**Title Page**

**PROJECT TITLE:** PREDICTIVE MODELLING FOR COVID-19 IN PUBLIC HEALTH

**PREPARED BY:** CHRISTOPHER CHIBUIKE ENUKOHA

**FELLOW ID: FE/23/47729944**

**DATE:** 26/11/2024

A project presented to **HealthGuard Analytics** on covid-19 pandemic, using prediction models.

**Table of Content**

1.Introduction

2. Methodology

2.1 Problem Statement

2.2. Data preparation

2.3. Feature Engineering

2.4. Exploratory Data Analysis (EDA)

3. Model Development

3.1 Time-Series Modeling using RandomForestRegressor

3.2 Classification Modeling using RandomForestClassifier

Introduction

This project is on covid-19 model prediction building, the project will cover extensively all the methodology of data science, from feature engineering to conducting exploratory data analysis or better understanding of the matrix in the data set, down to model building which will be used for the predictions. In this project two models were used to do the predictions they are Time-Series Model where I used RandomForestRegressor to build the model for predictions, I used RandomForestRegressor because I am dealing a time record that’s records of the past to predict the future.

The second model used in this project is Classification Model I used RandomForestClassifier to build the model for predictions, this model works based on classification trees to be able to predict an out outcome.

Problem Statement

Using Predictive Modelling to build a predictive modeling system on covid-19 that will help HealthGuard Analytics to get actionable insights to inform policies, anticipate future outbreaks, and improve health resource allocation.

Objective

The primary objective of this project is to a build a model that will help to predict **COVID-19** cases based on various features such as confirmed cases, recovered cases, active cases, and new cases. The model was trained to help understand the spread of COVID-19 and identify trends within different data segments, which can be used for decision-making and resource allocation.

Data preparation

The Data for this project is from Kaggle, data cleaning and transformations were done for better model prediction building.

