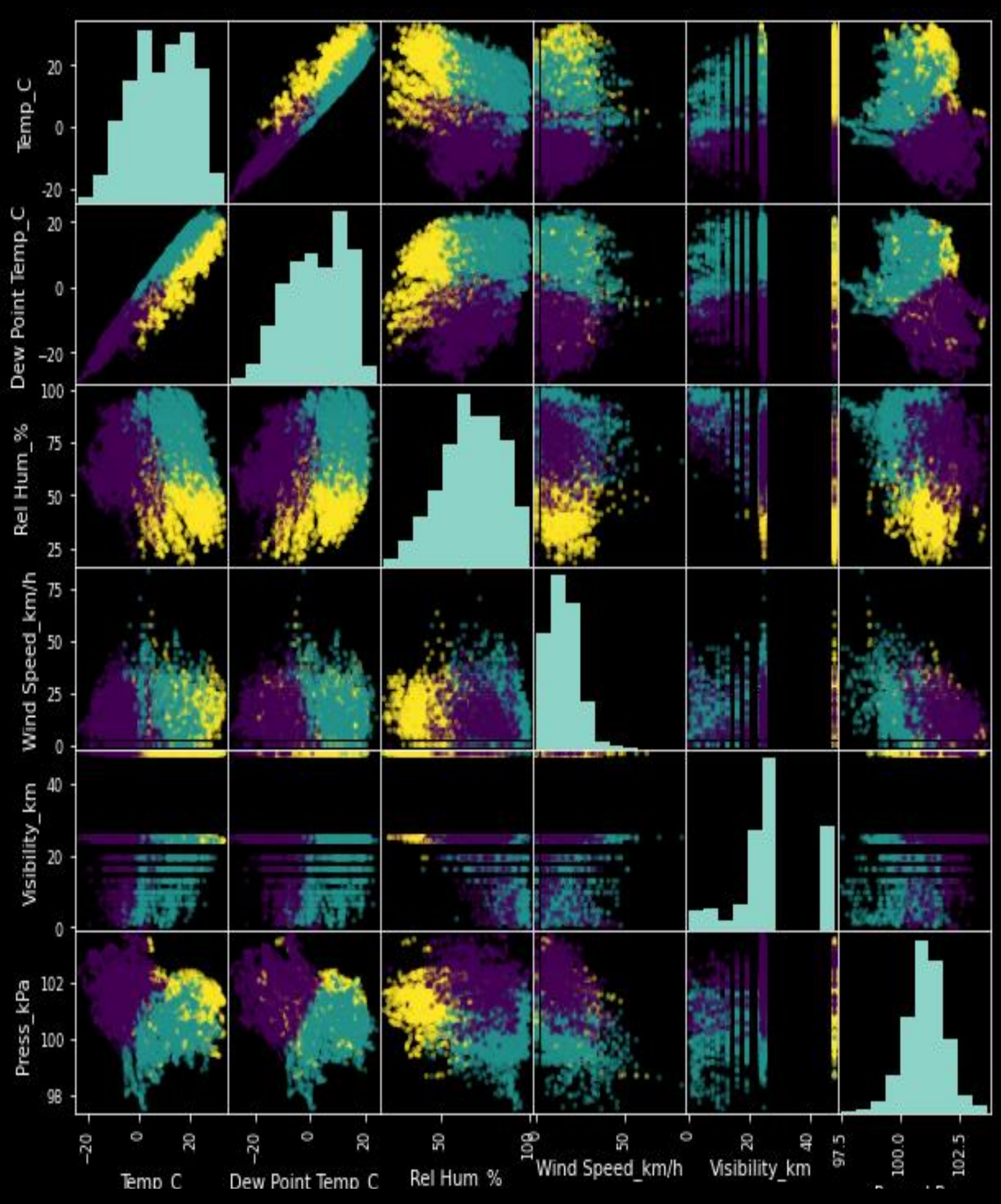
Abstract:

