A PROJECT ON

“RAINFALL PREDICTION WITH DATA ANALYTICS”

SUBMITTED IN

PARTIAL FULFILLMENT OF THE
REQUIREMENTFOR THE COURSE OF

DIPLOMA IN BIG DATA ANALYSIS

SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY

‘Plot no R/2’, Market
yard road,Behind
hotel Fulera,Gultekdi
Pune – 411037.

MH-INDIA

SUBMITTED BY:

Amandeep Singh Rathor (75493)
-
Anita Dogra (75408)

UNDER THE GUIDENCE OF:

Mr. Amit Kulkarni

Sunbeam Institute of Information Technology, PUNE.

SUNBEAM

CERTIFICATE

This is to certify that the project work under the title ‘Rainfall Prediction with Data Analytics”is done by Anita Dogra and Amandeep Singh Rathr in partial fulfillment of the requirement for award of Diploma in Big Data Analysis Course.

Mr. Amit Kulkarni
Project Guide

Mrs. Pradnya Dindorkar
Course Co-ordinator

Date:

ACKNOWLEDGEMENT

A project usually falls short of its expectation unless aided and guided by the right persons at the right time. We avail this opportunity to express our deep sense of gratitude towards Mr. Nitin Kudale (Center Coordinator, Sunbeam, Pune) and Mrs. Pradnya Dindorikar(Course Coordinator, Sunbeam ,Pune) and Project Guide.

We are deeply indebted and grateful to them for their guidance, encouragementand deep concern for our project. Without their critical evaluation and suggestions at every stage of the project, this project could never have reached its present form.

Last but not the least we thank the entire faculty and the staff members of SunbeamInstitute of Information Technology, Pune for their support.

Anita Dogra
DBDA March 2023
Batch, Sunbeam
Pune

Amandeep Singh Rathor
DBDA March
2023 Batch,
Sunbeam Pune

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Introduction And Objectives

Rainfall Prediction is one of the difficult and uncertain tasks that have a significant impact on human society. Timely and accurate forecasting can proactively help reduce human and financial loss. This study presents a set of experiments that involve the use of common machine learning techniques to create models that can predict whether it will rain tomorrow or not based on the weather data for that day in major cities in Australia. The "Rainfall Prediction with Data Analytics" project focuses on improving rain forecasts for different parts of Australia. This matters for farming, water supply, and safety. We use advanced methods to predict rain and help people prepare.

Objectives

The "Rainfall Prediction with Data Analytics" project aims to:

- Improve rain forecasts: Make better guesses about when and where rain will fall using past data.
- Understand local differences: Figure out why some places get more rain than others to make predictions more accurate.
- Study timing and seasons: Learn when rain usually happens to help with farming and water planning.
- Consider climate change: See how rain might change in the future due to climate change effects.
- Aid in emergencies: Provide predictions to help people get ready for heavy rain and other weather challenges.
- Simplify information: Create an easy tool to show when rain is expected, helping people make informed decisions.

In short, the "Rainfall Prediction with Data Analytics" project aims to enhance rain forecasts, making it easier for people to prepare and adapt, especially as the weather changes.

1.2 Why this problem needs To be Solved?

This problem needs to be solved because accurate rainfall predictions in Australia are crucial for:

- Farming: Planning crops and reducing losses.
- Water Management: Ensuring a steady water supply.
- Disaster Readiness: Warning and preparing for floods and landslides.
- Infrastructure: Building resilient structures.
- Business Impact: Adapting to weather for businesses.
- Climate Adaptation: Dealing with changing weather patterns.
- Research: Advancing meteorology and data science.
- Community Readiness: Being prepared for rainy days.

Solving this benefits agriculture, safety, the economy, the environment, and society.

1.3 Dataset Information

The dataset consists of 220,094 rows and 24 columns.