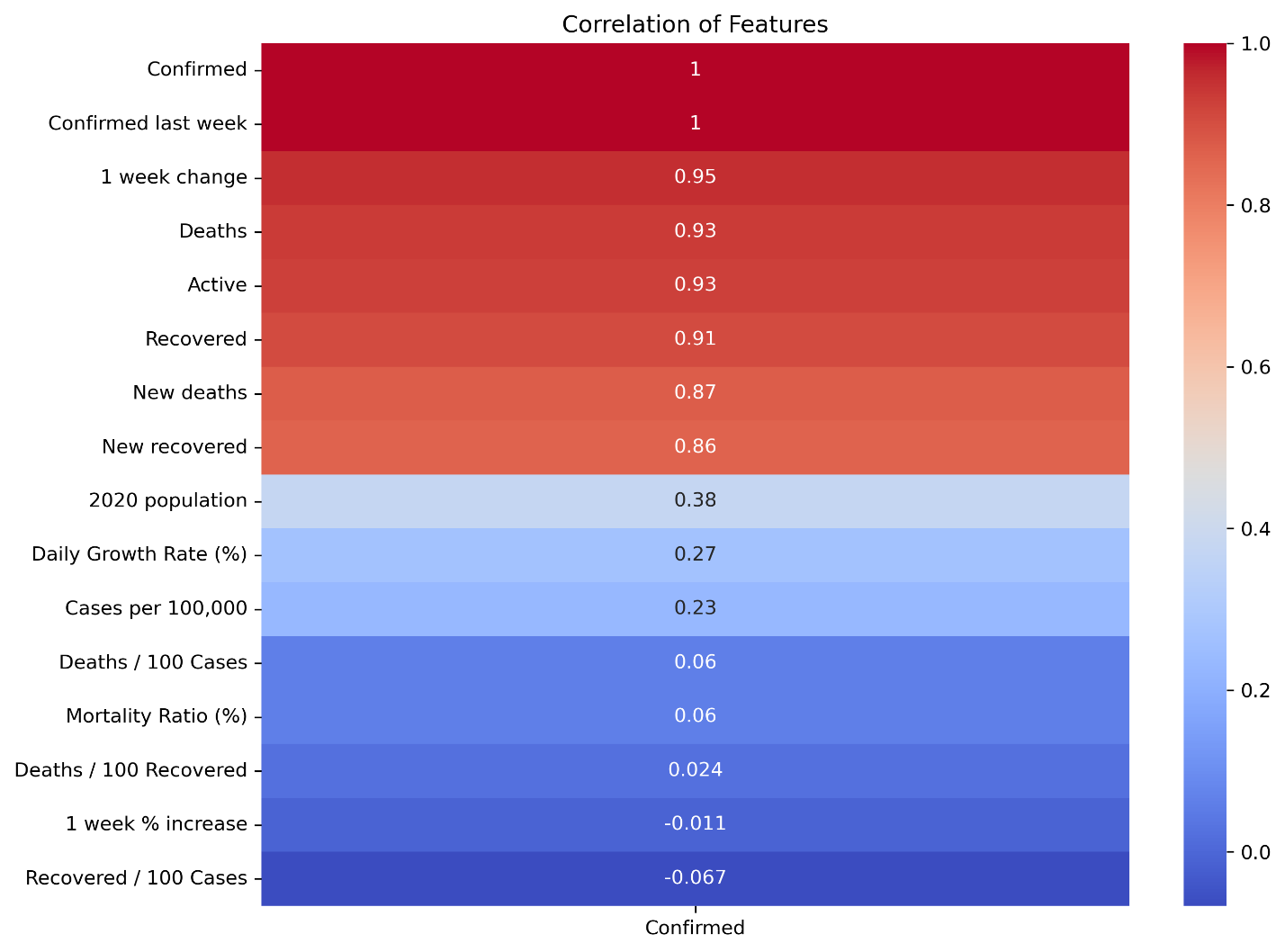
Feature Engineering

Feature engineering was conducted in the project for model better performance, the future engineering was done in two parts.

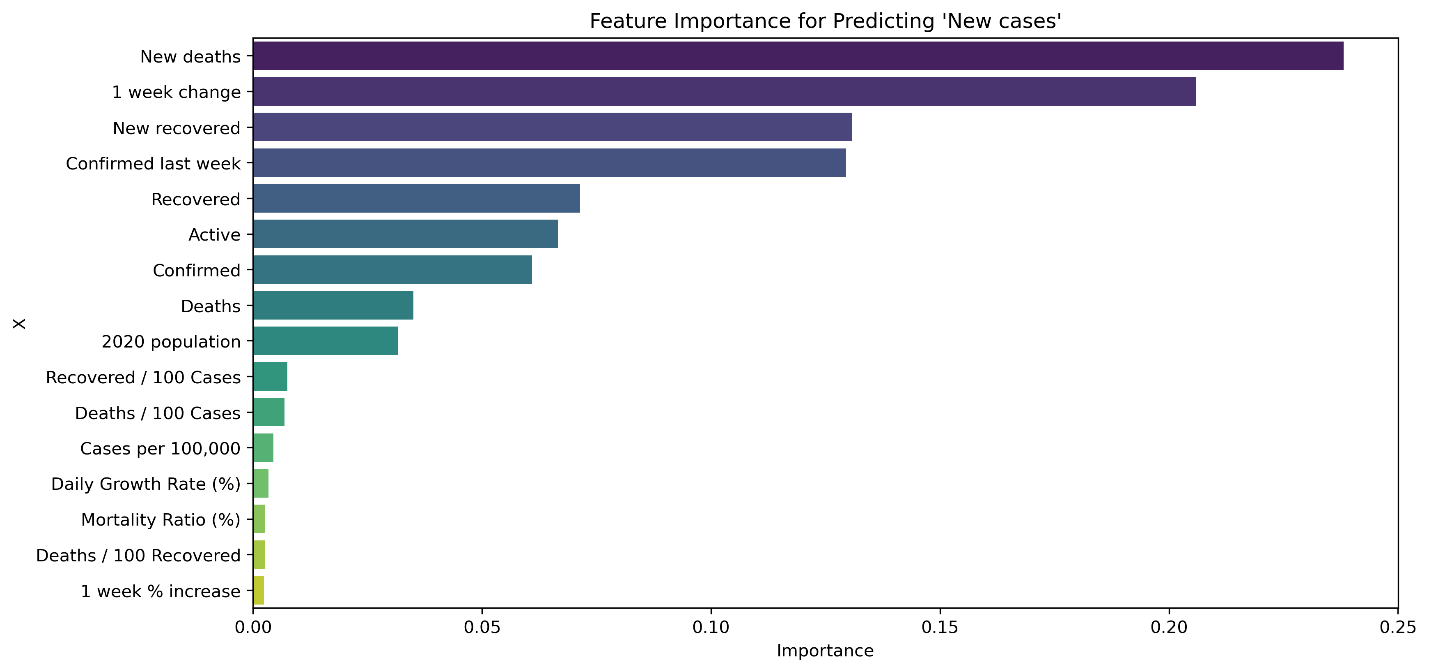
Feature generation: I generated new features to strengthen the prediction strength of the model the feature includes population, 100,000 cases per population, daily growth rates etc.