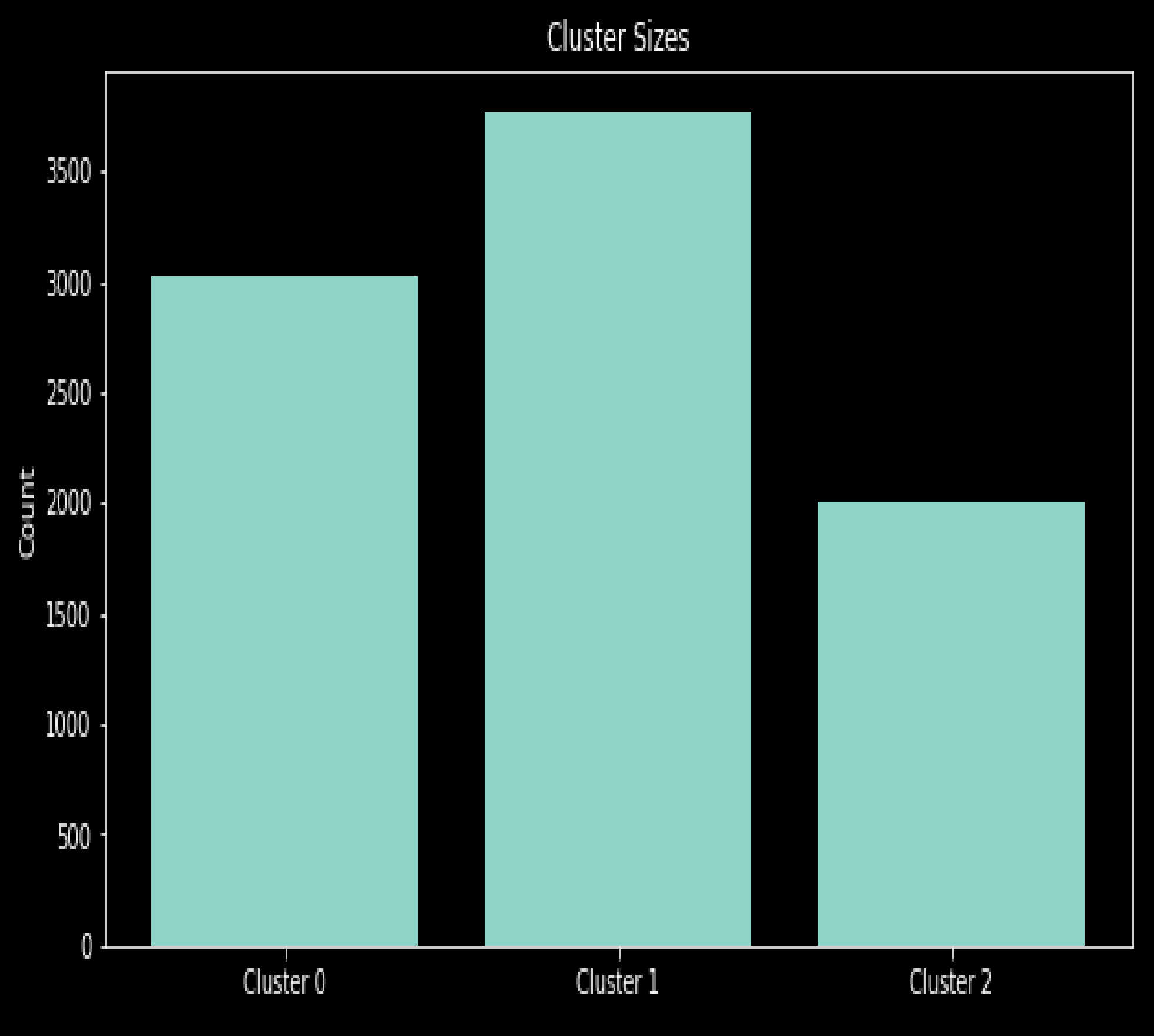
This code performs k-means clustering on weather data to identify three distinct clusters based on temperature, dew point, relative humidity, wind speed, visibility, and pressure. The clusters are visualized using scatter plots and histograms. Additionally, an exponential growth model is fit to the temperature data, and confidence ranges are estimated to predict future temperature trends.