WeatherAUS.csv

Date- The date of Observation

Location - The common name of the location of the weather station

MinTemp - The minimum temperature in degrees celsius

MaxTemp - The maximum temperature in degrees celsius

Rainfall - The amount of rainfall recorded for the day in mm

Evaporation - The so-called Class A pan evaporation (mm) in the 24 hours to 9am

Sunshine - The number of hours of bright sunshine in the day.

WindGustDir - The direction of the strongest wind gust in the 24 hours to midnight

WindGustSpeed - The speed (km/h) of the strongest wind gust in the 24 hours to midnight

WindDir9am - Direction of the wind at 9am

WindDir3pm - Direction of the wind at 3pm

WindSpeed9am - Wind speed (km/hr) averaged over 10 minutes prior to 9am

WindSpeed3pm - Wind speed (km/hr) averaged over 10 minutes prior to 3pm

Humidity9am - Humidity (percent) at 9am

Humidity3pm - Humidity (percent) at 3pm

Pressure9am - Atmospheric pressure (hpa) reduced to mean sea level at 9am

Pressure3pm - Atmospheric pressure (hpa) reduced to mean sea level at 3pm

Cloud9am - Fraction of sky obscured by cloud at 9am. This is measured in "oktas", which are a unit of eigths.

Cloud3pm - Fraction of sky obscured by cloud (in "oktas": eighths) at 3pm. See Cload9am for a description of the values

Temp9am - Temperature (degrees C) at 9am

Temp3pm - Temperature (degrees C) at 3pm

RainToday - Boolean: 1 if precipitation (mm) in the 24 hours to 9am exceeds 1mm, otherwise 0

RainTomorrow - The amount of next day rain in mm. Used to create response variable RainTomorrow. A kind of measure of the "risk".

2. Problem Definition and Algorithm

2.1 Problem Definition

The challenge entails constructing a rainfall prediction model through diverse machine learning algorithms, utilizing a dataset containing historical weather information. The aim is to forecast whether rain will transpire based on these dataset attributes. To discern the optimal algorithm, a range of classifiers—Logistic Regression, Decision Tree, Random Forest, CatBoost and XGBoost—are employed. Through a comparison of accuracy and precision metrics, we intend to pinpoint the algorithm that yields the most reliable predictions for rainfall events, ultimately offering enhanced forecasting capabilities.

In this pursuit, the chosen algorithm will empower us to develop an accurate and dependable rainfall prediction model, facilitating improved preparedness and decision-making in response to changing weather conditions.

2.2 Algorithm Definition

Logistic Regression: Logistic Regression is a linear classification algorithm used for binary classification tasks. It calculates the probability of a data point belonging to a particular class and then assigns it to the class with the highest probability. It's commonly used when the dependent variable is binary (two classes), and it models the relationship between the features and the probability of the target class.

Decision Tree Classifier: The Decision Tree Classifier is a tree-like structure used for both classification and regression tasks. It breaks down the dataset into smaller subsets by making decisions based on features. Each internal node of the tree represents a decision based on a feature, leading to different branches representing possible outcomes. It's a versatile algorithm that's easy to interpret but can suffer from overfitting if not managed well.

Random Forest Classifier: The Random Forest Classifier is an ensemble learning method that builds multiple decision trees and combines their predictions to achieve better accuracy and generalization. Each tree is trained on a random subset of the data, and the final prediction is determined by aggregating the predictions of individual trees. Random Forest reduces overfitting and enhances predictive performance.

CatBoost Classifier: CatBoost (Categorical Boosting) Classifier is a gradient boosting algorithm that excels at handling categorical features without the need for extensive preprocessing. It uses a combination of ordered boosting and oblivious trees, which makes it efficient and effective in capturing complex relationships in the data. CatBoost automatically deals with categorical variables, reducing the need for manual encoding.

XGBoost Classifier: XGBoost (Extreme Gradient Boosting) Classifier is another gradient boosting algorithm that's highly efficient and widely used for classification tasks. It builds a series of weak learners (decision trees) in a sequential manner, with each tree attempting to correct the errors made by the previous ones. XGBoost is known for its performance and versatility, making it a popular choice in competitions and real-world applications.

