BOSTON INSTITUTE OF ANALYTICS

RAIN FORECASTING USING MACHINE LEARNING

Agenda

- **Problem statement**
- **Objective**
- **Exploratory data analysis**
- **Model building**
- **Comparing accuracy of models**
- **Model deployment**
- **Conclusion**

Problem Statement

Accurate rainfall forecasting is critical for effective water management, including planning water allocation, storage, and distribution strategies. In Mumbai, India, unpredictable rainfall patterns pose significant challenges for ensuring a consistent water supply. This project aims to develop a machine learning model to forecast rainfall for the upcoming months in Mumbai. By accurately predicting rainfall, we can optimize water allocation and reservoir management, minimizing operational costs and ensuring a reliable water supply throughout the year.

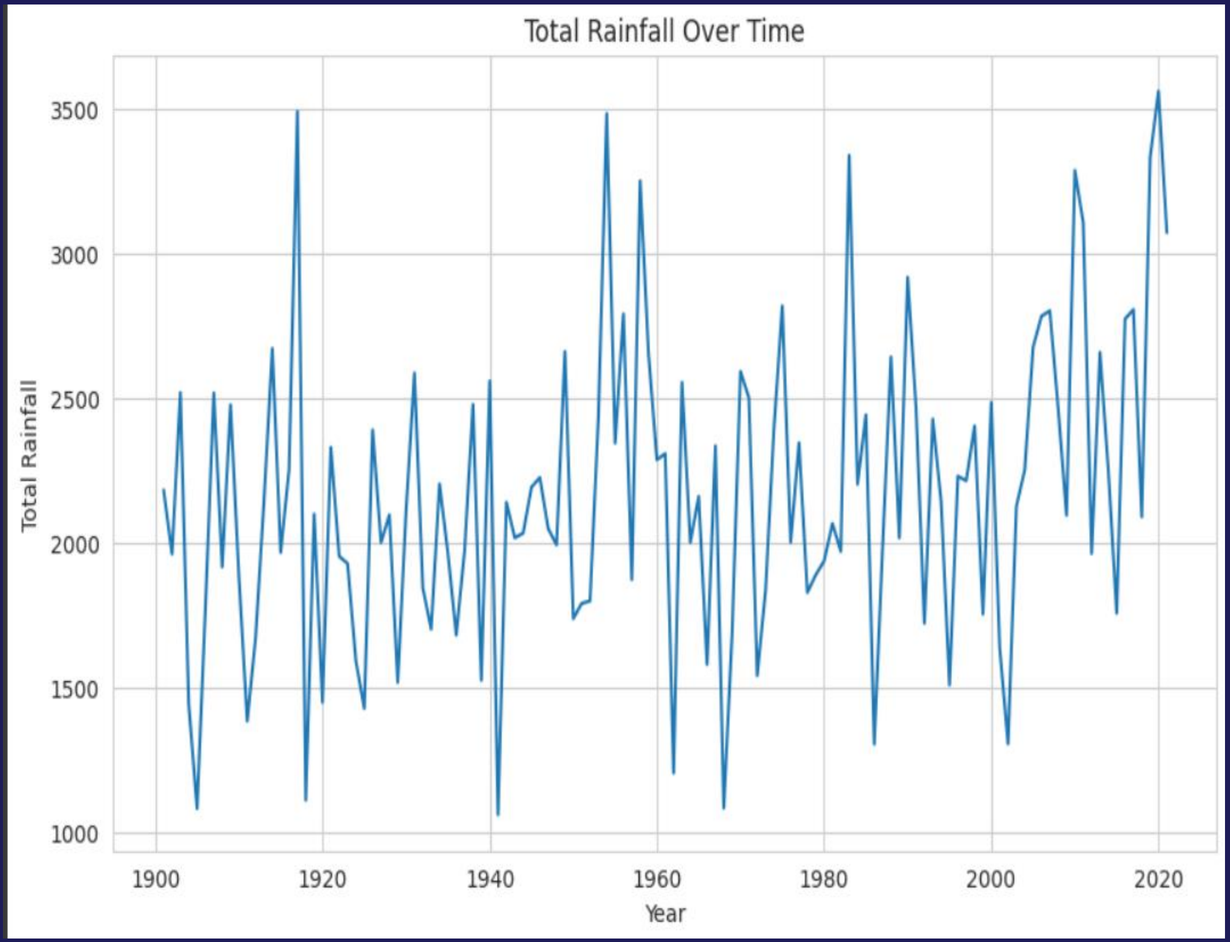
Objective

The primary objective of this project is to develop well defined predictive Model to forecast rainfall patterns in upcoming months. This model will leverage historical data and external factors to accurately predict anticipating rainfall patterns will allow them to optimize water allocation and storage strategies, ultimately reducing costs and ensuring a reliable water supply for Mumbai . By planning infrastructure maintenance and upgrade projects during dry periods, minimizing disruptions and maximizing efficiency during the rainy season.

