CPSC 66 Final Report: Optimizing Next-Day Rainfall Predictions in Australia Using Machine Learning Models

Syed Ali SALI3@SWARTHMORE.EDU
Kisanet Gabreselassie KGABRES1@SWARTHMORE.EDU
Sonja Rebarber SREBARB1@SWARTHMORE.EDU
Marcus Wright BWRIGHT1@SWARTHMORE.EDU

Abstract

This project uses ten years of daily weather data from Australia to find the best machine-learning models for predicting whether or not it rains the next day. We compare KNN, Logistic Regression, Decision Trees, and Random Forests performance under a range of preprocessing conditions to understand how data cleaning affects model performance. These conditions include different strategies for handling missing values, altering the feature set, and examining the impact of different representations of time, location, and other discrete features such as wind direction. Through exploring various data preprocessing techniques, we aimed to identify the most accurate model and preprocessing pipeline for our chosen dataset. Ultimately, we found that despite implementing various data processing techniques, there was minimal change in the accuracy of the models.

1. Introduction

One of the most important components of weather prediction is precipitation. Up until the 21st century, the primary method of predicting precipitation was numerical weather predictions (NWP), which used equations based on atmospheric physics to simulate and predict weather patterns (Reilly, 2024). Forecasters have used multiple different NWP models, but the equations that NWP models rely on have built-in systematic errors in predicting weather patterns. Moreover, when compounded with issues related to precipitation parameterization and physical world applications, these errors become near impossible to reduce (Yang et al., 2024)

CPSC 66 Machine Learning Proceedings, Swarthmore College, Spring 2021.

In recent years, Machine Learning (ML) has emerged as a promising tool in weather prediction by uncovering complex relationships that traditional NWPs cannot capture. Studies have shown that models such as ClimaX and GraphCast that incorporate ML techniques outperform NWPs in accurately predicting weather forecasts (Liu et al., 2023). Research has continued in this field to find the best machine learning-based weather prediction (MLWP). ML models have historically been treated as black boxes and trained on vast amounts of weather data (Bochenek & Ustrnul, 2022). It's important to note, however, that the quality of data preparation significantly impacts the performance of the models. Specifically, the model's performance greatly improves with various preprocessing techniques such as feature selection, class balancing, and imputation.

In this project, we aim to predict next-day rainfall in Australia based on weather-related features from a specific dataset. To address this challenge, we are utilizing a variety of machine learning algorithms—including K-Nearest Neighbors (KNN), Logistic Regression, Decision Trees, and Random Forests—and analyzing different data engineering approaches to maximize the accuracy of predicting next-day rainfall while providing full transparency into the underlying logic. We explore a range of imputation methods, location and time encodings, and feature selection strategies to enhance model performance. Our project provides valuable insights into how preprocessing techniques influence the accuracy and effectiveness of machine-learning models in predicting weather patterns.

2. Methods

Our dataset, "Rain in Australia," provides roughly a decade of daily observations of daily weather observations from numerous locations across Australia and was sourced from the Australian Bureau of Meteorology. The dataset includes various weather features such as date, location, minimum temperature, maximum temperature, rainfall, evap-

oration, sunshine, wind gust direction, wind gust speed, and wind direction at 9am. Although the dataset is well-documented, after exploratory analysis some data cleaning and feature modification was necessary. We discovered the dataset had some missing values, categorical variables that needed encoding, and outliers that could negatively impact the performance of machine learning models.

2.1. Imputation

Missing data was addressed using four pre-processing methods: data removal, mean/mode imputation, predictive imputation, and a hybrid approach. The removal method, a straightforward strategy, involves eliminating entries with missing values (”NaN”). Mean/mode imputation replaces missing values with the mean for continuous variables or the mode for categorical variables, ensuring consistency with existing data but risking distortion from outliers and potential overfitting due to synthetic data. Predictive imputation, by contrast, estimates missing values using machine learning models trained specifically for each feature. A Random Forest Regressor was employed for continuous features, while a Random Forest Classifier was used for discrete features. These models were trained on the remaining data to predict the missing values, providing a context-aware solution tailored to local patterns. However, we observed that some predictive models exhibited poor performance for certain features. To address this, our hybrid method incorporated a certainty threshold: if the accuracy of a predictive model was below 50%, mean/mode imputation was used instead. This hybrid approach mitigated the limitations of both imputation methods, ensuring a more reliable and robust handling of missing data.