3. Experimental Evaluation

3.1 Methodology/Model

Here's a methodology for creating the rainfall prediction model using the mentioned algorithms:

To begin, we import the necessary libraries and load the dataset using the Pandas library.

1. Loading the Data:

```
df = pd.read_csv("/home/lenovo/Downloads/weatherAUS.csv")
```

Before diving into the analysis first We use the df.info() method to check the dataset's length and identify any missing values. The output will show information about the number of non-null entries in each column and the dataset's overall structure.

2. Data Exploration:

When working with this dataset first understanding the nature of its features is essential for effective analysis and modeling. This section will categorize the features in our rainfall prediction dataset and count the number of features in each category.

3. Data Cleaning and Data Preprocessing:

- Categorical target "RainTomorrow" was transformed into binary values (0 and 1) for consistency.

```
df['RainTomorrow'] = df['RainTomorrow'].replace({'Yes':1,"No":0}) •
```

- Class imbalance shown in Figure 1.1 was identified in the target variable "RainTomorrow," motivating the adoption of oversampling techniques to balance the dataset.

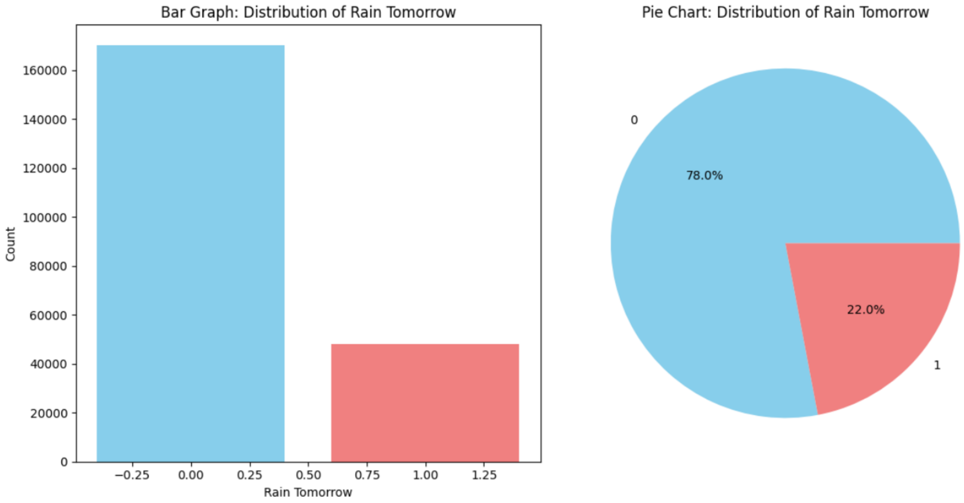


Figure 1.1 Class Imbalance of Target Variable

```
X = df.drop('RainTomorrow', axis=1)
Y = df['RainTomorrow']
over_sampler = RandomOverSampler(random_state=1234)
X_over, Y_over = over_sampler.fit_resample(X, Y)
```

- Missing data shown in Figure 1.2 & Figure 1.3 in features such as "Evaporation," "Sunshine," "Cloud9am," and "Cloud3pm" were handled using Mode.

```
def fill_categorical_variables_mode(df,variables):
        mode = df[variables].mode().iloc[0]
        df[variables] = df[variables].fillna(mode)
        return df
fill_categorical_variables_mode(df,categorical_variables_NA)
```