Exploratory Data Analysis
Graph Analysis

Highs and Lows

- The highest rainfall occurs around the mid-1950s and 1960s, with values exceeding 3500 units.
- The lowest rainfall periods are around the early 1900s and 1920s, with values close to 1000 units.

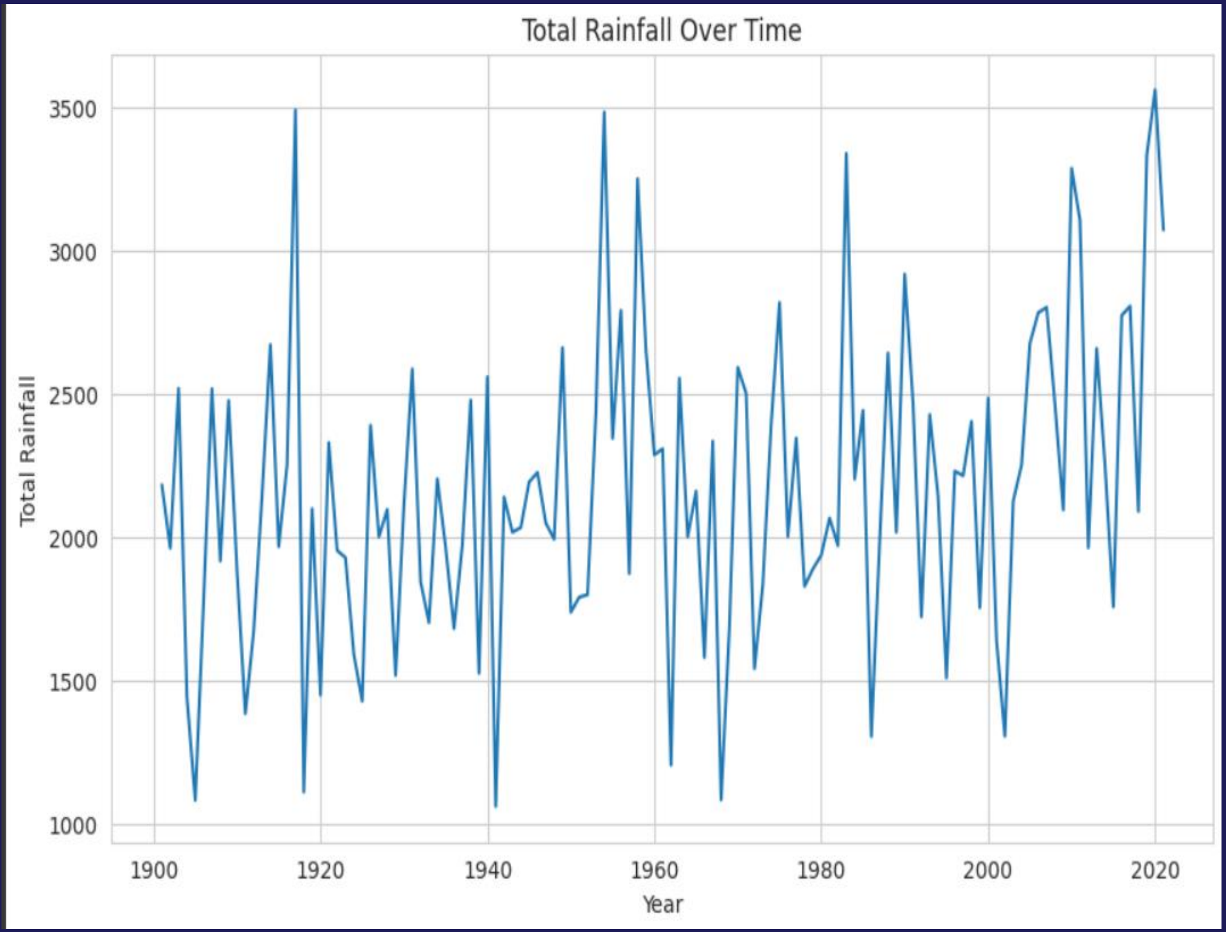


Graph Analysis

- Early 1900 and before 1920 Rainfall dips where rainfall is around or below 1500 units.

