Assessment Report

on

Rainfall Prediction Model:

“Build a model to predict whether it will rain tomorrow using classification algorithms and weather data.”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY

DEGREE

SESSION 2024-25

in

ARTIFICIAL INTELLIGENCE

By

Shahid Siddiqui (202401100300223)

Under the supervision of

“Mr. Abhisekh Shukla Sir”

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

(Formerly UPTU)

May, 2025

RAINFALL PREDICTION MODEL USING RANDOM FOREST CLASSIFICATION

1. Introduction

Weather forecasting plays a crucial role in agriculture, disaster prevention, and urban planning. Predicting whether it will rain tomorrow helps in scheduling daily tasks and preventing weather-related disruptions. This project uses machine learning classification algorithms to predict the likelihood of rainfall using historical weather data from the Australian Bureau of Meteorology.

---

2. Methodology

The project consists of three main parts:

1. **Data Preprocessing:**
    - Dropped features with excessive missing values (e.g., Evaporation, Sunshine).
    - Handled missing values in numeric columns using mean imputation.
    - Encoded categorical columns using LabelEncoder.
2. **Model Building:**
    - Split the dataset into training (80%) and testing (20%) sets.
    - Trained a RandomForestClassifier model.
3. **Evaluation:**
    - Calculated accuracy, precision, recall, and F1-score.
    - Plotted confusion matrix and feature importance graph.

3. Code

```
from google.colab import files

uploaded = files.upload()

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import LabelEncoder

from sklearn.impute import SimpleImputer

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

import joblib

# Load dataset

df = pd.read_csv('weatherAUS.csv')

# Drop columns with many missing values

df = df.drop(columns=['Evaporation', 'Sunshine', 'Cloud9am', 'Cloud3pm'])

# Drop rows with missing target

df = df.dropna(subset=['RainTomorrow'])


# Fill numeric missing values with mean

numeric_cols = df.select_dtypes(include=['float64']).columns

imputer = SimpleImputer(strategy='mean')

df[numeric_cols] = imputer.fit_transform(df[numeric_cols])

# Encode all object-type columns using LabelEncoder

categorical_cols = df.select_dtypes(include=['object']).columns

label_encoders = {}
```

```
for col in categorical_cols:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col].astype(str))
    label_encoders[col] = le

# Separate features and target
X = df.drop(columns=['RainTomorrow'])
y = df['RainTomorrow']
# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Train the model
rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)
# Make predictions
y_pred = rf_model.predict(X_test)
# Evaluate
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
# Generate the confusion matrix
cm = confusion_matrix(y_test, y_pred)
# Plot it using seaborn heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['No', 'Yes'],
yticklabels=['No', 'Yes'])
plt.xlabel('Predicted')
```

```
plt.ylabel('Actual')
plt.title('Confusion Matrix for RainTomorrow Prediction')
plt.tight_layout()
plt.show()
# Feature Importance Graph
importances = rf.feature_importances_
features = X.columns
indices = importances.argsort()[::-1]

plt.figure(figsize=(10, 6))
sns.barplot(x=importances[indices], y=features[indices])
plt.title('Feature Importances from Random Forest')
plt.xlabel('Importance')
plt.ylabel('Features')
plt.tight_layout()
plt.show()
```