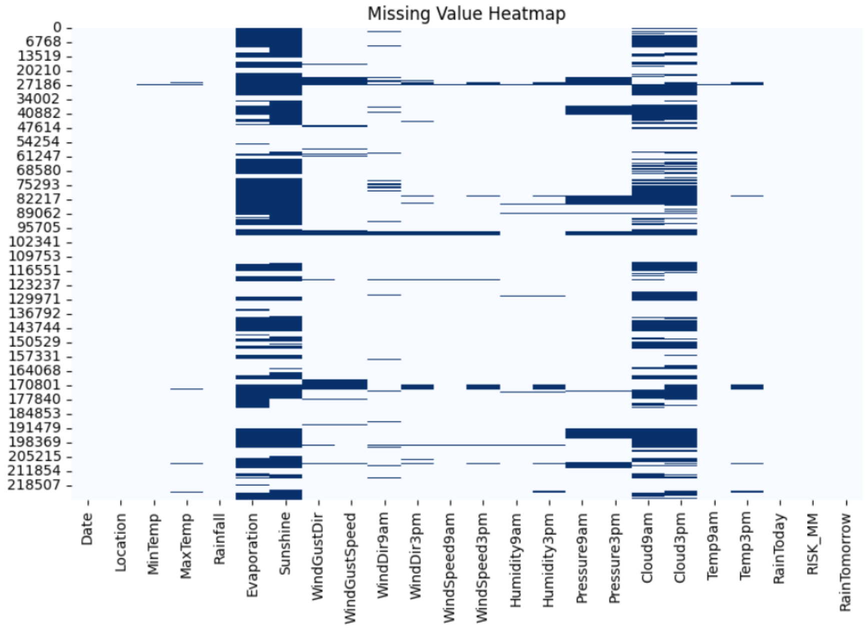


Figure 1.2 Heatmap of missing values

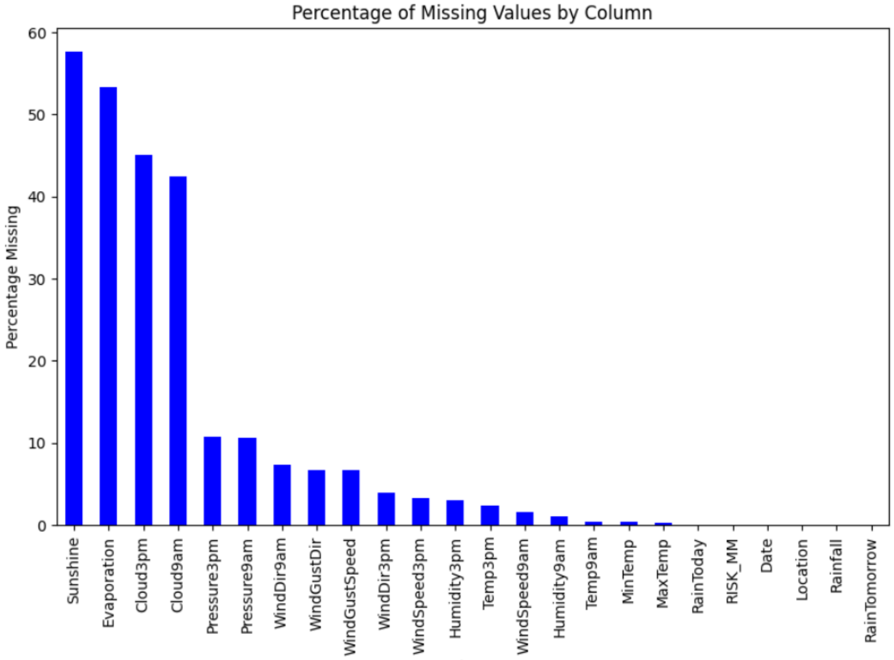


Figure 1.3 Percentage of Missing values

- Categorical features underwent label encoding after imputation, converting them into numerical representations.

```
encoder = LabelEncoder()
for variable in categorical_variables:
        df[variable] = encoder.fit_transform(df[variable])
```

- The Multiple Imputation by Chained Equations (MICE) algorithm was employed to impute remaining missing data.

```
from sklearn.experimental import enable_iterative_imputer
from sklearn.impute import IterativeImputer
final_df = df.copy(deep=True)
mice_imputer = IterativeImputer()
final_df.iloc[:, :] = mice_imputer.fit_transform(df)
```

- A correlation analysis revealed relationships between features, guiding decisions on feature retention.