•**1950-1960**: The decade shows significant variability, with both high peaks and low troughs.

•**2000-2020**: Re increased variability with significant peaks and troughs.



BIA BOSTON INSTITUTE OF ANALYTICS

Model building- Model 1

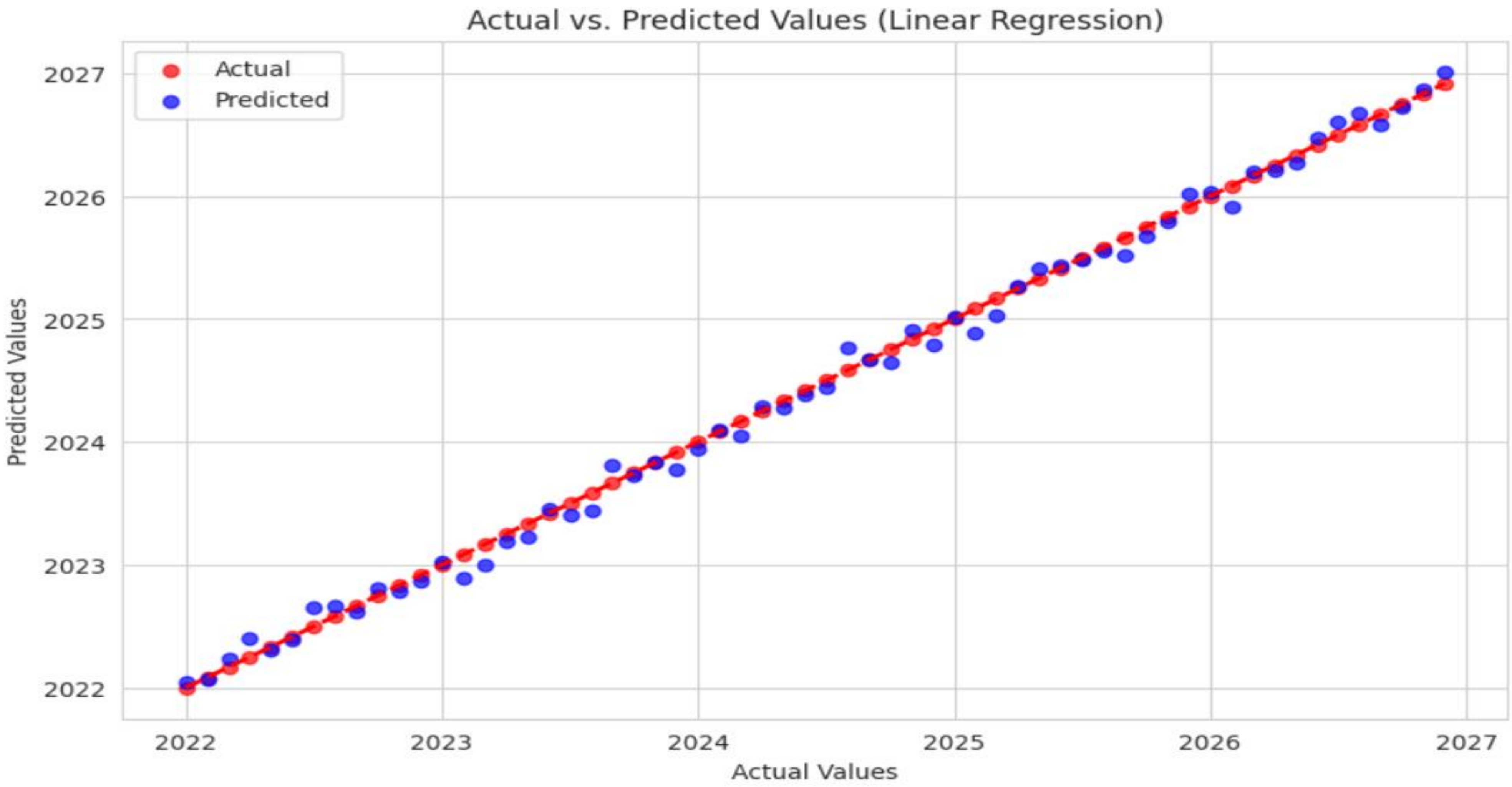
LINEAR REGRESSION

```
# Initialize the models
 lr = Linear Regression()

# Train and evaluate Linear Regression
lr.fit(X_train_scaled, y_train)
y_pred_lr = lr.predict(X_test_scaled)
print('Linear Regression')
print('MSE:', mean_squared_error(y_test,
y_pred_lr))
print('R2 Score:', r2_score(y_test, y_pred_lr))
```

