Predicting Rain Tomorrow in Australia

Introduction:

Exploring the machine learning techniques by tackling a real-world problem: predicting whether it will rain tomorrow in various locations across Australia. Weather forecasting is a crucial application of machine learning, with far-reaching implications for agriculture, transportation, and public safety.

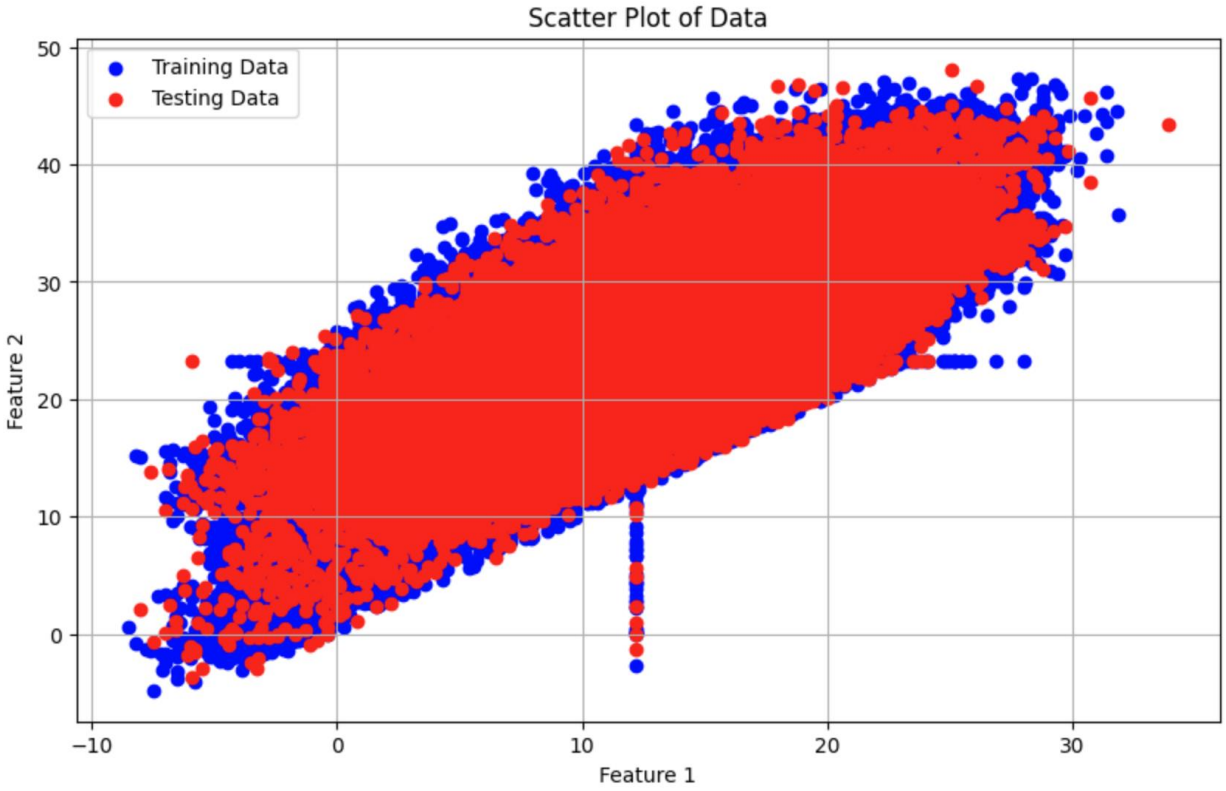
Dataset Description:

The dataset contains approximately 10 years of daily weather observations from multiple locations across Australia. Each observation includes various features such as temperature, humidity, wind speed, and rainfall.