Introduction:

In this project, we analyze a comprehensive weather dataset to uncover hidden patterns and gain insights into the relationship between various meteorological parameters. The dataset consists of features such as temperature (Temp_C), dew point temperature (Dew Point Temp_C), relative humidity (Rel Hum_%), wind speed (Wind Speed_km/h), visibility (Visibility_km), and pressure (Press_kPa). Our primary objectives are to identify distinct clusters within the data and employ an exponential growth model to predict future temperature trends. This analysis will not only help us understand the underlying structure of the dataset but also shed light on potential long-term changes in temperature, which could have significant implications for climate change and related policies.







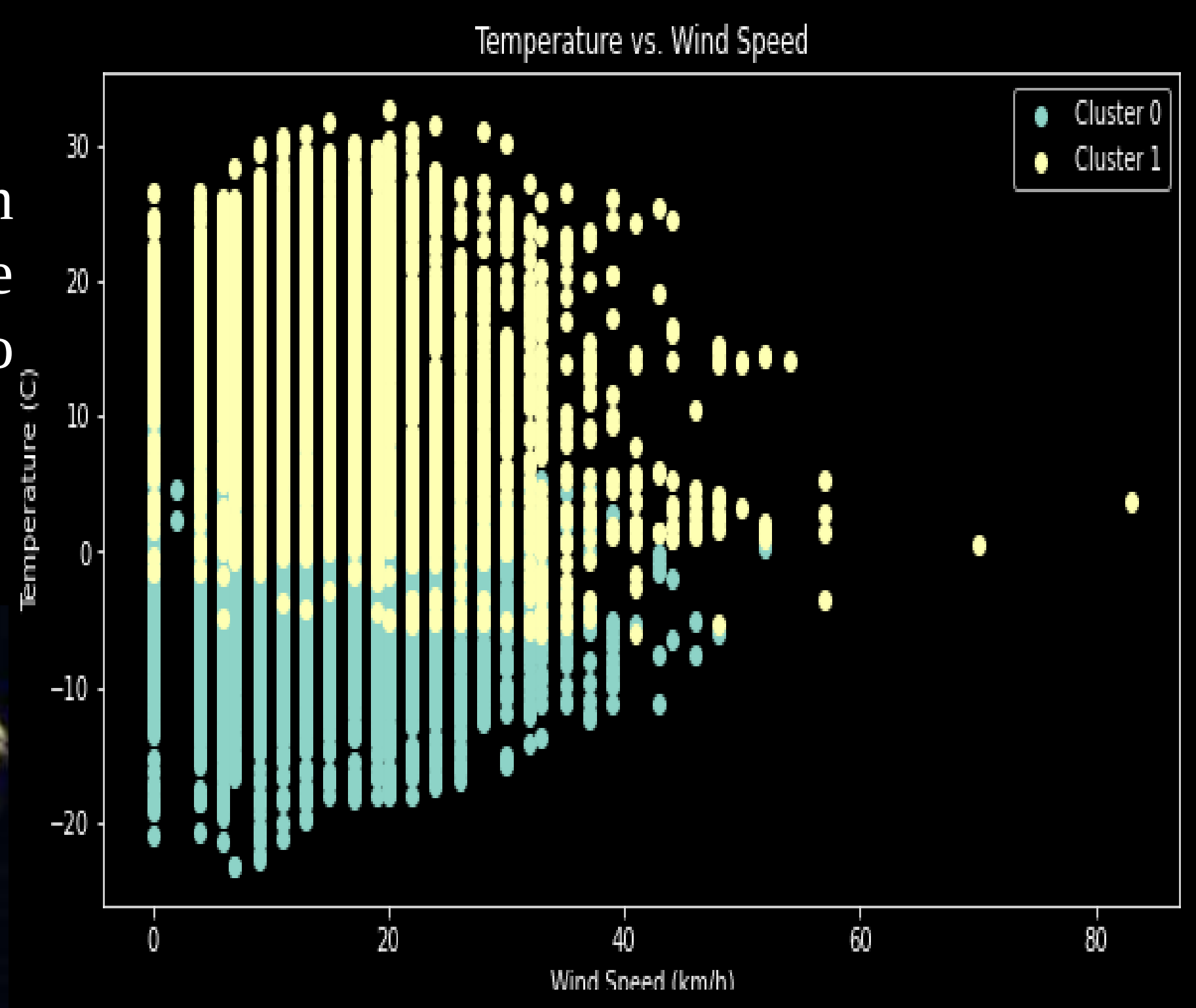
Data Pre-processing:

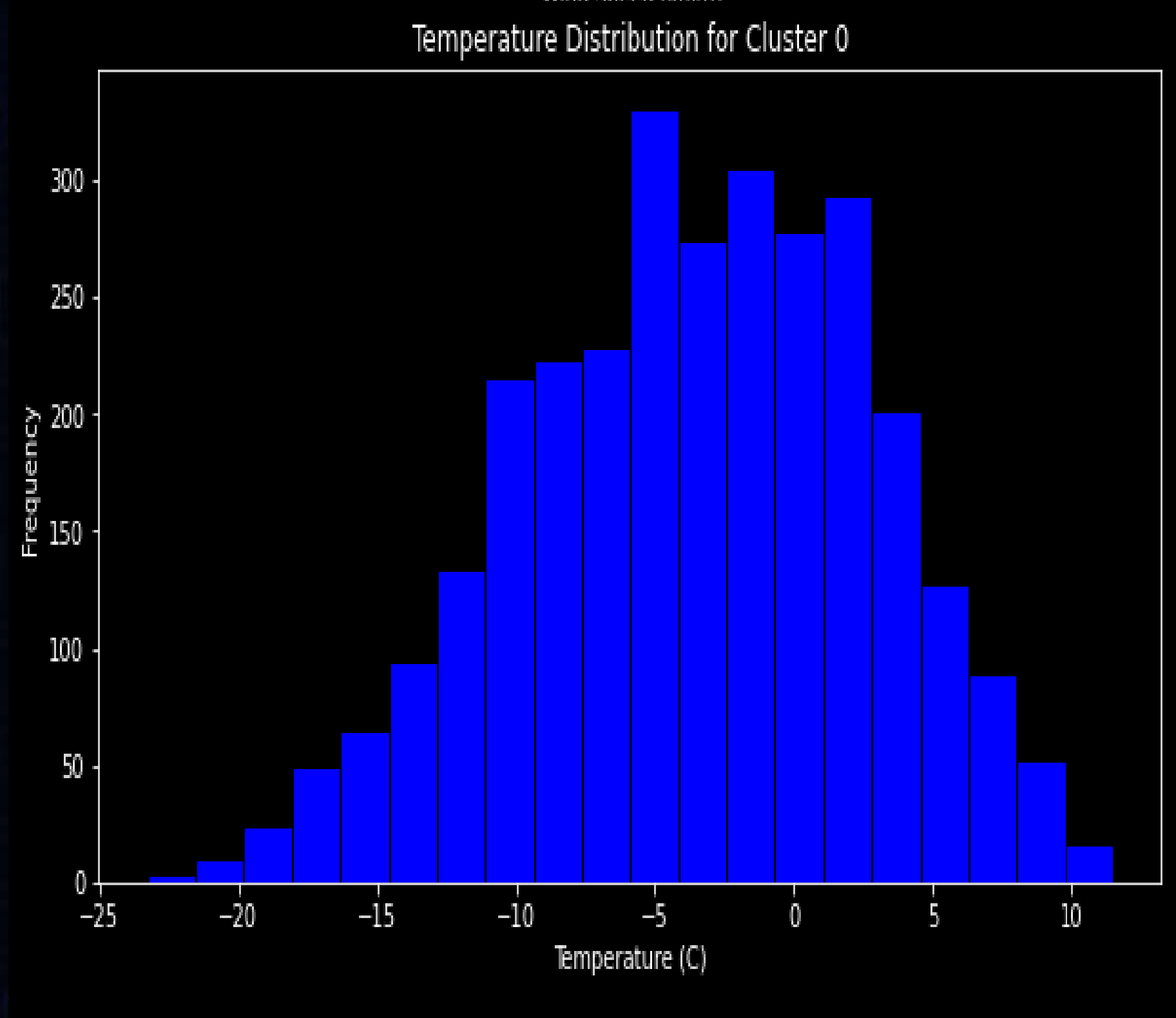
This code is an example of using k-means clustering to group data into clusters based on similarity. It first normalizes the data using the StandardScaler function from the scikit-learn library, which transforms the data to have a mean of 0 and a standard deviation of 1. This is done to ensure that each feature has equal weight in the clustering process.