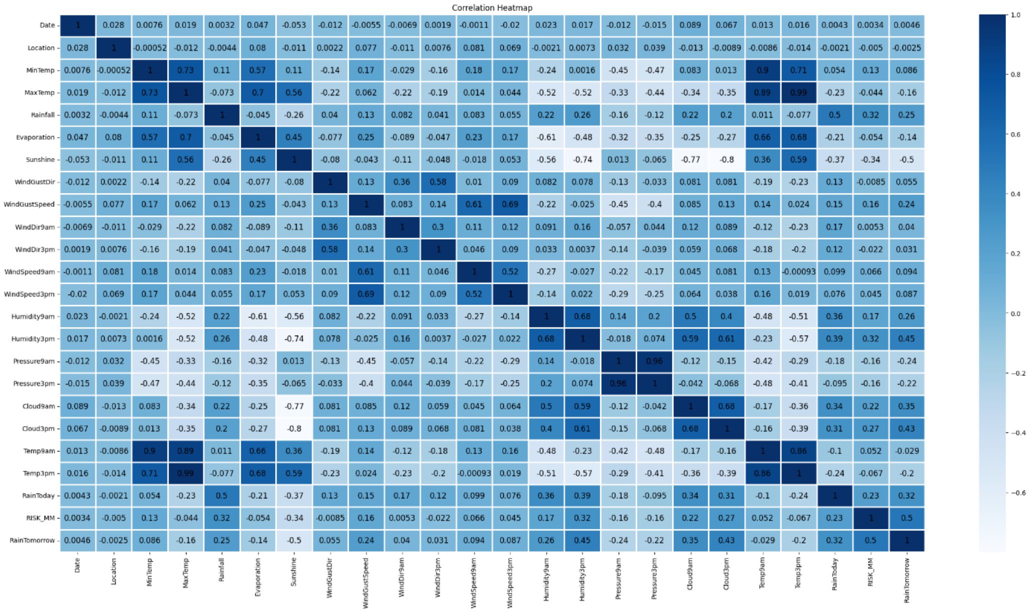


Figure 1.4 Heatmap of a correlation matrix

The following feature pairs have a strong correlation with each other:

- MaxTemp and MinTemp
- Pressure9h and pressure3h
- Temp9am and Temp3pm
- Evaporation and MaxTemp
- MaxTemp and Temp3pm But in no case is the correlation value equal to a perfect “1”. We are therefore not removing any functionality. However, we can delve deeper into the

pairwise correlation as shown in Figure 1.5 between these highly correlated characteristics by examining the following pair diagram. Each of the paired plots shows very clearly distinct clusters of RainTomorrow's "yes" and "no" clusters. There is very minimal overlap between them.

