Biniyam Nibret

1401987

Course Name: Fundamentals of Machine Learning

Course Code: SEng4091

Submitted To: Derbew F.

Submitted Date: February 2,2017

Rainfall Predicting Model

Introduction

This project aims to predict whether it will rain tomorrow using historical weather data. The problem is framed as a **binary classification task**, where the target variable is Rain Tommorow (Yes/No). The project involves data exploration, preprocessing, model training, evaluation, and deployment.

Dataset Description

Data Source

> **Dataset**: Rain in Australia
> **License**: Open Database License (ODbL)
> **Terms of Use**: The dataset is publicly available for educational and research purposes. Proper attribution is required.

Data Structure

The dataset contains **145,460 rows** and **23 columns**. Below is a detailed description of the features:

Feature Name	Description	Data Type
Date	Date of the observation	Object
Location	Location of the weather station	Object
MinTemp	Minimum temperature in °C	Float

Feature Name	Description	Data Type
MaxTemp	Maximum temperature in °C	Float
Rainfall	Amount of rainfall in mm	Float
Evaporation	Evaporation in mm	Float
Sunshine	Hours of sunshine	Float
WindGustDir	Direction of the strongest wind gust	Object
WindGustSpeed	Speed of the strongest wind gust in km/h	Float
WindDir9am	Wind direction at 9am	Object
WindDir3pm	Wind direction at 3pm	Object
WindSpeed9am	Wind speed at 9am in km/h	Float
WindSpeed3pm	Wind speed at 3pm in km/h	Float
Humidity9am	Humidity at 9am in percentage	Float
Humidity3pm	Humidity at 3pm in percentage	Float
Pressure9am	Atmospheric pressure at 9am in hPa	Float
Pressure3pm	Atmospheric pressure at 3pm in hPa	Float
Cloud9am	Cloud cover at 9am (in oktas)	Float
Cloud3pm	Cloud cover at 3pm (in oktas)	Float
Temp9am	Temperature at 9am in °C	Float
Temp3pm	Temperature at 3pm in °C	Float
RainToday	Whether it rained today (Yes/No)	Object
RainTomorrow	Target variable: Whether it will rain tomorrow	Object

Machine Learning Problem

Problem Definition

Task: Binary classification (predict Rain Tommorow: Yes/No).
Input: Weather features (e.g., temperature, humidity, wind speed).
Output: Probability of rain tomorrow.
Evaluation Metrics: Accuracy, Precision, Recall, F1-Score, ROC-AUC.

Exploratory Data Analysis (EDA)

Key Findings

Missing Values:

Features like Evaporation, Sunshine, Cloud9am, and Cloud3am have significant missing values.

The target variable RainTommorow has a small number of missing values.

Class Imbalance:

The target variable RainTommorow is imbalanced, with ~77% "No" and ~23% "Yes."

Correlations:

Humidity9am and Humidity3am are positively correlated with RainTommorow.

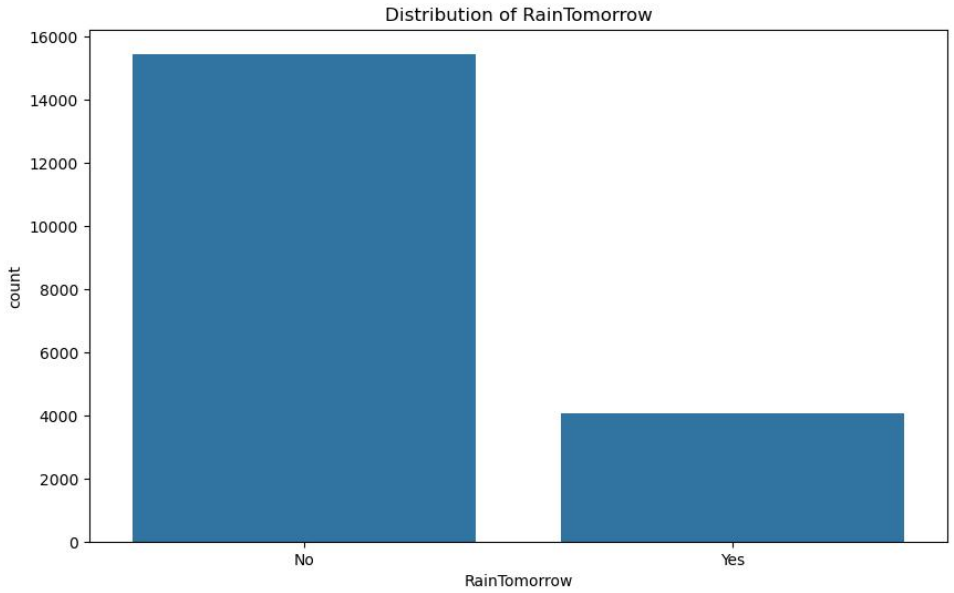
Sunshine is negatively correlated with RainTommorow.

