# Modelling and Evaluation Plan

## Problem Statement

My aim in this project is to build a predictive model to forecast whether it will rain tomorrow in a specific region using the “[Weather Dataset - Rattle Package](https://www.kaggle.com/datasets/jsphyg/weather-dataset-rattle-package/data)” dataset. Our goal is to achieve the highest possible accuracy in predicting rain, and I am preparing this document to evaluate various machine learning models to accomplish this task.

## Model Selection

There are multiple machine learning models, and we need to find the one that performs best on our dataset. These models include:

1. **Logistic Regression:** This will serve as our baseline model due to its simplicity.
2. **Decision Trees:** These can capture non-linear relationships in the data.
3. **Random Forest:** An ensemble method that combines multiple decision trees.
4. **Support Vector Machines (SVM):** Useful for both linear and non-linear classification problems.
5. **Gradient Boosting:** Models like XGBoost or LightGBM which are powerful ensemble methods.
6. **Neural Networks:** Deep learning models for complex pattern recognition.

## Evaluation Metrics

To assess the performance of these models, we can use the following evaluation metrics:

1. **Accuracy:** To measure the overall correctness of the model's predictions.
2. **Precision:** To determine the proportion of positive predictions that were correct.
3. **Recall:** To calculate the proportion of actual positives that were correctly predicted.
4. **F1-Score:** To find a balance between precision and recall.
5. **ROC AUC:** To evaluate the model's ability to distinguish between positive and negative classes.

In cases of class imbalance, accuracy can be misleading because the model might achieve high accuracy by predicting the majority class most of the time, while failing to capture the minority class. In such scenarios, it's important to consider additional metrics such as precision, recall, F1-score, and the Area Under the ROC Curve (AUC-ROC), which provide a more comprehensive understanding of the model's performance, especially when false positives and false negatives have different consequences.

We will select these metrics based on the nature of the problem (binary classification) and its specific requirements, which may prioritize precision over recall or vice versa.

The choice between precision and recall for my binary classification problem depends on the specific objectives and constraints of the application.

1. **Precision:** I will prioritize precision if the False Positives are Costly - If predicting rain when it won't rain (false positives) has significant consequences, such as causing unnecessary resource allocation or disruptions, then we should prioritize precision. In this case, we want to be very sure that when the model predicts rain, it's highly likely to be accurate.
2. **Recall:** I will prioritize recall if the False Negatives are Costly - If failing to predict rain when it will rain (false negatives) has substantial consequences, like missing important rainfall events with potential impacts on agriculture, water resource management, or safety, then we should prioritize recall. Here, we want to capture as many instances of actual rain as possible.
3. **Balanced Priorities:** If the costs of false positives and false negatives are relatively balanced and there is no strong preference for one over the other, we can aim for a balanced F1-score, which combines precision and recall.

Ultimately, the choice between precision and recall should align with the goals and priorities of the project and the potential impacts of the model's predictions in the real world.

## Model Evaluation / Fine-Tuning

1. **Data Splitting:** We will split the dataset into training and testing sets. The training set will be used to train the models, and the testing set will be used for evaluation. We may also consider using techniques like cross-validation for more robust evaluation to assess generalization to unseen data.
2. **Hyperparameter Tuning:** We will optimize the hyperparameters for each model using techniques such as grid search, random search, or Bayesian optimization.
3. **Feature Importance:** For models like Decision Trees and Random Forest, we will examine feature importance to understand which features have the most significant impact on the predictions.
4. **Ensemble Techniques:** We will explore ensemble methods like stacking or bagging to combine the predictions of multiple models.
5. **Regularization:** If necessary, we will apply regularization techniques to prevent overfitting.
6. **Model Interpretability:** For better understanding and trust in the model, we will examine the interpretability of the selected model(s).

## Model Comparison

To select the final model, we will:

1. Create a summary table or visualization of the evaluation metrics for all models.
2. Analyze trade-offs between accuracy, precision, recall, and AUC-ROC.
3. Consider the interpretability and computational cost of each model.

## Model Selection

Make a well-informed decision based on the collective analysis of evaluation metrics, trade-offs, interpretability, and cost.

1. Compare the performance of all models using the evaluation metrics mentioned above.
2. Consider the trade-offs between accuracy, precision, recall, and other metrics based on the project's objectives.
3. Consider the interpretability of the model if it is essential for decision-making in the application.
4. Assess the computational cost and scalability of the model, especially for large-scale deployment.

The final model will be the one that offers the best balance of performance, interpretability, and suitability for the application.