Feature selection: I conducted a feature selection using the Corr method and feature importance method. The feature selection played a great role in determining which feature is relevant to the model building.

**The Corr method**



The feature Importance



Exploratory Data Analysis (EDA)

I conducted exploratory data analysis to uncover key metrics in the datasets and the EDA products the following results.

1. USA has the highest number of confirmed cases.
2. The Distribution of Daily Growth Rate of covid-19 confirmed cases lies between 0 and 10000.
3. The highest distribution of cases per 100,000 of a country’s population is 100 cases to 100,000 population.
4. The highest distribution of Mortality Ratio % is 1% of each confirmed cases with a high frequency of over 40.
5. The country with the least confirmed cases is Fiji.

More the EDA can be accessed via ……………

Model Development

In this project two models were developed Times series mode and classification model.

Time-Series Modeling using RandomForestRegressor

**Model Summary**

The objective of this analysis was to develop a predictive model for covid-19 cases, forecasting the number of new cases . For this purpose, a Random Forest Regressor was chosen due to its robustness and ability to handle complex relationships in the data. The data was split into training and testing sets with a 70-30 split, and the model was trained and evaluated based on the test data.

**Evaluation Metrics**

The performance of the model was measured using several evaluation metrics:

* **Mean Absolute Error (MAE)**: **138.39**
* **Mean Squared Error (MSE)**: **92,052.65**
* **Root Mean Squared Error (RMSE)**: **303.40**
* **R-squared (R²)**: **0.663**

These metrics indicate the following:

1. **MAE (Mean Absolute Error: 138.39)**

* This measures the average absolute difference between the actual and predicted values. On average, the model's predictions are off by **138.39 units**.
* A smaller value means more accurate predictions. This is a straightforward metric showing the typical prediction error.

2. **MSE (Mean Squared Error: 92,052.65)**

* This calculates the average squared difference between actual and predicted values, giving more weight to larger errors.
* MSE is mainly useful for comparing models. The smaller the MSE, the better the model.