2.2. Feature Correlation and Removal

We conducted a detailed analysis of feature correlations to identify potential issues with multicollinearity and redundancy. Even with a limited set of features, multicollinearity can still distort model performance by inflating standard errors and weakening the reliability of feature estimates. Through the use of correlation matrices and visualizations, see Figure 1, we identified several highly correlated features, including temperature at 9am, temperature at 3pm, pressure at 9am, pressure at 3pm, minimum temperature and maximum temperature, which were likely to provide overlapping information.

To address these correlations, we implemented an approach that involved removing certain highly correlated features to observe their effect on model performance. Preliminary results indicated that removing some of these highly correlated features resulted in better model performance by reducing overfitting and decreasing redundancy in the data. This approach helped refine the feature set, ensuring the model retained only the most informative and non-redundant features, which contributed to enhanced prediction accuracy.

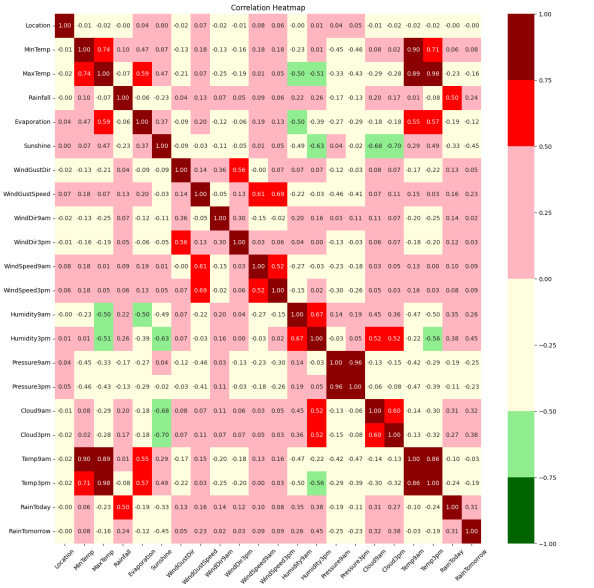

Figure 1. Correlation Matrix for all features in the dataset. Dark red boxes signify a higher correlation between features.

2.3. Feature Engineering and Modification

Data preprocessing included label encoding, outlier handling, standardization, normalization, and transformations for circular and location-based features. Label encoding was applied to convert categorical variables, such as wind direction, into numerical values. To ensure consistency, identical labels (e.g., ”NE” for northeast) were encoded with the same numerical value across different features, preserving interpretability and alignment in the data representation. This uniformity helps maintain consistency in downstream analysis, particularly when comparing or aggregating information across features.

Outliers in continuous variables were handled by standardizing the data and removing instances with a z-score exceeding 3. Normalization followed, ensuring all features were scaled to a uniform range, facilitating comparability and improving model convergence.

Features with inherent cyclic properties, such as months and wind direction, were transformed to reflect their circular nature. For example, while January and December are linearly distant, they are temporally adjacent, with weather patterns in December often continuing into January. Similarly, wind directions (e.g., north and northeast) exhibit periodic relationships that are not captured well by linear encoding. To account for these patterns, such features were

mapped onto a circular basis using sine and cosine transformations. This representation preserves periodicity and ensures the encoded values accurately reflect their natural relationships, improving the interpretability and performance of models that rely on geometric or distance-based calculations.

For the discrete location feature, we converted each location to its corresponding longitude and latitude values. This approach acknowledges the spatial relationships between regions: areas closer in proximity may experience similar weather patterns, while more distant locations may differ significantly. Representing locations in this way allows models to capture these spatial dependencies more effectively, enhancing predictions by incorporating the influence of geographic proximity.

In this section, we outlined the preprocessing methods considered for our experiments, including label encoding, correlation handling, and feature transformations. These methods were tested in various configurations to assess their impact on model performance. The subsequent section details the experimental setup, including the different variations of these preprocessing techniques, and presents the results from our evaluation.

3. Experiments and Results

Our pipeline used stratified k-fold and an exhaustive grid search for hyperparameters, as outlined in Lab 4, to return five values of each precision metric for each of the four base learners we tested. We ran the pipeline on each preprocessed dataset, and we created a new dataset for each combination of imputation methods and feature engineering variations. The precision metrics were accuracy, no-precision, no-recall, yes-precision, and yes-recall, and the four base learners we used were scikit-learn's implementations of K-Nearest Neighbors, Random Forest, Decision Tree, and Stochastic Gradient Descent. The imputation methods and feature variations we used are outlined in the Methods section. Thus, our results include five values for each of the five precision metrics, across the four base learners, for each feature variation, and for each imputation method. These values represent how well each of these preprocessing combinations were able to predict rainfall the following day based on features describing weather from the previous day.