Visualizing the actual vs. predicted values in Linear Regression

```
def plot_actual_vs_predicted(actual, predicted, model_name):
    plt.figure(figsize=(10, 6))
    plt.scatter(actual, actual, color='red', alpha=0.7, label='Actual')
    plt.scatter(actual, predicted, color='blue', alpha=0.7, label='Predicted')
    plt.plot([min(actual), max(actual)], [min(actual), max(actual)], 'r--', lw=2)
    plt.xlabel('Actual Values')
    plt.ylabel('Predicted Values')
    plt.title(f'Actual vs. Predicted Values ({model_name})')
    plt.legend()
    plt.grid(True)
    plt.show()

# Plot for Linear Regression
plot_actual_vs_predicted(actual_values, predicted_values_lr, 'Linear Regression')
```

Visualizing the actual vs. predicted values in Linear Regression



Model building- Model 2

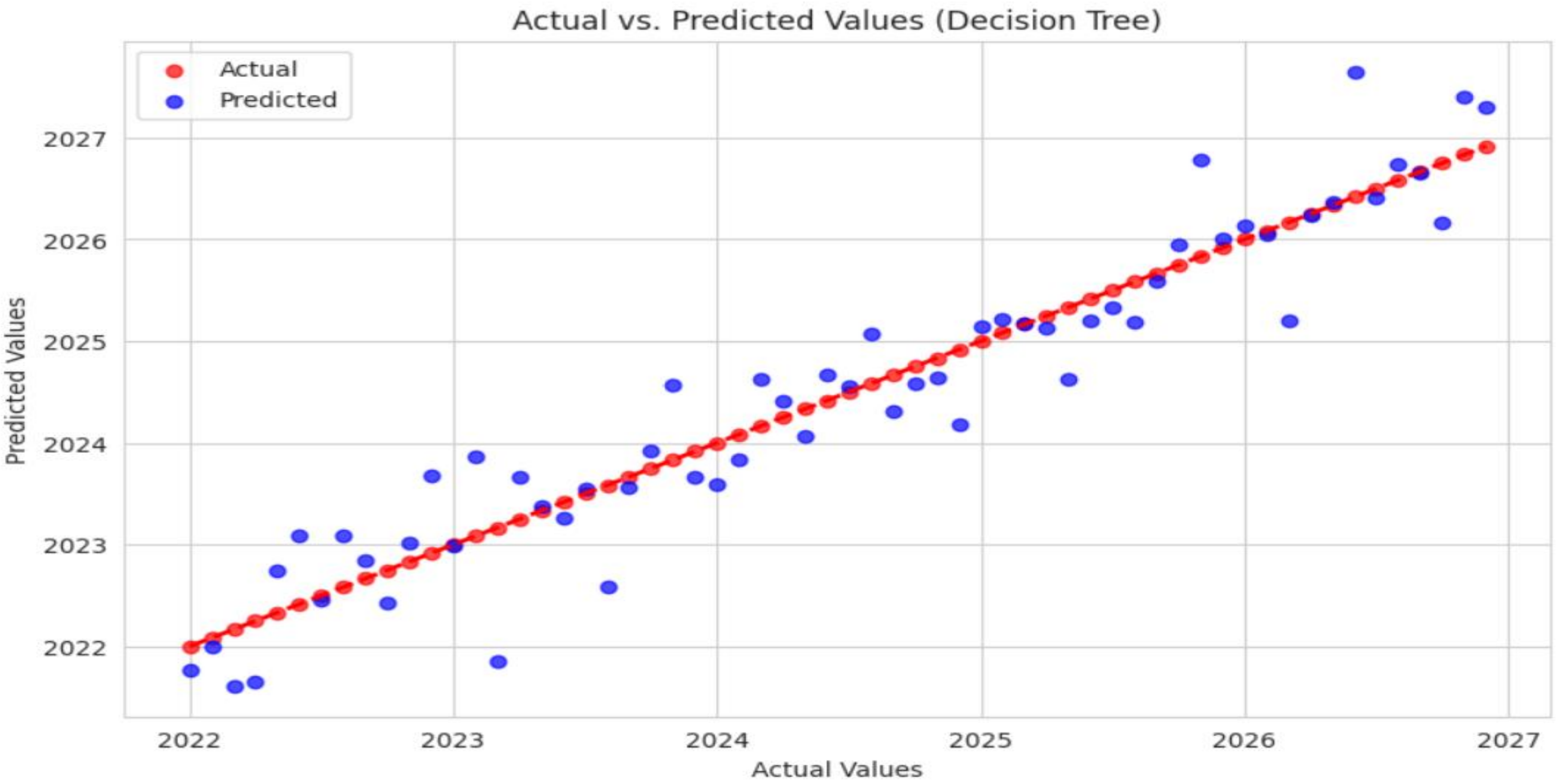
DECISION TREE REGRESSOR

```
# Initialize the models
dt = DecisionTreeRegressor(random_state=42)


# Train and evaluate Decision Tree Regressor
dt.fit(X_train, y_train)
y_pred_dt = dt.predict(X_test)
print('Decision Tree')
print('MSE:', mean_squared_error(y_test, y_pred_dt))
print('R2 Score:', r2_score(y_test, y_pred_dt))
```