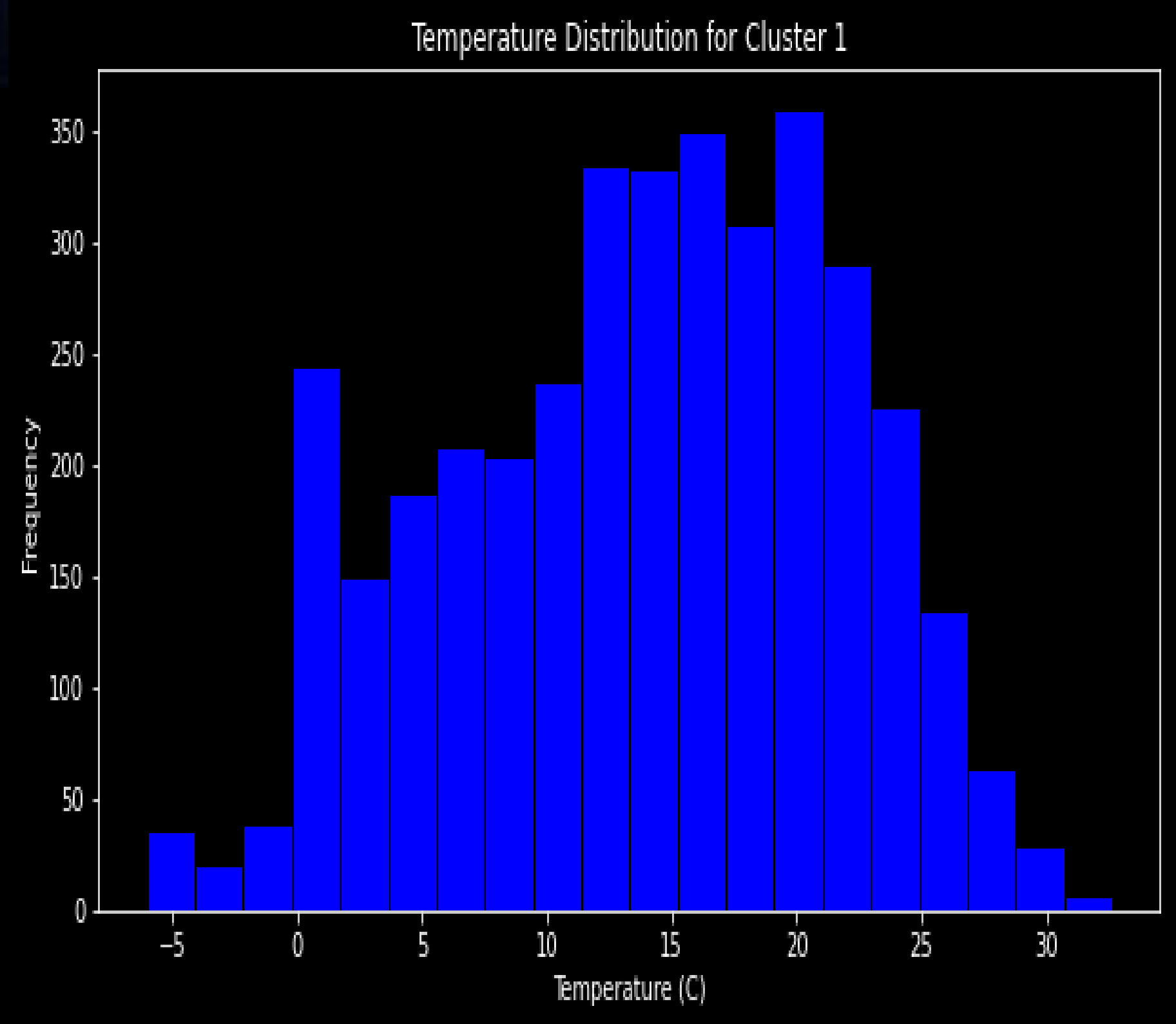
The KMeans function from the scikit-learn library is used to perform the clustering. In this case, three clusters are specified using the n_clusters parameter. The fit method is then called on the normalized data to train the k-means model, and the labels attribute is used to obtain the cluster labels for each data point.