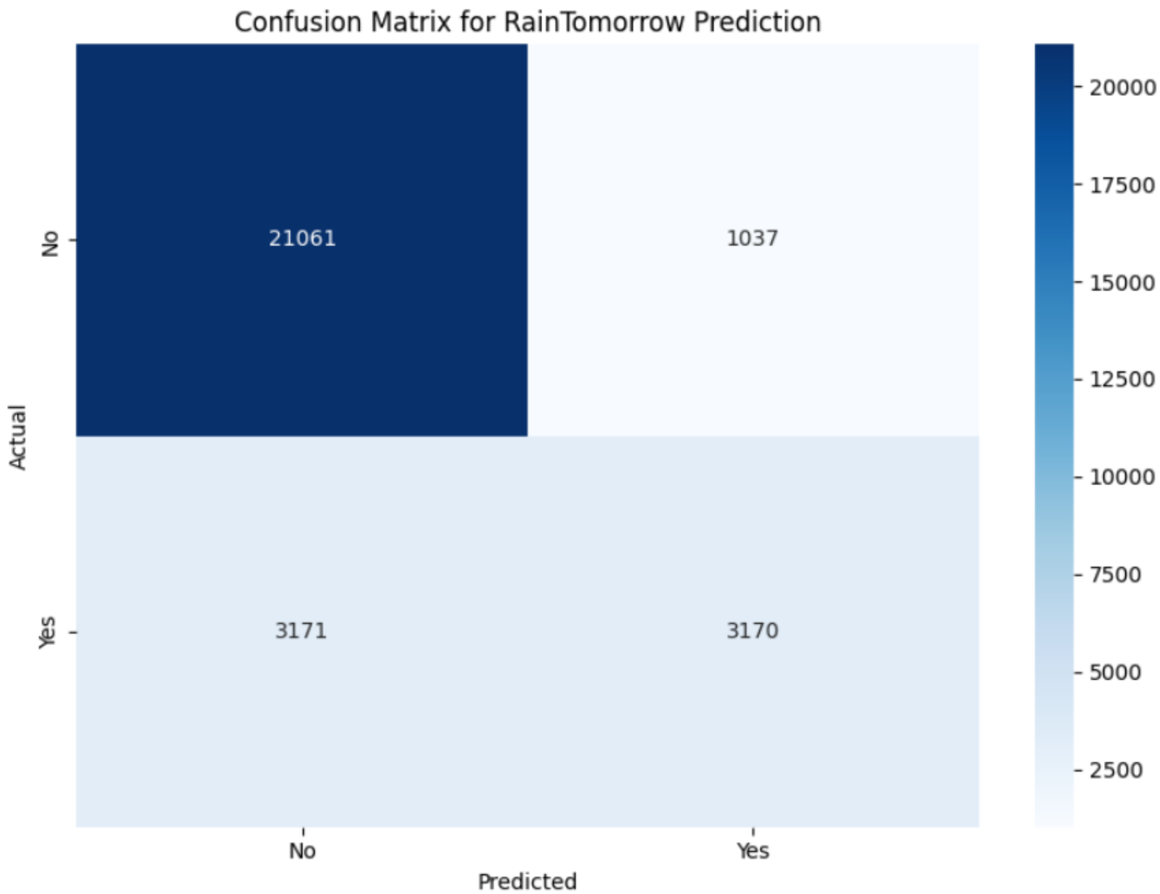
4. Output

4.1 Classification Results:

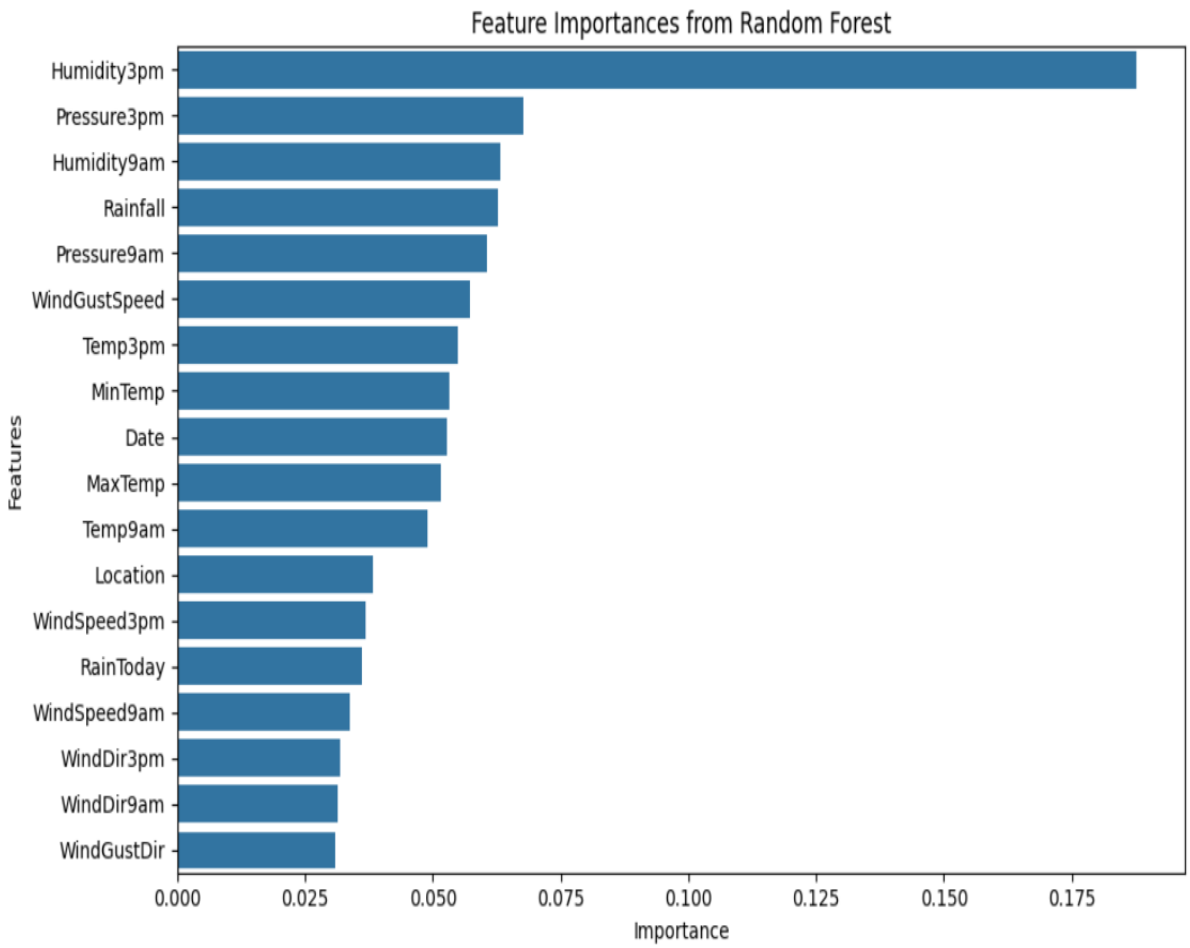
- **Accuracy: ~85%**

```
Accuracy: 0.8520341784169626
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.95      0.91     22098
           1       0.75      0.50      0.60      6341

    accuracy                           0.85     28439
   macro avg       0.81      0.73      0.76     28439
weighted avg       0.84      0.85      0.84     28439
```

- **Confusion Matrix: (visual shown in graph)**



- **Feature Importance Graph: (displays most impactful weather features)**



5. References

- Dataset: Kaggle - Australian Weather Data
- Seaborn, Matplotlib, Scikit-learn documentation
- Project developed by Uday Gangwar, KIET Group of Institutions