Visualizing the actual vs. predicted values in Decision Tree Regression

```
def plot_actual_vs_predicted(actual, predicted, model_name):
    plt.figure(figsize=(10, 6))
    plt.scatter(actual, actual, color='red', alpha=0.7, label='Actual')
    plt.scatter(actual, predicted, color='blue', alpha=0.7, label='Predicted')
    plt.plot([min(actual), max(actual)], [min(actual), max(actual)], 'r--', lw=2)
    plt.xlabel('Actual Values')
    plt.ylabel('Predicted Values')
    plt.title(f'Actual vs. Predicted Values ({model_name})')
    plt.legend()
    plt.grid(True)
    plt.show()

# Plot for Decision Tree
plot_actual_vs_predicted(actual_values, predicted_values_dt, 'Decision Tree')
```

Visualizing the actual vs. predicted values in Decision Tree Regression



Model building- Model 3

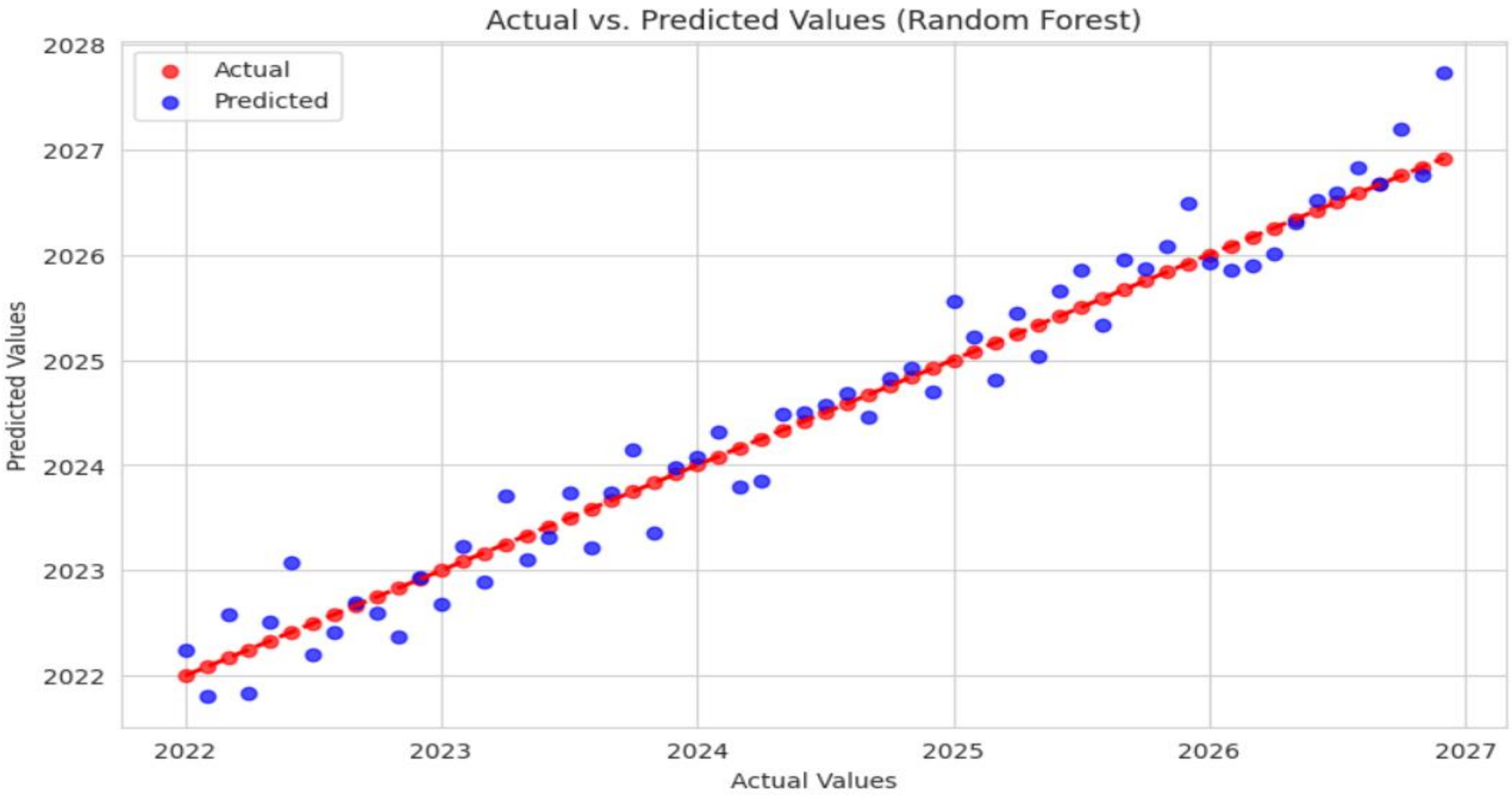
RANDOM FOREST REGRESSION

```
# Initialize the models
rf = RandomForestRegressor(random_state=42)

# Train and evaluate Random Forest Regressor
rf.fit(X_train, y_train)
y_pred_rf = rf.predict(X_test)
print('Random Forest')
print('MSE:', mean_squared_error(y_test, y_pred_rf))
print('R2 Score:', r2_score(y_test, y_pred_rf))
```

Visualizing the actual vs. predicted values in Random Forest Regression

```
def plot_actual_vs_predicted(actual, predicted, model_name):
    plt.figure(figsize=(10, 6))
    plt.scatter(actual, actual, color='red', alpha=0.7, label='Actual')
    plt.scatter(actual, predicted, color='blue', alpha=0.7, label='Predicted')
    plt.plot([min(actual), max(actual)], [min(actual), max(actual)], 'r--', lw=2)
    plt.xlabel('Actual Values')
    plt.ylabel('Predicted Values')
    plt.title(f'Actual vs. Predicted Values ({model_name})')
    plt.legend()
    plt.grid(True)
    plt.show()

