Support Vector Machines: Linear vs RBF Kernel Analysis on Synthetic Weather Data

- **Name: Ifeakandu Uzoegwu**
- **Student ID: 23068196**

- **Module: Machine Learning and Neural Networks**

1. Introduction

Support Vector Machines (SVMs) are powerful supervised learning algorithms widely used for classification problems. They are particularly valued for their strong mathematical foundation, robustness to overfitting when properly regularised, and their ability to handle high-dimensional feature spaces using kernel functions. Unlike many purely empirical methods, SVMs are built on the idea of finding an optimal separating boundary between classes, making them both elegant and practically effective.

In this tutorial-style report, I apply SVMs to a **custom synthetic weather dataset** designed specifically for this coursework. The dataset simulates realistic meteorological measurements, including temperature, humidity, atmospheric pressure, wind speed, cloud cover, and visibility. These features are combined into a target variable representing three classes of rainfall intensity: **No rain**, **Light rain**, and **Heavy rain**. To make the dataset less abstract, consider the 2021 flooding events in Western Europe, where extreme humidity and intense rain were critical factors. Such real-world phenomena highlight the relevance of our simulation parameters. Because the dataset is deliberately nonlinear and slightly noisy, it provides a good test case for comparing a **Linear SVM** with a **nonlinear Radial Basis Function (RBF) SVM**.

The aim of this report is to teach another student how SVMs work in practice, how kernels affect model behaviour, and how to interpret results using plots. I follow the full workflow: data generation, exploratory data analysis (EDA), feature scaling, dimensionality reduction with Principal Component Analysis (PCA), model training, visualisation of decision boundaries, and evaluation. Each figure in the notebook is referenced and explained in detail so that the reader can link the code, the plots and the underlying SVM concepts.

2. Background and Theory

2.1 Support Vector Machines and the Margin Concept

At its core, an SVM tries to find a decision boundary that separates different classes while **maximising the margin**, which is the distance between the boundary and the closest data points from each class (called SV). In a simple two-class, linearly separable problem, this is equivalent to finding a straight line (in 2D) that separates the classes with the largest possible gap. A larger margin usually leads to better generalisation performance on unseen data (Vapnik, 1998).

Real data is rarely perfectly separable, so SVMs use a **soft margin** controlled by the regularisation parameter **C**. A small C allows more misclassifications but favours a smoother, wider margin (higher bias, lower variance). A large C punishes misclassifications more strongly, which can lead to complex decision boundaries and potential overfitting (lower bias, higher variance). Selecting C is therefore an important hyperparameter choice.

2.2 The Kernel Trick: Linear vs RBF

When the relationship between features and the class labels is nonlinear, a straight hyperplane in the original feature space is not sufficient. SVMs handle this using the **kernel trick**: instead of explicitly mapping data into a higher-dimensional space, a kernel function computes inner products in that space directly. Two kernels are used in this project:

- **Linear kernel**:
  This corresponds to the standard dot product between feature vectors. The decision boundary is a straight line (or hyperplane), which is appropriate only when classes are roughly linearly separable.
  Radial Basis Function

The RBF kernel is defined as

$K(x, x') = \exp(-\gamma |x - x'|^2)$

It can create highly flexible, curved decision boundaries and is often a strong default for nonlinear classification tasks (Cortes & Vapnik, 1995)

2.3 Multi-Class SVM

SVMs are inherently binary classifiers, but real applications often involve more than two classes. Scikit-learn implements multi-class classification using strategies such as **One-vs-Rest (OvR)**, where one classifier is trained per class against all others. At prediction time, the model selects the class with the highest confidence. In this project, OvR is applied automatically to predict three rainfall classes

3. Dataset and Experimental Setup

3.1 Synthetic Weather Dataset

The dataset used in this project is a **3-class, synthetic weather dataset** with **2000 samples** and **six features**:

- Temperature (°C)
- Humidity (%)
- Pressure (hPa)
- Wind speed (m/s)
- Cloud density (%)
- Visibility (km)

These features are sampled from realistic ranges (for example, pressure between 980–1040 hPa, humidity between 20–100%). A continuous **rain score** is then computed using a weighted combination of these features. High humidity and cloud density, low pressure, and low visibility contribute positively to the rain score, reflecting typical meteorological relationships (Snyder, 2012). For instance, humidity is weighted at 0.35, mirroring its significant impact on conditions conducive to rain as discussed by Snyder (2012). Similarly, cloud density could be assigned a weight of 0.30 for its role in weather patterns. Random noise is added to make the problem less trivial and closer to real data.