The cluster labels are then added to the original dataframe, and the mean and describe methods are used to print summary statistics for each cluster. This provides information on how the data is partitioned into distinct groups, and can help to identify any patterns or trends within the data.

Finally, a scatter plot matrix is created to visualize the data with the cluster labels, and a bar chart is created to show the size of each cluster. The scatter plot matrix allows for the visualization of relationships between pairs of features and how they relate to the clusters. The bar chart provides a quick overview of the sizes of each cluster, allowing for an assessment of how balanced or imbalanced the partitioning of the data is.







This dataset containing temperature and wind speed information for different samples. The samples were divided into two clusters based on certain characteristics. To better understand the relationship between temperature and wind speed within each cluster, a scatter plot was created. The plot was designed to display both clusters with different colours and labels. Axis labels and a title were added for clarity. The scatter plot helped understand the relationship between temperature and wind speed in each cluster. Finally, the legend and plot were displayed using plt.legend() and plt.show() respectively. Overall, this visualization was a valuable tool for gaining insights into the data.

Once upon a time, there was a dataset that contained information about the temperature of different samples. The samples were divided into two distinct clusters based on certain characteristics. To better understand the temperature distribution within each cluster, histograms were created.

The code looped over each cluster and created a new DataFrame for each one. A histogram was then created for each cluster with a size of 10 inches in width and 5 inches in height.

The histograms were designed to display the frequency of temperatures with the x-axis representing "Temperature (C)" and the y-axis representing "Frequency". The color of the histograms was set to blue, and the edgecolor was set to black to distinguish between the bars. The title of each histogram clearly indicated the cluster number and the phrase "Temperature Distribution for Cluster". Thanks to this visualization, the temperature distribution within each cluster was much clearer and easier to understand. The histograms were displayed using the plt.show() function, making them easily accessible to those who wished to analyze the data.

An exponential growth model (exp_model) is defined using the NumPy library. The function takes three parameters, a, b, and c, and returns the product of a times e raised to the power of b times x plus c.

Next, the temperature data is extracted from the DataFrame (df) and saved as two NumPy arrays, xdata and ydata.