3.**RMSE (Root Mean Squared Error: 303.40)**

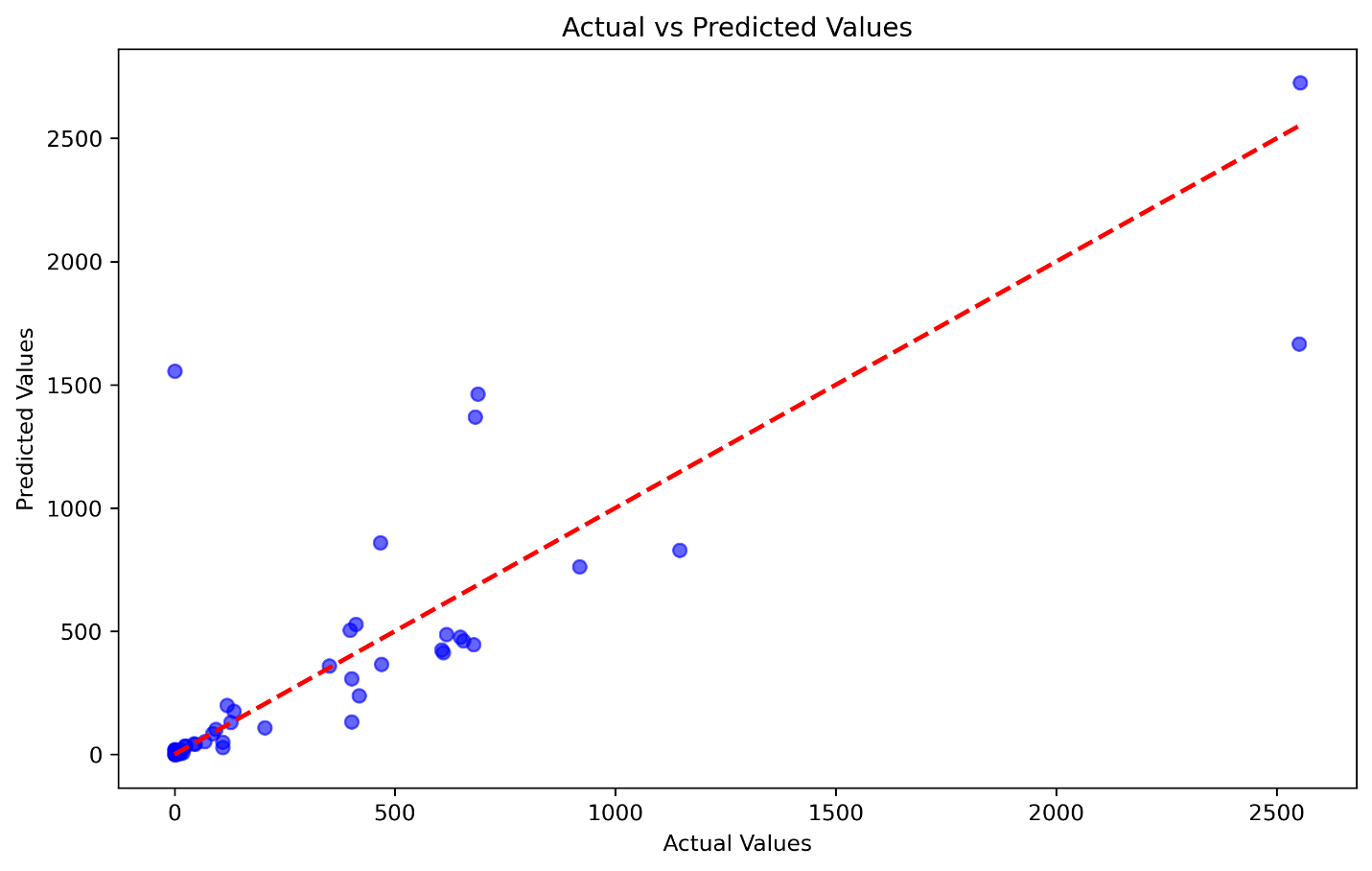
* This is the square root of the MSE, keeping the error in the same unit as the original data. The model’s typical prediction error is **303.40 units**.
* Like MAE, it tells how far predictions typically are from actual values, but it emphasizes larger errors more than MAE does.