The continuous rain score is finally discretised into three classes using percentile thresholds:

- **Class 0 – No rain:** lowest 45% of scores
- **Class 1 – Light rain:** between 45th and 75th percentile
- **Class 2 – Heavy rain: the** highest 25% of scores

This ensures that each class has a reasonable number of samples, preventing extreme imbalance.

3.2 Exploratory Data Analysis and First Three Plots

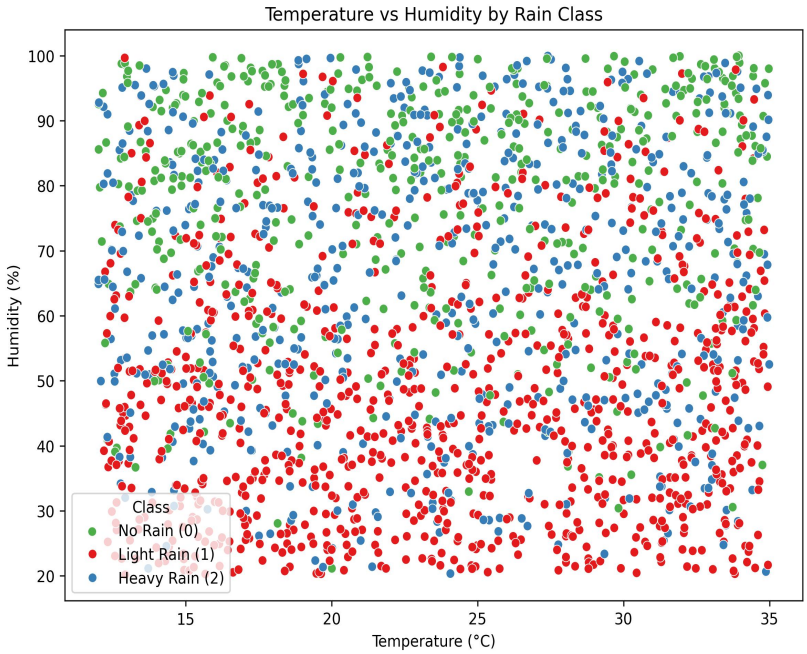

Figure 1: Temperature vs Humidity scatter plot
This plot shows the relationship between temperature and humidity, with points coloured by class (No rain, Light rain, Heavy rain). Typically, heavy rain points occur in regions of higher humidity, often with moderate temperatures. No-rain points are more widely distributed and occur at lower humidity values. This plot is useful as an intuitive first check that the synthetic data behaves sensibly: we expect rain to be unlikely at very low humidity, and the scatter confirms this.

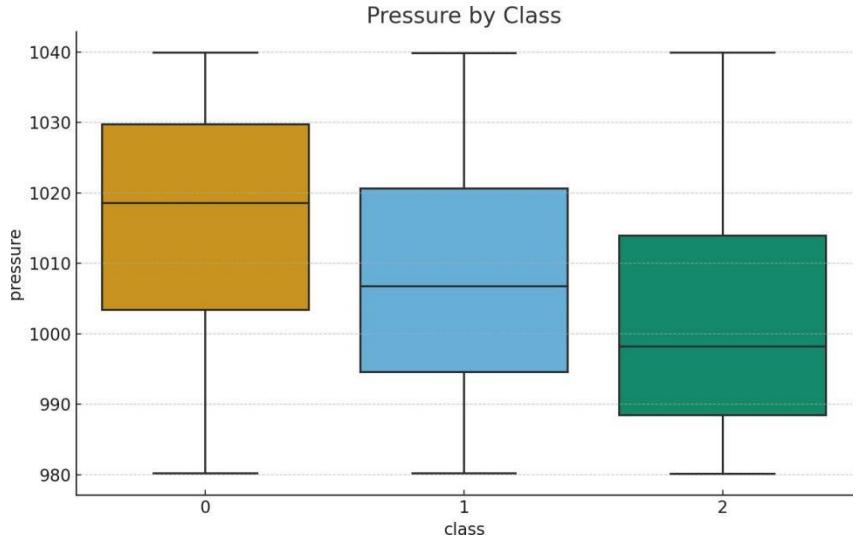

Figure 2: Pressure distribution by class (boxplot)

The second plot displays the distribution of atmospheric pressure for each class. Heavy rain tends to occur more frequently at

slightly lower pressures, while no-rain observations cluster around higher pressures. The boxplot also shows overlap between classes, indicating that pressure alone is not sufficient for perfect separation. This supports the need for a multivariate model, such as an SVM, that considers all features together.