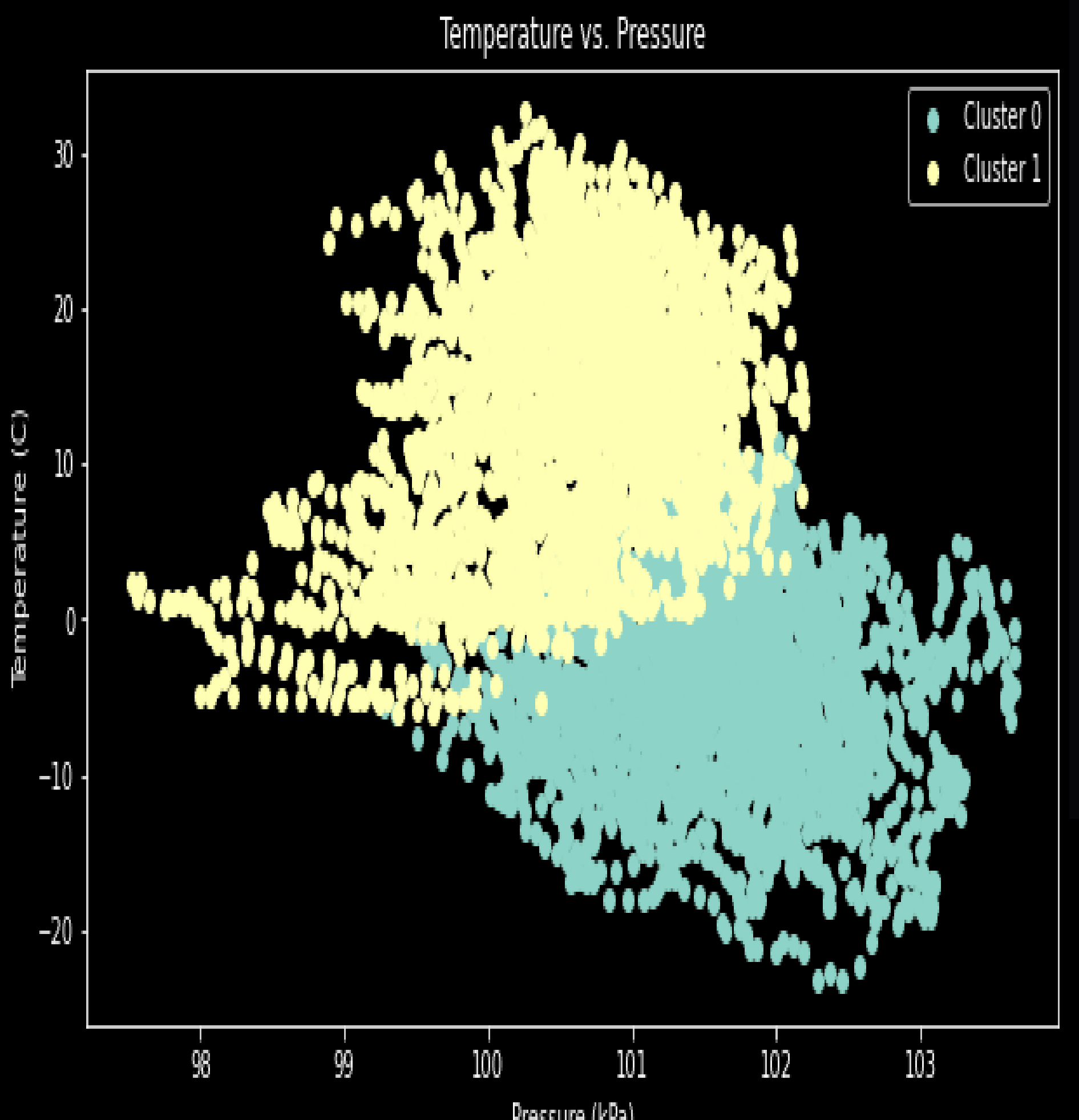
Then, the curve_fit function from the SciPy library is used to fit the exp_model to the temperature data. The resulting parameters of the best-fit model (popt) and the estimated covariance of popt (pcov) are stored.

After that, predictions are generated for the next 20 years using the exp_model and the best-fit parameters. The x-values for the predictions are generated using NumPy's arange function.

A function called err_ranges is defined to estimate the confidence range for the predictions. The function takes in the best-fit parameters, the estimated covariance of popt, and the xdata, and returns the range of the error in the predictions.

Finally, a scatter plot of the temperature data is created using the xdata and ydata. The best-fit function (ypred) and the confidence range (conf_range) are plotted using the xpred array generated earlier. The plot has axis labels, a title, and a legend.

A dataset with pressure and temperature information of different samples was divided into two distinct clusters. To better understand the relationship between pressure and temperature within each cluster, a scatter plot was created. The plot displayed both clusters with different colours and labels. Axis labels and a title were added for clarity. The scatter plot helped understand the relationship between pressure and temperature in each cluster.



Conclusion:

Overall, this code demonstrates various data analysis techniques, including data normalization, k-means clustering, scatter plots, histograms, and curve fitting. The project provides valuable insights into the relationship between temperature, pressure, and wind speed in weather data and the growth pattern of temperature over time.