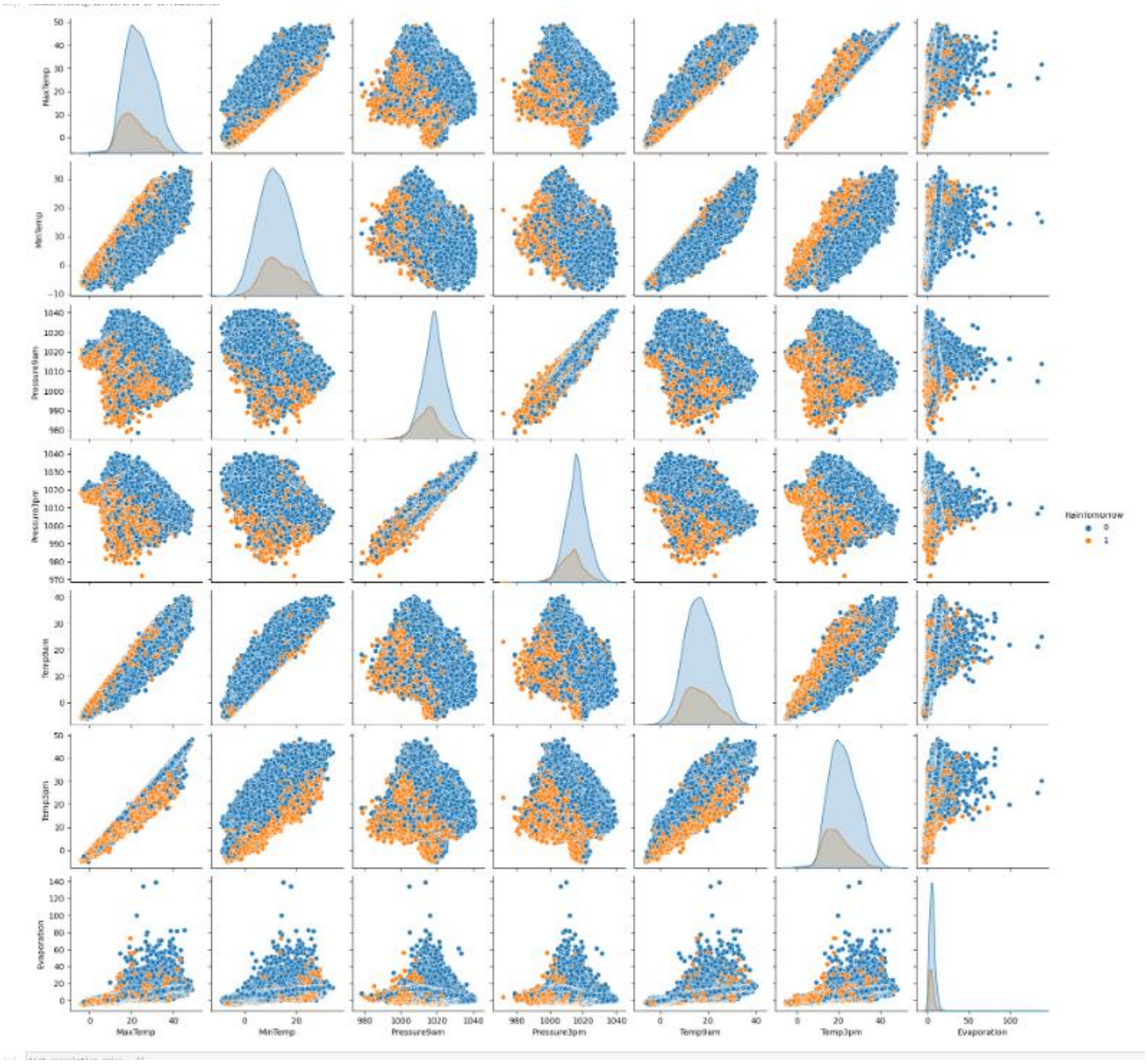


Figure 1.5 Pair Plot of highly corealated Columns

4. Feature Selection:

- Two distinct approaches were utilized for feature selection: The filter method and the wrapper method.

- The filter method employed the chi-square test to determine feature importance, resulting in the identification of significant features including "Sunshine," "Humidity9am," "Humidity3pm," and more.

```
from sklearn.feature_selection import SelectKBest, chi2
X = modified_data.drop('RainTomorrow',axis =1)
Y = modified_data['RainTomorrow']
selector = SelectKBest(chi2, k=5)
selector.fit(X, Y)
X_new = selector.transform(X)
print(X.columns[selector.get_support(indices=True)])
```

- The wrapper method utilized a Random Forest classifier for feature selection, highlighting key attributes like "Sunshine," "Cloud3pm," and "RISK_MM."

```
from sklearn.feature_selection import SelectFromModel
from sklearn.ensemble import RandomForestClassifier as rf

X = final_df.drop('RainTomorrow', axis=1)
y = final_df['RainTomorrow']
selector = SelectFromModel(rf(n_estimators=100, random_state=0))
selector.fit(X, y)
support = selector.get_support()
features = X.loc[:,support].columns.tolist()
print(features)
print(rf(n_estimators=100, random_state=0).fit(X,y).feature_importances_)
```