Out of all of our imputation and feature variations, none of them made a huge positive difference in model accuracy. Table 1 shows the greatest, smallest, and mean values for each performance metric across all imputation-variation combinations (averaged over all base learners). It's clear from the table that while the models produce fairly accurate results, the yes recall is fairly low, sometimes less than 50%. This means that the model demonstrates lots of false negatives, often predicting that there will not be rain tomorrow when really there is. This is likely due to a class imbalance, because Australia has more days without rain than with rain, so around 20% of our dataset has a target value of 'Yes' for rain tomorrow.

Performance Metric	Min	Mean	Max
Accuracy	0.813540	0.843823	0.853430
No precision	0.848500	0.862795	0.874305
No recall	0.915751	0.949809	0.956826
Yes precision	0.693918	0.737016	0.753249
Yes recall	0.411174	0.470720	0.517099

Table 1. Mean, minimum, and maximum score for each performance metric across all imputation methods and feature engineering combinations, averaged over all base learners.

However, the table also makes it clear that the range of values that appeared for each performance metric is fairly small, indicating that it's possible that our imputation and feature variations made little impact on any of the performance metrics. Thus, we decided to focus on accuracy because it is the performance metric which is most frequently used and balances both recall and precision. However, it's important to note that the other metrics have specific implications for this dataset. Specifically, it is noteworthy to mention the high "no-recall" value, reaching 95.68%. The high value indicates that there is a great number of false negatives, where the model fails to label the rainfall as 'yes' for the target. Similarly, there is a high "no precision" value which rashes up to 87.43%. This value indicates that when the model predicts a day to be rainy, it will most likely be accurate. Again, this can be accredited to the class imbalance in the target column.

For each combination of imputation methods and feature variations, we calculated the average accuracy averaged over all base learners. The vast majority of accuracy values are between 84% and 85.5%, with a few as low as 81%. The only exceptions are the three variations which represent bigger changes to our set of features, including removing the "Rain Today" feature, removing all features except for "Rain Today," and removing all features except for "Rain Today" and temperature features. None of these provided noticeable improvements in model performance.

Because most of the accuracy values are fairly similar, we concluded that none of the feature engineering variations or imputation methods we tried had a significant impact on model performance. This indicates that the dataset is fairly robust. Minor changes could be made to this dataset, intentionally or unintentionally, and it's unlikely that it would change the effectiveness of a machine learning model's ability to predict the target value of Rain Tomorrow. However, there were small variations in the average accuracies

for each imputation method, feature variation, and base learner, shown in Table 2.

Imputation Method	Mean Accuracy (%)
Removal	85.08
Hybrid	84.21
Regression/Classification	84.13
Mean/Mode	84.12

Feature Variations	Mean Accuracy (%)
rm-1	84.63
rm-temp2	84.62
rm-temp1	84.61
rm-pres2	84.59

Base Learner Models	Mean Accuracy (%)
Random Forest	85.30
SGD	84.25
KNN	84.14
Decision Tree	83.84

Table 2. Performance comparison of Different Imputation Methods, Feature Variations, and Base Learner Models on Mean Accuracy for Rainfall Prediction
- rm-1 represents removing temperature at 9am and maximum temperature
- rm-temp2 represents removing minimum and maximum temperature
- rm-temp1 represents removing temperature at 9am and 3pm
- rm-pres2 represents removing pressure at 3pm

As shown, there were minor differences in average accuracies in each of these categories. For example, removing NaN values instead of using an imputation method resulted in an average accuracy of more than 85%, less than 1% better than the next most accurate imputation, the hybrid method. The highest average accuracy for a variation was removing temperature at 9am and maximum temperature. (This variation was chosen because during the exploration phase, we found that in general, removing this combination of features during preprocessing allowed KNN to perform more accurately on the dataset. Feature variations were also chosen based on the combinations that allowed the other four base learners to perform most accurately individually, but these did not perform as well overall). However, this feature variation performs less than 0.01% more accurately than the next best variation, so this difference does not appear to be significant. Out of all base learners, Random Forest performed the best, more than 1% better than the next most accurate model.

At first glance, these results appear to be insignificant. To validate this, we performed a series of Kruskal-Wallis and ANOVA tests on these datasets. We used these statistical tests to compare our model across imputation methods, feature engineering variations, and base learner models, all separately. Both the Kruskal-Wallis and ANOVA tests are used to determine whether there are significant differences between the means of three or more groups. The ANOVA test is parametric, meaning that for results to be meaningful, the datasets must be normally distributed. The Kruskal-Wallis test is non-parametric, meaning that it does not assume that the datasets are normally distributed. The drawback of Kruskal-Wallis is that it is rank-based, meaning that it focuses only on the order of values, so information about magnitude is not taken into account. Some of the datasets we compared are close to normally distributed, while others are not, so we chose to perform both tests in case it provided a clearer picture of our data. We performed these two tests to compare mean accuracies for the four imputation methods, then separately for all of the feature variations, and then again for the four different base learners. The results of these tests are shown in Table 3.