4.**R² (R-squared: 0.663)**

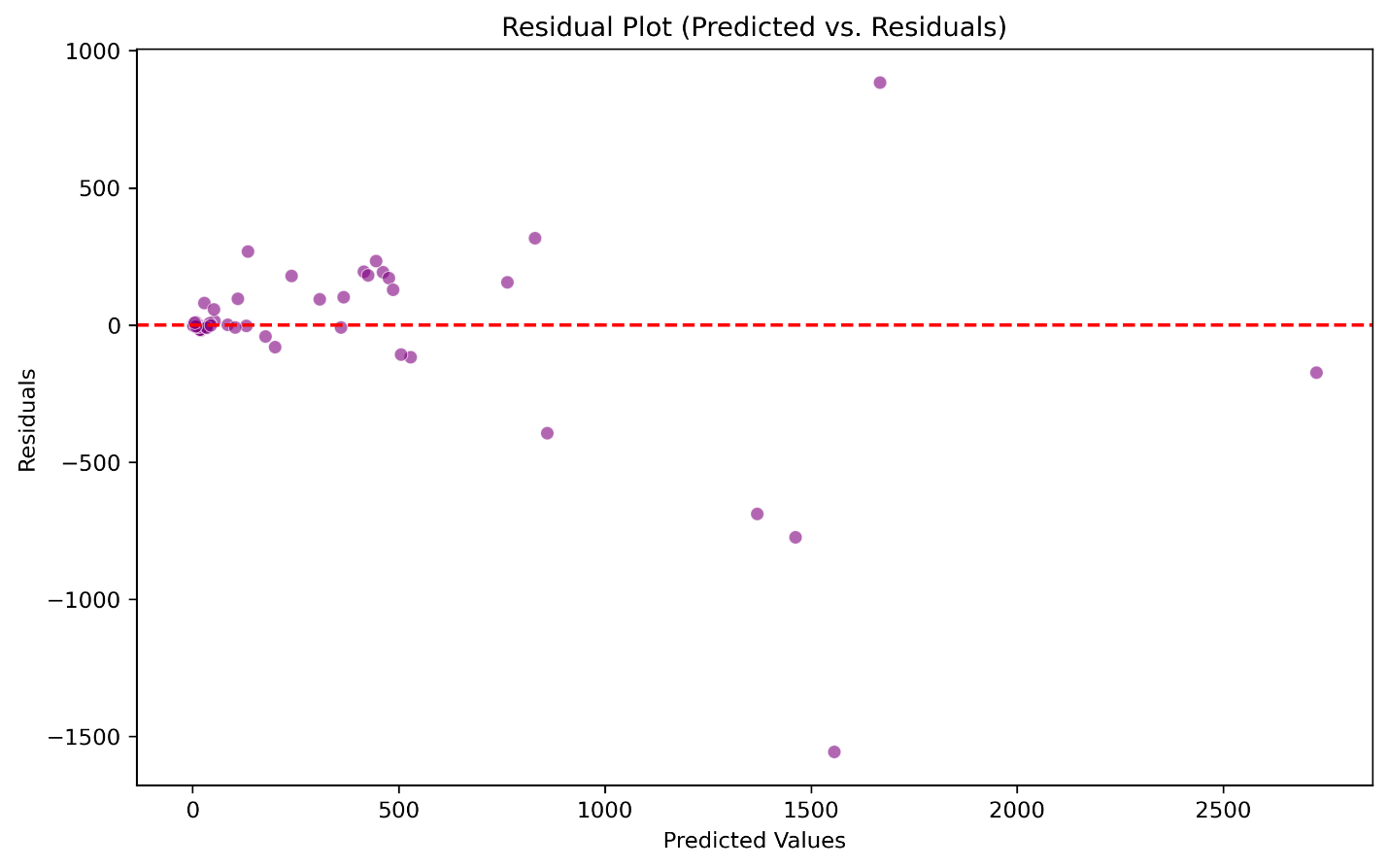
* This measures how much of the variation in the target variable is explained by the model. An R² value of **0.663** indicates that 66.3% of the variability in the target variable is captured by the model.