5. Training Rainfall Prediction Model with Different Models:

5.1 Dataset Split:

The dataset is divided into two parts: a training set and a test set. The training set consists of 80% of the data, while the test set contains the remaining 20%. This division ensures that the model is trained on a substantial portion of the data and tested on unseen data to evaluate its generalization performance.

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.20,
random_state=12345)
```

5.2 Model Training

Various machine learning models are trained using the standardized training data (X_train and y_train). These models include algorithms like Logistic Regression, Random Forest ,Decision Tree, CatBoost, XGBoost. Each model learns patterns and relationships in the training data to make predictions about whether it will rain tomorrow.

5.3 Model Evaluation

After training, the models are evaluated using the standardized test data (X_test and y_test). Evaluation metrics such as accuracy, ROC AUC (Receiver Operating Characteristic Area Under Curve), Cohen's Kappa, and classification reports are computed to assess how well the models perform on unseen data.

6 Results and discussion:

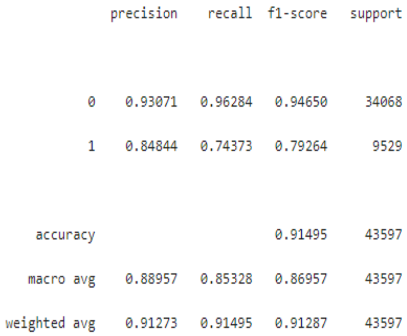
Logistic Regression, Decision Tree, Random Forest, Cat Boost and XG Boost algorithm were used to predict ‘RainfallTomorrow’. Among the given algorithms XG Boost Machine algorithm was the best performing one as it provided the highest Accuracy 0.91, Precision 0.93, Recall 0.96 and F1-Score 0.94 as shown in figure.

```
from sklearn.ensemble import RandomForestClassifier

params_rf = {'max_depth': 16,
  'min_samples_leaf': 1,
  'min_samples_split': 2,
  'n_estimators': 100,
  'random_state': 12345}

model_rf = RandomForestClassifier(**params_rf)
model_rf, accuracy_rf, roc_auc_rf, coh_kap_rf, tt_rf =
```

run_model(model_rf, X_train, y_train, X_test, y_test)



```
              precision    recall  f1-score   support

           0    0.93071   0.96284   0.94650     34068
           1    0.84844   0.74373   0.79264      9529