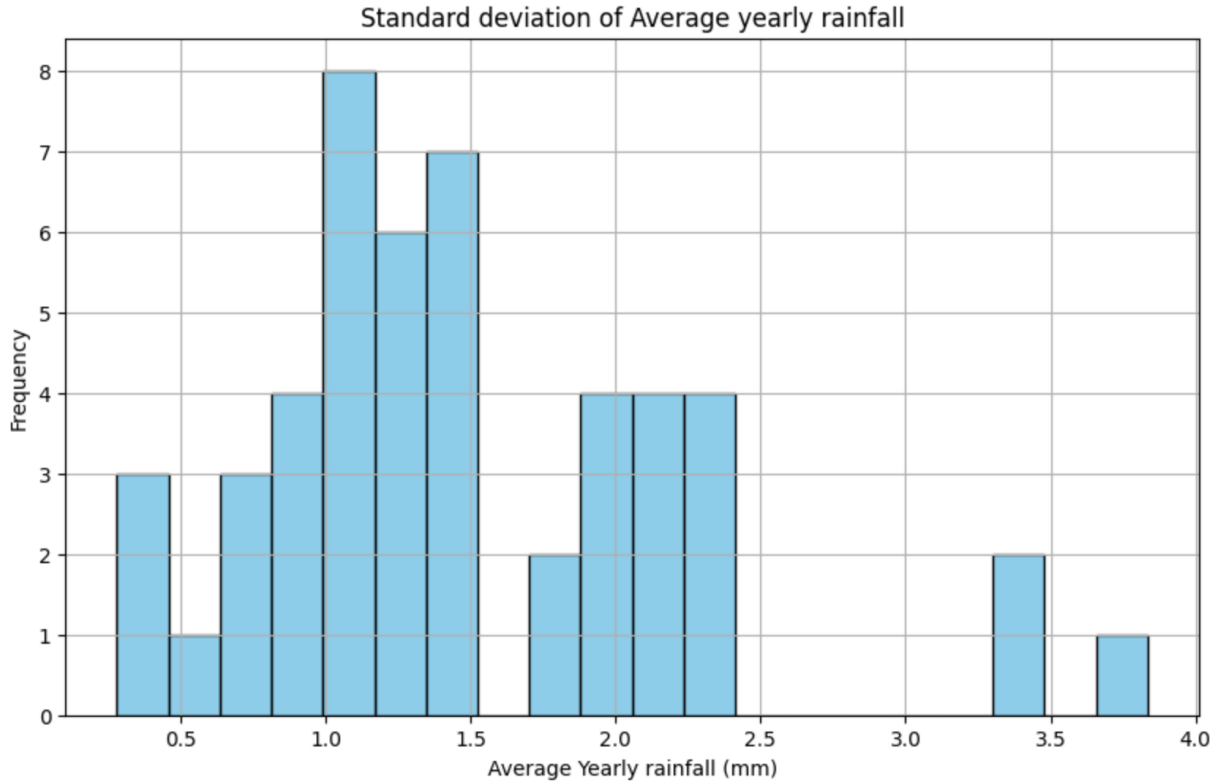
The target variable, Rain Tomorrow, indicates whether it rained the following day, with a binary classification of "Yes" or "No". Specifically, if the rainfall for a given day exceeds 1mm, Rain Tomorrow is labeled as "Yes."

Preprocessing:

The given data contains non numerical and also missing values. In order to handle this, I used mean and mode to handle missing values and also, I drop some non-numerical columns. Using a scatter plot we can also see how the test and training dataset looks like bellow.



We have more tasting data as expected. We can also do scaling to ensures that all features have the same scale, preventing certain features from dominating the learning algorithm and reduces the number of features in a dataset while preserving its essential information using dimension reduction. I use correlation matrix to visualize how the data of each column are related to each other.



The histogram indicates the frequency of average yearly rainfall across different locations. Most locations receive low rainfall, with the majority falling within the 0 to 8 mm range. Only one location experiences notably higher rainfall, averaging around 3.5-4.0 mm annually. The x-axis spans from 0 to 4.0 mm, reflecting the range of average yearly rainfall values.

Models comparation and evaluation

Accuracy: Overall accuracy of the model, which is the ratio of correctly predicted instances to the total instances

Precision: Precision is the ratio of correctly predicted positive observations to the total predicted positives.

Recall (also known as sensitivity): Recall is the ratio of correctly predicted positive observations to all observations in actual class.

F1-score is the harmonic mean of precision and recall. It provides a balance between precision and recall.

Confusion matrix is a table often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.

1, Decision tree classification model:

After we train the model, we get around 76% of testing accuracy but after improving the model using post pruning its test accuracy increases to 82.8%.