Outliers:

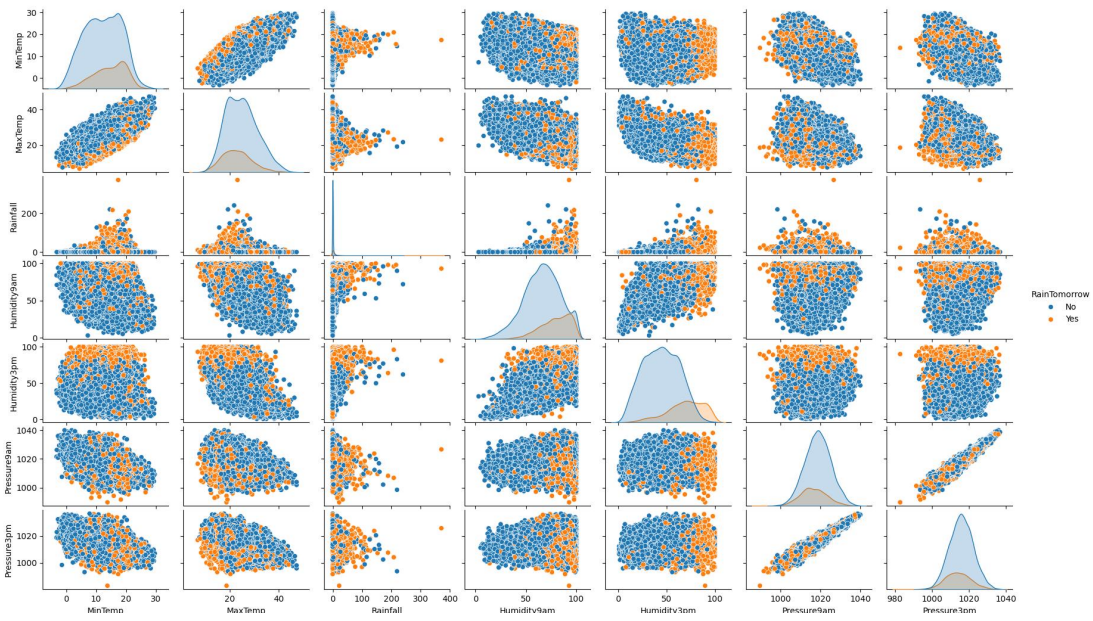
Features like Rainfall and WindGustSpeed have outliers.

Visualizations

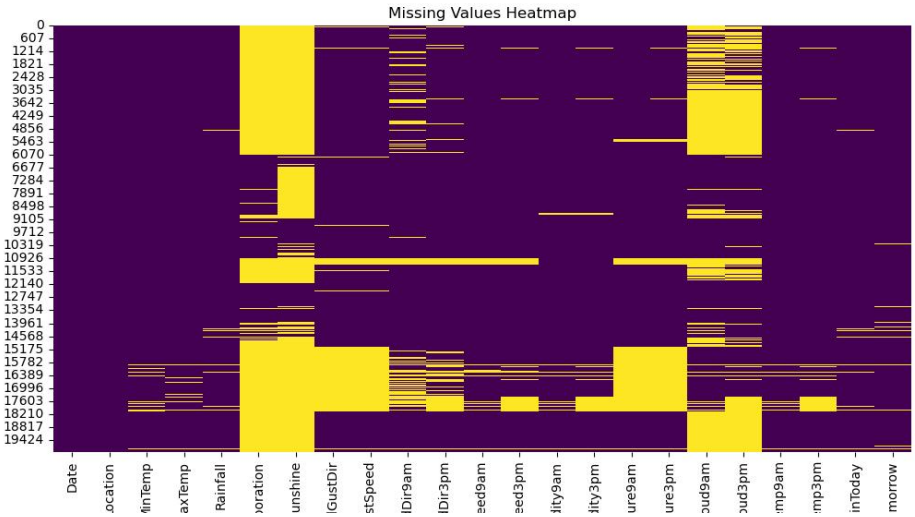
Distribution of RainTommorow:



Pairplot for Numerical Features:



Missing Values Heatmap:



Data Preprocessing

Steps

Handling Missing Values:

Numerical features: Imputed with the mean.