Comparison	Kruskal-Wallis p-value	Statistically Significant?
Imputation methods	5.026 E-10	Yes
Feature engineering variations	0.02755	Yes
Different base learners	8.177 E-35	Yes

Comparison	ANOVA p-value	Statistically Significant?
Imputation methods	7.623 E-11	Yes
Feature engineering variations	0.005995	Yes
Different base learners	2.031 E-25	Yes

Table 3. Results for Kruskal-Wallis and Anova Statistical Tests

Both the Kruskal-Wallis test and the ANOVA test returned statistically significant p-values for all comparisons. According to the standard 5% confidence level, these values would tell us to reject the null hypothesis. In this case, our null hypothesis was that there was no difference between the two groups. Thus, both tests indicate that we can reject our null hypotheses. This implies that there is a difference between the set of imputation methods, the set of feature variations, and the set of base learners. It says that these preprocessing variables do, in fact, affect model accuracy. This statistically significant result is contrary to what we would believe by simply looking at the table of values. It's important to distinguish between statistical significance and practice significance. While these groups of datasets might have statistically significant differences between them, as humans we can tell that the difference between an 84% and 85% accuracy is marginal. It's also very important to note that our datasets do not fully meet the as-

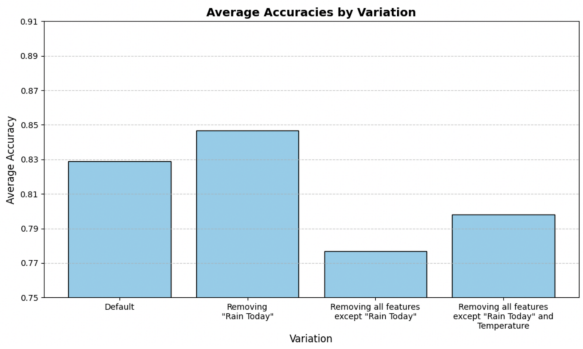

Figure 2. Performance Comparison Between Different Variations of the Dataset on Predicting Rainfall

sumptions of either the Kruskal-Wallis test or the ANOVA test. The ANOVA tests assume a normal distribution of data points, which some of our datasets do not have. Also, the Kruskal-Wallis test, while not requiring a normal distribution, focuses on the rank of values and not the magnitude, whereas for our purposes, the magnitude of values is very important. These unmet assumptions could easily explain our statistically significant results, when really we don't believe that these groups are meaningfully different from each other.

3.1. Special Variations

There were three variations which did have a meaningful impact on model accuracy, shown in Figure 2. These variations were different from the others in that these are big modifications to the feature set. These three variations were: 1) removing the "Rain Today" feature, 2) removing all features except for "Rain Today," and 3) removing all features except for "Rain Today" and the temperature features. We plotted the mean accuracy of these variations against the mean accuracy of the "default" variation. We considered the default to be the dataset with all features, discrete month values (1-12) and latitude-longitude values instead of discrete locations. The purpose of this analysis is to ensure that the model is using features other than "Rain Today" and temperature features to predict "Rain Tomorrow." None of the variations performed better than the default, other than the dataset with "Rain Today" removed, which performs marginally better. In fact, the other variations performed notably worse than the default, indicating that many of the features (not just Rain Today and temperature) are being fully "considered" by these machine learning models to predict rainfall.

4. Social Implications

This project contributes to the growing investment in applying machine learning (ML) to climate predictions. ML models are increasingly being used for weather forecasting and it is essential to consider both their opportunities and limitations. ML models can match and sometimes exceed the performance of NWP systems while being more computationally efficient. However, they often produce overly uniform forecasts, fail to capture localized weather events, and underestimate the intensity of tropical cyclones (Bouallègue et al., 2024). Given these historical shortcomings, it is crucial to combine the findings from our project with traditional NWP systems to prevent over-reliance on our ML predictions.

Some potential stakeholders for this project include farmers, meteorologists, and urban planners. Farmers, particularly in communities where agriculture is the primary means of survival or economic activity, could utilize our ML algorithms to better plan for planting and harvesting. Additionally, meteorologists could benefit from this project for more comprehensive tools for weather forecasting as ML models are deriving better accuracy (Liu et al., 2023). Similarly, urban planners could leverage our ML algorithms to make informed decisions regarding city planning, particularly for flood management and other rain-related risks.