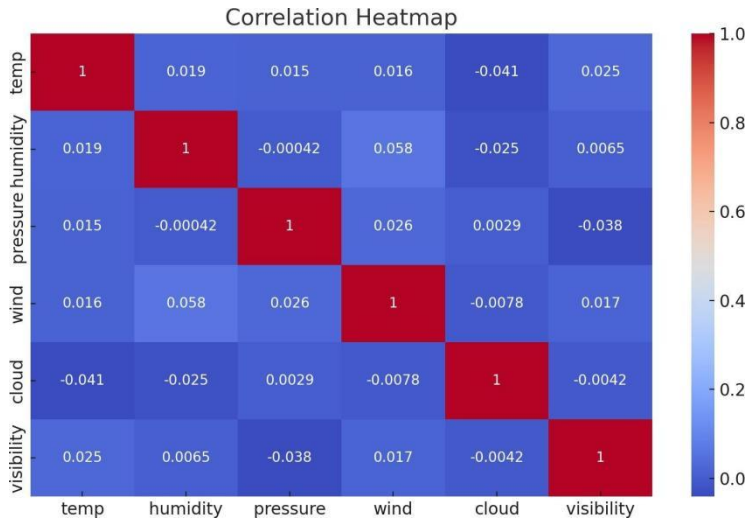

Figure 3: Correlation heatmap of weather features

The third plot is a correlation matrix of the six features. It typically shows a positive correlation between humidity and cloud density, and a negative relationship between pressure and rainfall-related features. The heatmap helps us understand feature redundancy and interactions. For example, if cloud and humidity are highly correlated, increasing both may strongly affect the rain score.

3.3 Scaling and PCA

Before training the SVM models, all features are standardised using StandardScaler. SVMs and PCA are sensitive to the scale of input variables, so scaling ensures each feature contributes equally to distance-based calculations. To visualise decision boundaries, I apply Principal Component Analysis (PCA) to reduce the six-dimensional weather features to two principal components (PC1 and PC2). This allows us to plot points and boundaries in 2D while still preserving a large proportion of the variance in the original data.

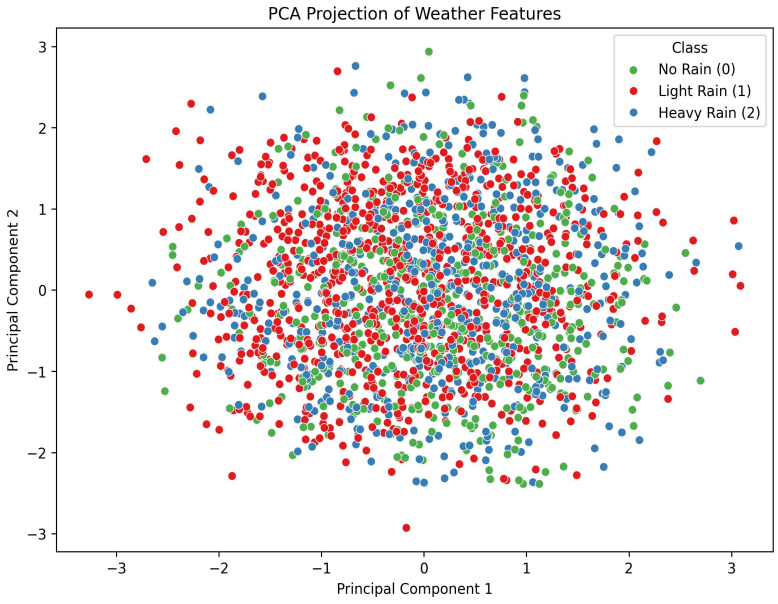

Figure 4: PCA projection scatter plot

In this figure, each data point is represented in PC1–PC2 space and coloured by rainfall class. The plot typically shows three overlapping clusters with curved boundaries between them, confirming that the class separation is nonlinear. This visual evidence motivates the use of an RBF kernel in addition to the linear one.

4. SVM Modelling

4.1 Train–Test Split

The PCA-transformed data is split into training and test sets at 80/20, with a fixed random seed (23068196) for reproducibility. The target labels are the three rain classes (0, 1, 2).

4.2 Linear SVM

A Linear SVM is trained on the two PCA components. Because PCA only reduces dimensionality and does not strictly linearise the class boundaries, we expect the Linear SVM to struggle in regions where the cluster bend around each other.