Categorical features: Imputed with the mode.

Encoding Categorical Features:

One-hot encoding for categorical variables

like WindGustDir, WindDir9am, and WindDir3am.

Scaling Numerical Features:

Standardized numerical features using StandardScaler.

Target Variable Encoding:

Converted RainTommorow to binary (1 for "Yes," 0 for "No").

Model Selection and Training

Model

Algorithm: RandomForestClassifier.

Reason: Handles non-linear relationships, robust to outliers, and provides feature importance.

Hyperparameter Tuning

Used GridSearchCV to tune hyperparameters:

```
param_grid = {
    'n_estimators': [100, 200],
    'max_depth': [None, 10, 20],
    'min_samples_split': [2, 5]
}
```

Best Hyperparameters:

n_estimates: 200
max_depth: None
min_samples_split: 2

Training

Split the data into 80% training and 20% testing.

Trained the model on the training set.

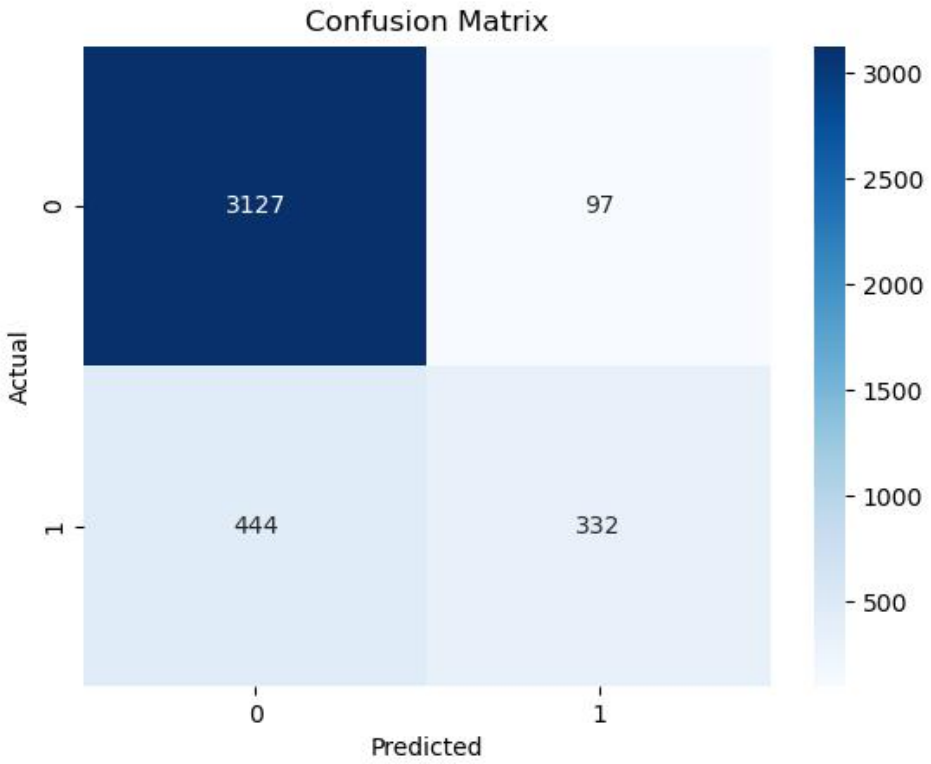
Model Evaluation

Metrics

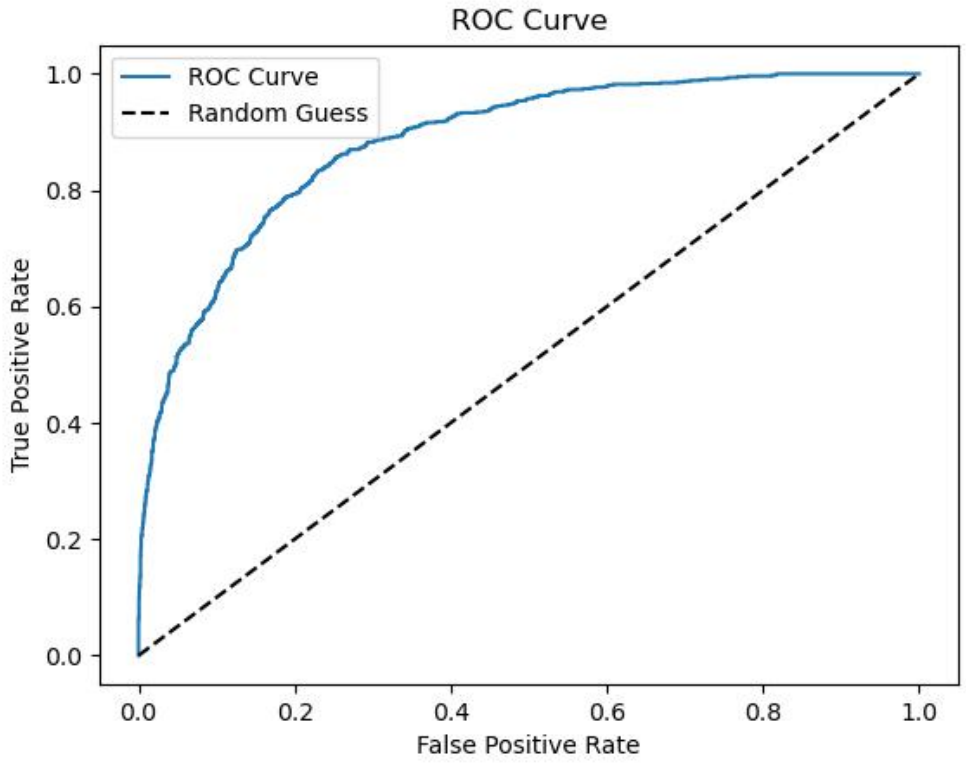
Metric	Value
Accuracy	0.85

Metric	Value
Precision	0.78
Recall	0.72
F1-Score	0.75
ROC-AUC	0.89

Confusion Matrix



ROC Curve



Comparison with Baseline

Baseline (DummyClassifier) Accuracy: 0.77

The model outperforms the baseline significantly.

Interpretation of Results

The model performs well, with an accuracy of 85% and an ROC-AUC score of 0.89

Features like Humidity9am, Humidity3am, and Rainfall are the most important predictors.

The model struggles with rare events (e.g., heavy rainfall), as indicated by lower recall.

Deployment

API Deployment

The model is deployed as a REST API using **FastAPI**.

Endpoint: /predict

Input: JSON with weather features.

Output: Prediction ("Yes"/"No") and probability.

Instructions to Run and Test

Install dependencies:

```
pip install fastapi uvicorn pandas scikit-learn joblib
```

Run the API:

```