Before

classification report

	precision	recall	f1-score	support
0.0	0.85	0.85	0.85	22672
1.0	0.47	0.48	0.47	6420
accuracy			0.77	29092
macro avg	0.66	0.66	0.66	29092
weighted avg	0.77	0.77	0.77	29092

After post pruning

Classification Report:

	precision	recall	f1-score	support
0.0	0.85	0.95	0.90	22672
1.0	0.69	0.40	0.50	6420
accuracy			0.83	29092
macro avg	0.77	0.67	0.70	29092
weighted avg	0.81	0.83	0.81	29092

Confusion Matrix:
[[21541 1131] [3871 2549]]

- 21541 instances were correctly classified as negative.
- 1131 instances were incorrectly classified as positive when they were actually negative.
- 3871 instances were incorrectly classified as negative when they were actually positive.
- 2549 instances were correctly classified as positive.

Strength: after improvement

- It is good on predicting actual false values with 85%.
- 95% of the actual negative instances were correctly classified as negative.

Weakness:

- Only 40% of the actual positive instances were correctly classified as positive.

2, Naïve bayes Classification model:

Accuracy before applying feature selection and Hyperparameter Tuning using GridSearchCV, its test accuracy was 80.4% but after improvement of the model its test accuracy increased to 82.2%. This accuracy was improved by selecting 4 features.

Before

classification report

	precision	recall	f1-score	support
0.0	0.85	0.90	0.88	22672
1.0	0.57	0.46	0.51	6420
accuracy			0.80	29092
macro avg	0.71	0.68	0.69	29092
weighted avg	0.79	0.80	0.80	29092

After

classification report

	precision	recall	f1-score	support
0.0	0.85	0.94	0.89	22672
1.0	0.67	0.39	0.49	6420
accuracy			0.82	29092
macro avg	0.76	0.67	0.69	29092
weighted avg	0.81	0.82	0.80	29092

Confusion Matrix:
[[21415 1257] [3909 2511]]

- 21415 instances were correctly classified as negative.
- 1257 instances were incorrectly classified as positive when they were actually negative.
- 3909 instances were incorrectly classified as negative when they were actually positive.
- 2511 instances were correctly classified as positive.

Strength: after improvement

- It is good on predicting actual false values with 85%.
- 95% of the actual negative instances were correctly classified as negative.

Weakness:

- Only 39% of the actual positive instances were correctly classified as positive.

3, SVM Classification mode:

The accuracy before applying RandomizedSearchCV was 83.8% but after model improvement using RandomizedSearchCV its accuracy increases in to 84%

classification report

	precision	recall	f1-score	support
0.0	0.85	0.96	0.90	22672
1.0	0.75	0.40	0.52	6420
accuracy			0.84	29092
macro avg	0.80	0.68	0.71	29092
weighted avg	0.83	0.84	0.82	29092

Classification Report

	precision	recall	f1-score	support
0.0	0.85	0.96	0.90	22672
1.0	0.75	0.41	0.53	6420
accuracy			0.84	29092
macro avg	0.80	0.69	0.72	29092
weighted avg	0.83	0.84	0.82	29092

Confusion Matrix:
[[21753 919] [3734 2686]]

- 21753 instances were correctly classified as negative.
- 919 instances were incorrectly classified as positive when they were actually negative.
- 3734 instances were incorrectly classified as negative when they were actually positive.

- 2686 instances were correctly classified as positive.

Strength: after improvement
- It is good on predicting actual false values with 85%.
- 96% of the actual negative instances were correctly classified as negative.

Weakness:
- Only 41% of the actual positive instances were correctly classified as positive. but it is compared to the above two models.
- Takes more time to run.

4, Logistic regression model:

The accuracy before applying GridSearchCV was 83.38% but after model improvement using GridSearchCV its accuracy increases in to 83.89%

classification report

	precision	recall	f1-score	support
0.0	0.86	0.95	0.90	22672
1.0	0.70	0.44	0.54	6420
accuracy			0.83	29092
macro avg	0.78	0.69	0.72	29092
weighted avg	0.82	0.83	0.82	29092

classification report

	precision	recall	f1-score	support
0.0	0.86	0.95	0.90	22672
1.0	0.70	0.44	0.54	6420
accuracy			0.83	29092
macro avg	0.78	0.69	0.72	29092
weighted avg	0.82	0.83	0.82	29092

Confusion Matrix:
[[21435 1237] [3595 2825]]
- 21435 instances were correctly classified as negative.
- 1237 instances were incorrectly classified as positive when they were actually negative.
- 3595 instances were incorrectly classified as negative when they were actually positive.
- 2825 instances were correctly classified as positive.