With the rising trend of incorporating ML into climate predictions, there is a growing demand for technologists skilled in applying ML techniques to meteorology. Stakeholders stand to benefit significantly from access to reliable forecasts. However, as ML algorithms become increasingly integral to weather forecasting, professionals relying on traditional methods may face a skills gap and may require additional education on weather related data to better utilize ML models.

5. Conclusions and Future Work

This project aimed to build a ML model that could predict next-day rainfall in Australia based on the Australia Rainfall dataset. We explored different data engineering methods such as imputation, location and time encodings, and feature selection strategies to enhance model performance. Ultimately, we found that these different data preprocessing techniques did not have a significant impact on model performance.

There are a multitude of directions that this project could be extended. Our ML models were fine-tuned to this specific dataset, however, it could be interesting to see if this ML model could generalize to other rain datasets from Australia and/or other regions. This would help us understand the robustness of our model and if weather patterns in Australia are similar to other countries. Another way to extend our project is by addressing class imbalance. Since there were far more "No Rain" days than "Rain" days, the model

had a high no-recall value for "No Rain." Using techniques like oversampling or undersampling could help balance the classes and improve the model's performance. Moreover, an important aspect of weather forecasting is temporal patterns. While the dataset captures and predicts daily patterns, it does not account for interconnected temporal dependencies within a given time frame. Integrating models such a Long Short-Term Memory Network or a Recurrent Neural Network could better handle these dependencies.

Our work in this project raised several problems and questions. While this model predicts whether it will rain tomorrow, it does not address the question of how much rain can be expected. Extending the model to forecast the quantity of rainfall, rather than simply predicting if it will rain, could make the model more robust and useful. Additionally, an important challenge highlighted by this project is how the model would perform under extreme weather conditions, such as heavy rainfall or floods, given that machine learning models have historically struggled to accurately capture the severity of similar natural disasters.

This project showed both the potential and the limitations of using machine learning for rainfall prediction. While our models achieved reasonable performance, they also revealed opportunities for improvement, particularly in handling temporal dependencies, class imbalances, and extreme weather events. These findings emphasize the importance of continual refinement and adaptation of machine learning techniques to complex, real-world problems. As future advancements address these challenges, the insights gained from this study can contribute to the development of more accurate and reliable weather prediction systems, supporting critical decision-making processes in agriculture, disaster preparedness, and climate resilience.

Acknowledgments

We would like to express our gratitude to Professor Ben Mitchell for his invaluable guidance and insightful feedback throughout the course of this project. His expertise and direction were instrumental in shaping our approach and ensuring the success of this project.

References

Bochenek, B. and Ustrnul, Z. Machine learning in weather prediction and climate analyses—applications and perspectives. *Atmosphere*, 2022.

Bouallègue, Z. Ben, Clare, M. C. A., Magnusson, L., Gascón, E., Maier-Gerber, M., Janoušek, M., Rodwell, M., Pinault, F., Dramsch, J. S., Lang, S. T. K., Raoult, B., Rabier, F., Chevallier, M., Sandu, I., Dueben, P., Chantry, M., and Pappenberger, F. The rise of data-driven weather forecasting: A first statistical assessment of machine learning–based weather forecasts in an operational-like context. *Bulletin of the American Meteorological Society*, 105(6):E864–E883, 2024. doi: 10.1175/BAMS-D-23-0162.1. URL https://doi.org/10.1175/BAMS-D-23-0162.1.

Liu, Qi, Lou, Xiao, Yan, Zhongwei, Qi, Yajie, Jin, Yuchao, Yu, Shuang, Yang, Xiaoliang, Zhao, Deming, and Xia, Jiangjiang. Deep-learning post-processing of short-term station precipitation based on nwp forecasts. *Atmospheric Research*, 295:107032, 2023. ISSN 0169-8095. doi: 10.1016/j.atmosres.2023.107032. URL https://doi.org/10.1016/j.atmosres.2023.107032.

Reilly, Jon. Using machine learning for accurate weather forecasts in 2023, 2024. URL https://www.akkio.com/post/weather-prediction-using-machine-learning.

Yang, Ruyi, Hu, Jingyu, Li, Zihao, Mu, Jianli, Yu, Tingzhao, Xia, Jiangjiang, Li, Xuhong, Dasgupta, Aritra, and Xiong, Haoyi. Interpretable machine learning for weather and climate prediction: A review. *Atmospheric Environment*, 338:120797, 2024. ISSN 1352-2310. doi: 10.1016/j.atmosenv.2024.120797. URL https://doi.org/10.1016/j.atmosenv.2024.120797.