**Visualizations**

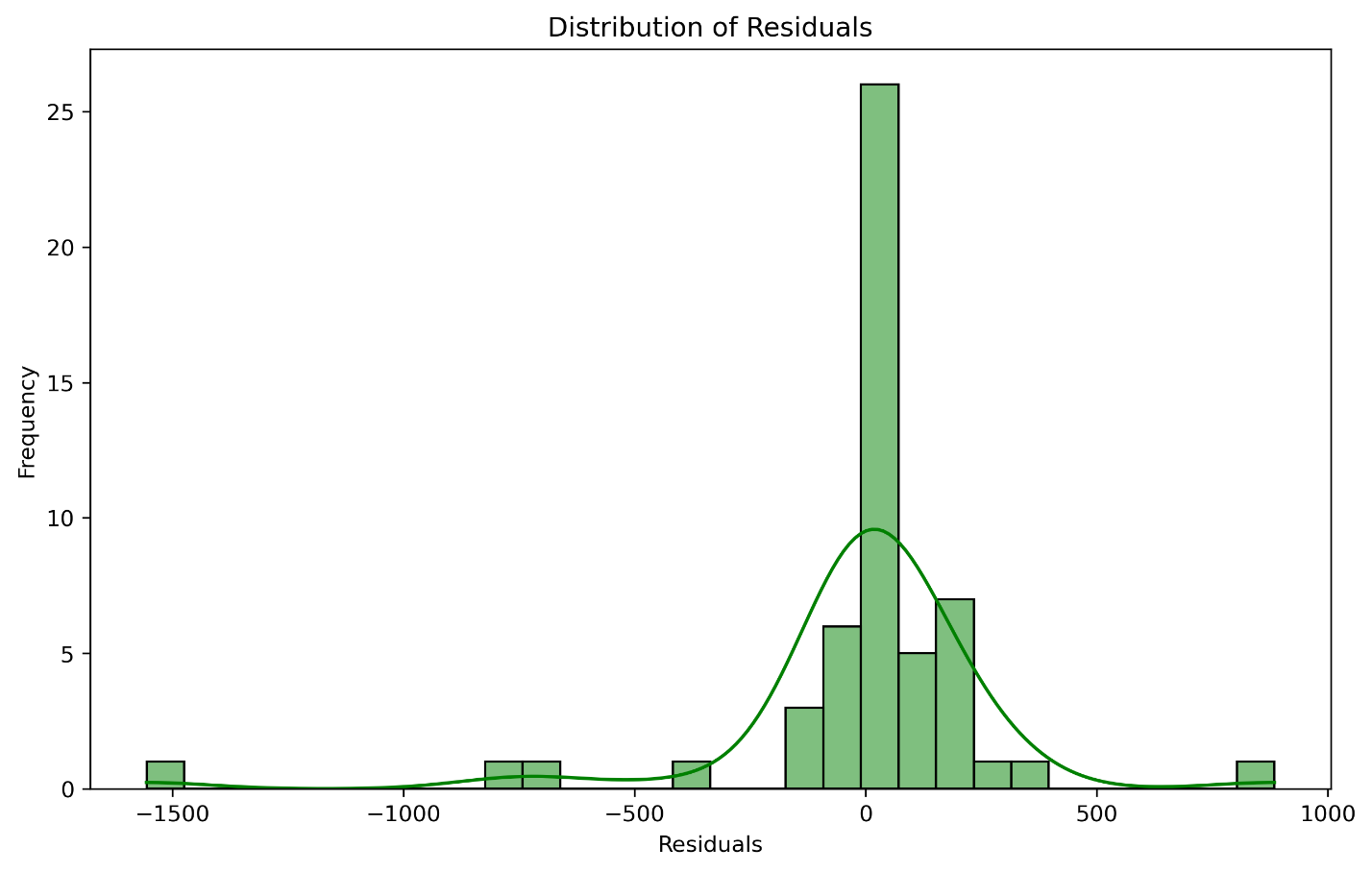
1. **Actual vs Predicted Plot**: This scatter plot (shown in the image below) illustrates the relationship between the actual and predicted values. Each point represents a prediction, with the actual value on the x-axis and the predicted value on the y-axis. The red dashed line represents the ideal case where predicted values perfectly match the actual values. Deviations from this line indicate prediction errors, with larger deviations observed in the higher range.



1. **Residual Plot**: This scatter plot below shows residuals versus predicted values. Ideally, residuals should be randomly scattered around zero. Here, most points are close to zero, but we see some outliers far from the center, suggesting that the model performs well overall but struggles with a few data points. These outliers might indicate areas for improvement or data points that differ significantly from the rest.



1. **Error Distribution Plot:** This histogram below shows the distribution of residuals, or prediction errors, for our model. The concentration of residuals around zero suggests that most predictions are close to actual values. However, the presence of a few extreme values (outliers) indicates occasional large errors.



**Conclusion**

Our model explains about 66.3% of the variability in the data, and the typical prediction error is around **138.39 units** (MAE) or **303.40 units** (RMSE). While this performance is reasonable, we may explore further improvements if the prediction accuracy needs to be higher for the use case.

Classification Modeling using RandomForestClassifier

A Random Forest Classifier was employed to handle the classification task. Random Forest is a robust ensemble learning method that reduces overfitting by building multiple decision trees and combining their results. Given its ability to handle high-dimensional data and its interpretability, Random Forest was a suitable choice for this classification problem.

The dataset was split into a training set and a test set with a 70-30 ratio to assess the model's performance on unseen data. The model was then trained on the training data and evaluated on the test set.

**Evaluation Metrics**

**1. Overall Accuracy:**

* The **accuracy** of 0.89 (88.8%) indicates that the model correctly predicts the outcomes for about 89% of the data points.
* While accuracy is a helpful general metric, it might not tell the whole story, especially if the data is imbalanced.

**2. Precision:**

* Precision measures the proportion of correctly predicted positive observations out of all predicted positive observations.
* Example: For class 0, a precision of **0.82** means that 82% of the predictions for this class are correct.
* Precision is crucial in applications where false positives are costly.