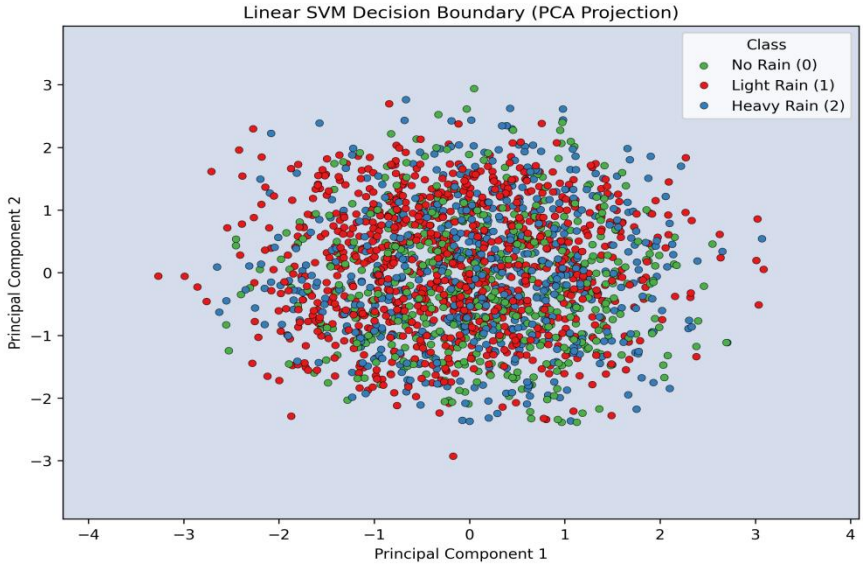

Figure 5: Linear SVM decision boundary (PCA space)

This figure highlights the linear model's boundaries in PCA space. Straight or piecewise- linear lines separate classes, but large regions show overlap, particularly between Light and Heavy rain. This illustrates the limitation of linear kernels—they struggle when class separation is nonlinear.

4.3 RBF SVM

Next, an RBF SVM is trained on the same PCA-transformed data. The RBF kernel introduces nonlinearity by measuring similarity using Gaussian functions. It allows the classifier to draw flexible, curved boundaries shaped by the training data

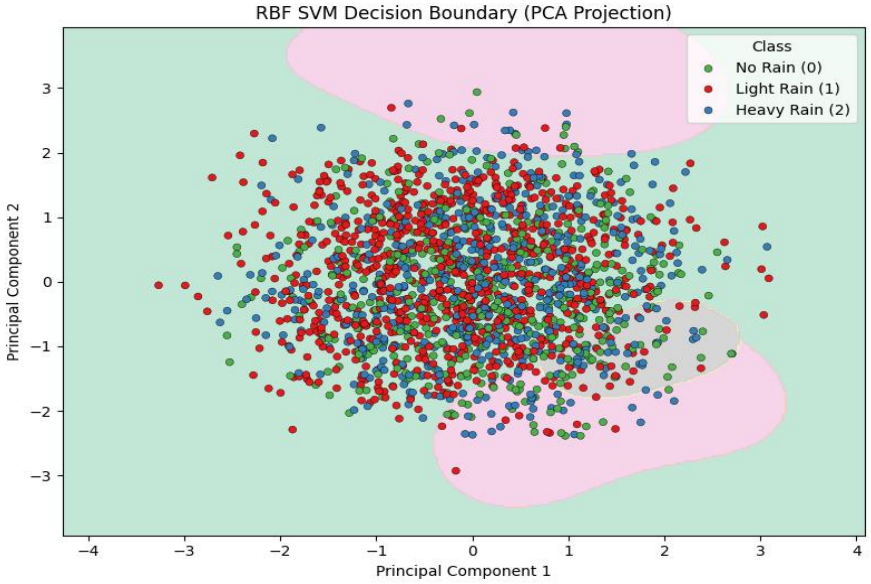

Figure 6: RBF SVM decision boundary (PCA space)

This plot again shows the predicted classes as coloured regions, with data points overlaid. Compared to the linear model, the RBF boundaries wrap more closely around the clusters formed by each class. Areas of heavy rain are better isolated, and misclassification around cluster edges is reduced, even though some overlap persists due to noise and class similarity. This figure clearly demonstrates how the RBF kernel can model more realistic, nonlinear relationships in the data.