    accuracy                        0.91495     43597
   macro avg    0.88957   0.85328   0.86957     43597
weighted avg    0.91273   0.91495   0.91287     43597
```

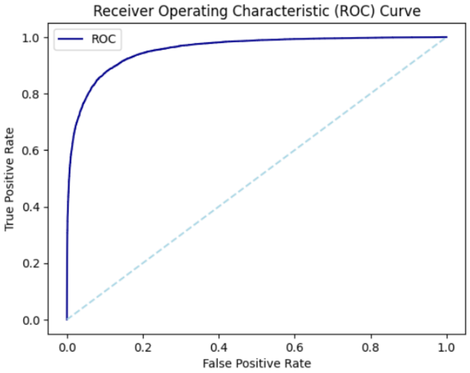


Figure 1.6 ROC-AUC Curve

7 GUI:

GUI is made using Flask framework. **Flask** is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

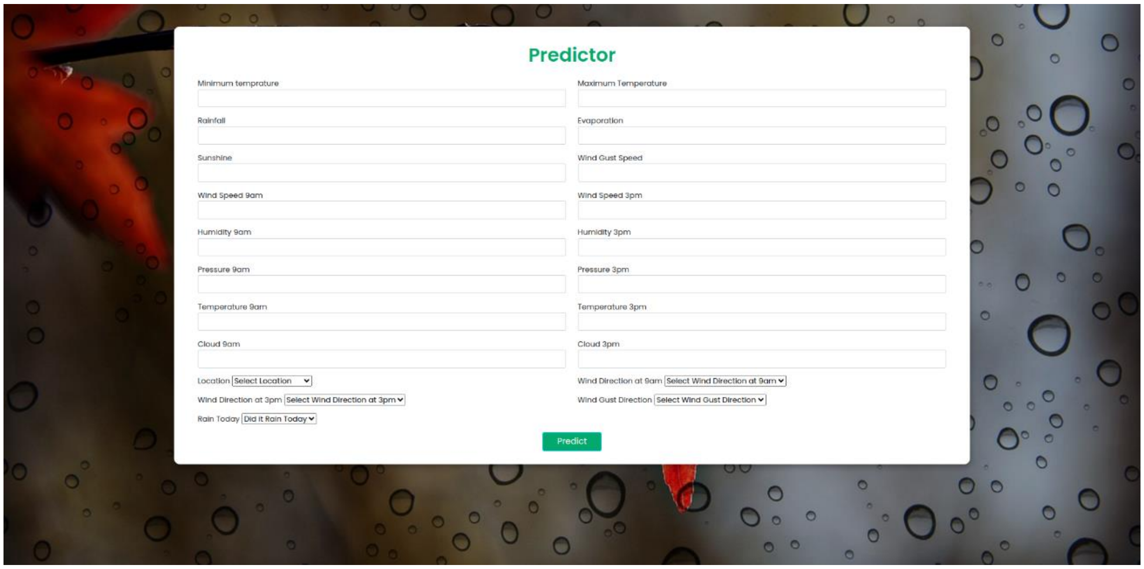


Figure 1.6 GUI representation

8 Future Work And Conclusion

8.1 Future Work: Future Work:

1. Enhanced Data Collection: Collect more diverse and accurate data related to weather conditions, atmospheric pressure, wind patterns, and other relevant factors. Incorporate real-time data streams from satellites, weather stations, and IoT devices to improve the accuracy of predictions.

2. Advanced Machine Learning Models: Investigate the effectiveness of more sophisticated machine learning models such as deep learning architectures (convolutional neural networks, recurrent neural networks) or ensemble methods (Random Forest, Gradient Boosting) to capture intricate relationships within the data.

3. Hyperparameter Tuning: Fine-tune the hyperparameters of the chosen machine learning algorithm to optimize model performance. Utilize techniques like grid search, random search, or Bayesian optimization to find the best combination of hyperparameters.

4. Model Interpretability: Implement techniques for model interpretability, such as SHAP (SHapley Additive exPlanations) values or LIME (Local Interpretable Model-Agnostic Explanations), to provide insights into how the model arrives at its predictions. This will help build trust and understanding among stakeholders.

5. Online Learning: Develop a mechanism for the model to adapt and learn continuously as new data becomes available. Online learning can improve the model's performance over time and ensure it remains up-to-date with the latest weather information.

8.2 Conclusion:

In conclusion, the rain prediction project has shown promising results in accurately forecasting rainfall based on historical weather data. The combination of feature engineering and machine learning techniques has provided valuable insights into the complex relationships that influence rainfall patterns. However, there is still room for improvement and further research to enhance the accuracy and reliability of the predictions.

As we move forward, advancements in data collection, machine learning algorithms, and interpretability techniques will play a crucial role in refining the rain prediction model. This project has the potential to contribute significantly to various sectors, including agriculture, disaster management, and urban planning, by providing timely and accurate rainfall forecasts.

By addressing the future work suggestions mentioned above, we can continue to refine the model's performance and make it more adaptable to changing weather conditions. As technology evolves and more data becomes available, the rain prediction project can serve as a valuable tool for making informed decisions and mitigating the impacts of unpredictable weather events.