Strength: after improvement
- It is good on predicting actual false values with 86%.
- 95% of the actual negative instances were correctly classified as negative.

Weakness:
- Only 44% of the actual positive instances were correctly classified as positive. It is still better than the previous 3 models.

5, Multi-layer Perceptron (MLP)

The accuracy before applying GridSearchCV was 83.9% but after model improvement using GridSearchCV its accuracy increases in to 84.2%

Before After

classification report

	precision	recall	f1-score	support
0.0	0.87	0.93	0.90	22672
1.0	0.66	0.49	0.56	6420
accuracy			0.83	29092
macro avg	0.76	0.71	0.73	29092
weighted avg	0.82	0.83	0.82	29092

classification report

	precision	recall	f1-score	support
0.0	0.86	0.95	0.90	22672
1.0	0.71	0.46	0.56	6420
accuracy			0.84	29092
macro avg	0.78	0.70	0.73	29092
weighted avg	0.83	0.84	0.83	29092

Confusion Matrix:
[[21232 1440] [3145 3275]]

- 21232 instances were correctly classified as negative.
- 1440 instances were incorrectly classified as positive when they were actually negative.
- 3145 instances were incorrectly classified as negative when they were actually positive.
- 3275 instances were correctly classified as positive.

Strength: after improvement

- It is good on predicting actual false values with 86%.
- 95% of the actual negative instances were correctly classified as negative.

Weakness:

- Only 46% of the actual positive instances were correctly classified as positive. But it is still better than the previous 4 models.

6, Ensemble methods using AdaBoost

The accuracy before applying GridSearchCV was 83.2% but after model improvement using GridSearchCV its accuracy increases in to 83.4%

Before

Classification Report:

	precision	recall	f1-score	support
0.0	0.86	0.94	0.90	22672
1.0	0.68	0.45	0.54	6420
accuracy			0.83	29092
macro avg	0.77	0.70	0.72	29092
weighted avg	0.82	0.83	0.82	29092

After

classification report

	precision	recall	f1-score	support
0.0	0.86	0.94	0.90	22672
1.0	0.69	0.46	0.55	6420
accuracy			0.83	29092
macro avg	0.77	0.70	0.73	29092
weighted avg	0.82	0.83	0.82	29092

Confusion Matrix:
[[21358 1314] [3500 2920]]

- 21358 instances were correctly classified as negative.
- 1314instances were incorrectly classified as positive when they were actually negative.

- 3500 instances were incorrectly classified as negative when they were actually positive.
- 2920 instances were correctly classified as positive.

Strength: after improvement

- It is good on predicting actual false values with 86%.
- 94% of the actual negative instances were correctly classified as negative.

Weakness:

- Only 46% of the actual positive instances were correctly classified as positive.

Comparation of Improved models:

Models	TN	FP	FN	TP
DecesionTree	21542	1131	3871	2549
Naïve Bayes	21415	1257	3909	2511
SVM	21753	919	3734	2686
Logistic Regression	21435	1237	3595	2825
MLP	21232	1440	3145	3275
Ensemble (Ada boost)	21358	1314	3500	2920

Accuracy and Time Complexity:

- MLP achieved the highest accuracy of 84.2% but took a considerable amount of time to train and predict.
- SVM had a slightly lower accuracy of 84%, but its training time was significantly longer, especially with higher iterations.
- Logistic regression had a slightly lower accuracy (83.89%,) compared to MLP and SVM but was much faster to train and predict.

Model Performance on Different Prediction Scenarios:

- SVM performed well in predicting true negatives (when it correctly predicts no rain tomorrow).
- MLP excelled in predicting true positives (when it correctly predicts rain tomorrow).
- SVM performed well in minimizing false positives (when it incorrectly predicts rain tomorrow when it doesn't).
- MLP had fewer false negatives (when it incorrectly predicts no rain tomorrow when it does).

Considering both accuracy and time complexity, logistic regression emerges as the preferable choice for this particular dataset. It provides a good balance between accuracy and training efficiency.

SVM may be suitable when correctly predicting negatives is crucial, but it might not be the best option due to its longer training time.

MLP might be preferable when correctly predicting positives is crucial, even though it takes longer to train, but its advantage might not be significant enough to justify its longer training time compared to logistic regression.

Generally, logistic regression seems to offer the best trade-off between accuracy and training time for this specific classification task.