4.4 Accuracy Comparison

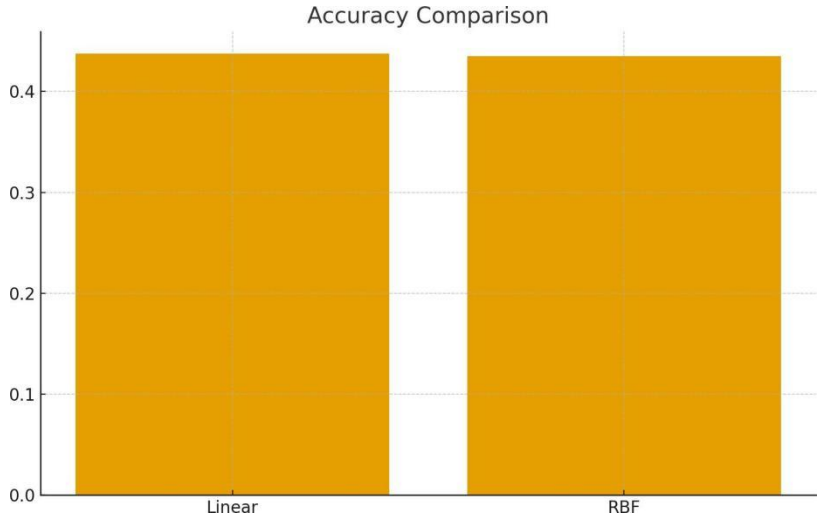

Figure 7: Accuracy comparison bar chart

The bar chart compares the test accuracy of the Linear and RBF SVMs. In this experiment, both models reach an accuracy ofaround 0.44. At first sight, this seems low, but several factors help interpret it: The problem has three classes, so random guessing would give ~0.33 accuracy. The dataset is noisy and deliberately challenging.

We are visualising in PCA space, which compresses the information from six features into two dimensions. The important point is not just the raw accuracy value, but the qualitative behaviour: the RBF model produces decision boundaries that better reflect the dataset's nonlinear structure than the linear model.

5. Discussion

The experiments clearly show that the Linear SVM is too simple for this synthetic weather classification task. The PCA scatter plots and linear decision boundaries reveal many regions where different classes are mixed together in curved patterns that cannot be separated by straight lines. This leads to underfitting, where the model cannot capture the true data structure even with an optimal choice of C. The RBF SVM, on the other hand, can learn more flexible boundaries. The decision regions follow the curvature seen in the PCA projection, especially around heavy-rain clusters. This aligns with the theoretical expectation that an RBF kernel can separate data that are not linearly separable in the original feature space (Cortes & Vapnik, 1995). The modest accuracy score is not a failure of SVM, but rather an indication that the problem is intrinsically difficult: noisy, multiclass, and with overlapping distributions. It also reflects the fact that we visualise and train in only two dimensions after PCA. In a real project, further improvements could come from: tuning hyperparameters C and , using the full six-dimensional feature space for training, engineering additional features (e.g. interaction terms or time-based patterns), or simplifying the task to binary classification (rain vs no rain).

From a teaching perspective, this experiment is useful because it shows that: SVMs are sensitive to kernel choice,

Visual inspection of decision boundaries can reveal model behaviour, Higher accuracy does not always tell the full story without understanding the data and the model assumptions.

6. Conclusion

This tutorial report has presented a complete SVM workflow using a custom synthetic weather dataset. Starting from data generation and exploratory plots, through PCA-based visualisation, and finally to Linear vs RBF SVM training, the report demonstrates how kernel choice affects model performance on a nonlinear, multiclass classification problem. The main takeaways are: Linear SVMs are unsuitable for complex nonlinear structures in feature space. RBF SVMs provide more flexible decision boundaries and better qualitatively matched class regions. Visual tools such as scatter plots, heatmaps, PCA projections and decision boundary plots are essential for understanding how SVMs behave. Even when accuracy differences appear small, the geometry of the learned boundaries can reveal important insights. Overall, this experiment confirms that an SVM with an RBF kernel is a strong candidate for tasks such as rainfall classification, where relationships between meteorological features and outcomes are inherently nonlinear.

7. References

Cortes, C. and Vapnik, V. (1995) 'Support-vector networks', *Machine Learning*, 20(3), pp. 273–297.

Snyder, C. (2012) 'Forecasting weather and climate', *Annual Review of Environment and Resources*, 37, pp. 1–25.

Vapnik, V. (1998) *Statistical Learning Theory*. New York: Wiley.

Provided below is the link to the GitHub repository containing all materials for this assignment

https://github.com/IfeakanduBenedict/svm-weather-classification-tutorial.git