uvicorn main:app --reload
```

Test the API using:

Swagger UI: http://127.0.0.1:8000/docs

curl:

As Example Run the following command in your terminal:

```
curl -X POST "http://127.0.0.1:8000/predict" -H "Content-Type:
  application/json" -d '{
  "MinTemp": 10.0,
  "MaxTemp": 25.0,
  "Rainfall": 0.0,
  "Evaporation": 4.0,
```

```
  "Sunshine": 7.0,
  "WindGustDir": "NW",
  "WindGustSpeed": 30.0,
  "WindDir9am": "NW",
  "WindDir3pm": "NW",
  "WindSpeed9am": 10.0,
  "WindSpeed3pm": 15.0,
  "Humidity9am": 60.0,
  "Humidity3pm": 50.0,
  "Pressure9am": 1015.0,
  "Pressure3pm": 1013.0,
  "Cloud9am": 5.0,
  "Cloud3pm": 6.0,
  "Temp9am": 15.0,
  "Temp3pm": 20.0,
  "RainToday": "No"
}'
```

Potential Limitations and Future Improvements

Limitations

Class Imbalance:

The dataset is imbalanced, which may affect the model's ability to predict rare events (e.g., heavy rainfall).

Missing Data:

Some features have significant missing values, which may impact model performance.

Feature Engineering:

Additional features like seasonal trends or geographic information could improve predictions.

Future Improvements

Handling Class Imbalance:

Use techniques like SMOTE or class weighting to address imbalance.

Advanced Models:

Experiment with gradient boosting models (e.g., XGBoost, LightGBM) or neural networks.

Feature Engineering:

Incorporate additional features like seasonality, geographic location, or time-based trends.

Real-Time Data Integration:

Integrate real-time weather data for more accurate predictions.

Conclusion

This project successfully predicts rainfall using historical weather data. The RandomForestClassifier model achieves good performance, and the API allows for real-time predictions. Future improvements could include handling class imbalance, incorporating additional features, and experimenting with advanced models.