# Plot for Random Forest
plot_actual_vs_predicted(actual_values, predicted_values_rf, 'Random Forest')
```

Visualizing the actual vs. predicted values in Random Forest Regression



Comparing accuracy of ML models

LINEAR REGRESSION	R2 Score: 1.0
DECISION TREE REGRESSION	R2 Score: 0.4975890443950213
RANDOM FOREST REGRESSION	R2 Score: 0.7235131640271839

Best Performing Model – Linear Regression

Superior R2 Score

- The Linear Regression model achieved an R2 score of 1.0, indicating a perfect fit to the data. This means that the model explains 100% of the variance in the target variable.
- Compared to the Decision Tree and Random Forest models, which had lower R2 scores, Linear Regression stands out as the most accurate model in predicting the rain forecasts. Given its perfect R2 score, simplicity, efficiency, robustness, and practical applicability, Linear Regression emerges as the best model for rain forecasting in this analysis. Its ability to perfectly explain the variance in the target variable, combined with its interpretability and computational advantages, makes it a superior choice compared to Decision Tree and Random Forest models.

CONCLUSION

The successful development and deployment of the Linear Regression model for rain forecasting demonstrate the power and utility of machine learning in predictive analytics.

The rain forecasting project demonstrates the potential of machine learning in enhancing predictive accuracy and operational efficiency in meteorology. By leveraging data-driven approaches, we can provide reliable forecasts that support various sectors, from agriculture to disaster management. The success of this project lays the groundwork for future innovations and improvements in weather forecasting technologies.

Thank You!