# Final Project Report

**Yuxiao Yao**

## 1. Who is your stakeholder

My stakeholders are meteorological departments that require accurate weather forecasting, such as agricultural businesses, airlines, and event planners, who rely on precise weather forecasts to make informed decisions.

## 2. What is the problem they are trying to solve

They are trying to accurately predict whether it will rain the next day in specific regions. This prediction is crucial for planning agricultural activities, scheduling flights, and organizing events.

## 3. Where your dataset is from (link to it or include it in your submission)

My dataset is sourced from the https://www.kaggle.com/datasets/jsphyg/weather-dataset-rattle-package , which contains daily weather observations from various regions across Australia.

## 4. What models did you try, why did you choose those models?

I experimented with two models: Random Forest and SVM. These models were chosen because:
**Randomforest**: Suitable for classification tasks, performs well with datasets having a large number of input variables, and is robust to outliers, making it applicable given the diverse features of our dataset. Randomforest can also deal with the imbalanced data.
**SVM:** Effective at handling non-linear decision boundaries, the kernel trick allows for handling multi-dimensional data efficiently, which helps in improving model accuracy. SVM can also handle imbalanced data

## 5. What features did you select/engineer? How did you choose those?

1. The category features I chosed are: ['Evaporation', 'Sunshine', 'WindDir3pm', 'WindDir9am', 'WindGustDir', 'RainToday'] Because from living experience, these features can directly effect tommorrow's weather. I use OnehotEncoder to transform them. For the feature 'Location', I use LabelEncoder to transform it.

2. For numeric features I chosed ['Cloud3pm', 'Cloud9am', 'Humidity3pm', 'Humidity9am', 'MaxTemp', 'MinTemp', 'Pressure3pm', 'Pressure9am', 'Rainfall', 'Temp3pm', 'Temp9am', 'WindGustSpeed', 'WindSpeed3pm', 'WindSpeed9am', 'location_encoded']. These features were chosen based on their direct impact on weather prediction. I use standardscaler to transform them into same scale for training.
3. For missing data, first I delete the rows where the df['RainTomorrow'] = nan. Because 'RainTomorrow' column is my objective data. This is a supervised model so I can't have missing values with the objective column. Then for column ['MinTemp', 'MaxTemp', 'Humidity3pm', 'Pressure9am', 'Pressure3pm', 'Temp9am', 'Temp3pm'], I fill the missing data with mean because they have nomal distribution. For column ['Rainfall', 'WindGustSpeed', 'WindSpeed9am', 'WindSpeed3pm', 'Humidity9am', 'Pressure9am', 'Cloud9am', 'Cloud3pm', 'RainToday'] wich don't have nomal distribution I filled the missing data with median(numeric data) and mode(category data)

## 6. How did you evaluate the model? What evaluation metrics did you use? Why?

The model evaluation was primarily based on accuracy, precision, recall, and F1 score. These metrics help understand how well the model performs in predicting rain:
**Accuracy**: Measures the overall ability of the model to correctly predict.
**Precision and Recall**: Important to not miss any actual rain events (high recall) while minimizing false alarms (high precision).
**F1 Score**: The harmonic mean of precision and recall, used to balance between the two.

## 7. What would you do different next time or given more time what would your future work be?

Given more time, I would explore additional feature engineering and other advanced models, such as deep learning networks, which might perform better in capturing complex patterns in the data. Additionally, I would conduct more extensive parameter tuning and use cross-validation to optimize model performance.

## 8. Do you recommend your client use this model? Is the precision/recall good enough for the intended use case?

I would recommend my client use the SVM model which the parameters are: classifier = SVC(kernel='rbf', C=1.0, gamma='scale', random_state=66,class_weight='balanced').
**Randomforest**:For my RandomForest modle, the Training Accuracy are very high(99%). But the recall and f1 score for Not rain tomorrow are pretty low. Indicate that the model

is over fitting.
**SVM**: for my SVM model, when I set kernel='rbf', class_weight='balanced'. The outperforms this model in both precision and recall for non-rainy days, indicating it is more accurate and reliable in identifying and predicting non-rainy conditions.
For rainy days (Class 1), SVM has a significantly higher recall than RandomForest, meaning SVM is better at recognizing impending rain.

So the SVM model would be the preferred choice for practical application, especially in scenarios where balancing false alarms and missed predictions is necessary.