**3. Recall (Sensitivity):**

* Recall indicates how well the model identifies all the true positive cases.
* Example: For class 0, a recall of **1.00** means the model identified all actual cases in this class.
* Recall is essential in contexts where missing true cases (false negatives) has severe consequences.

**4. F1-Score:**

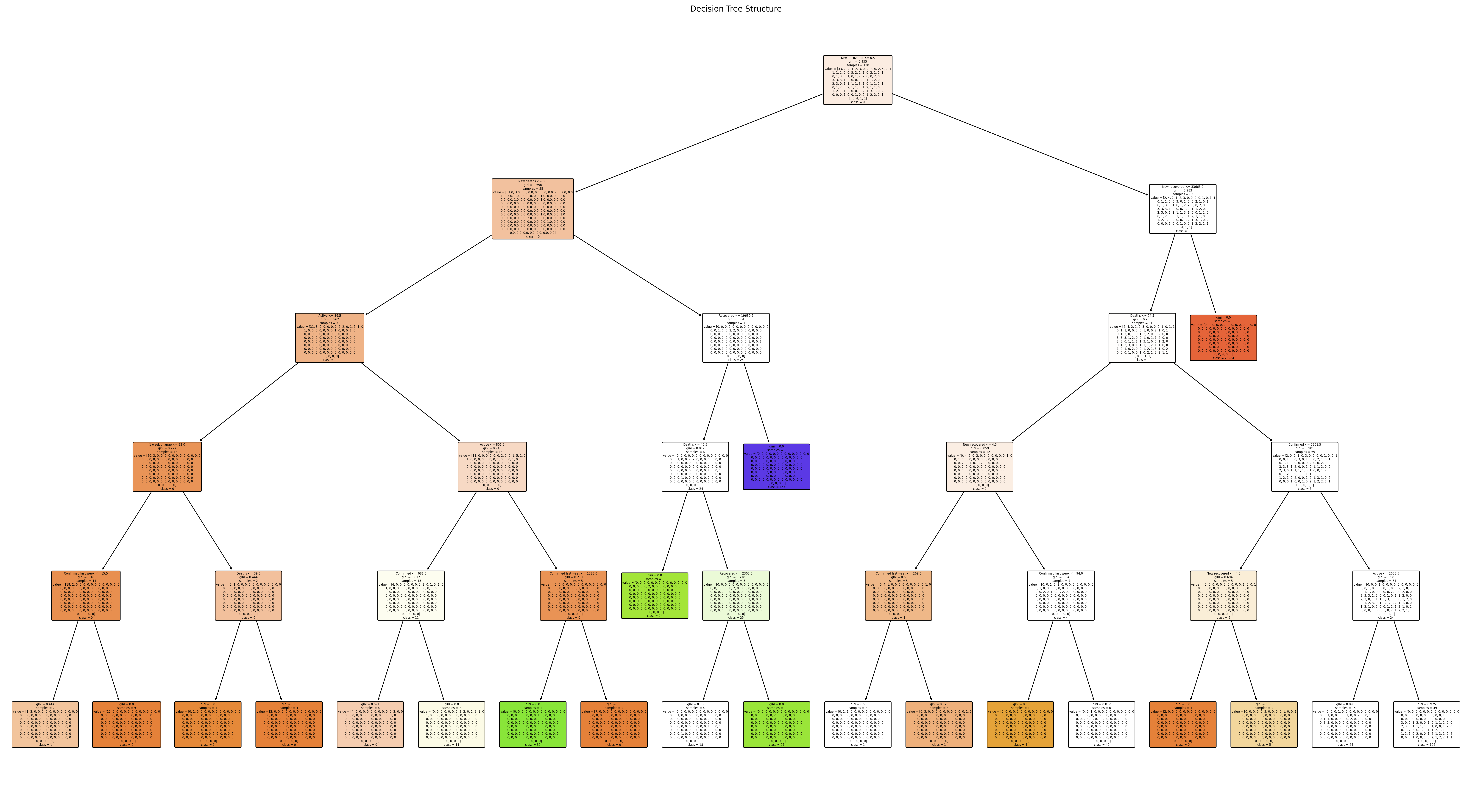
* This is the harmonic mean of precision and recall, providing a balanced metric for uneven class distributions.
* Example: The F1-score of **0.90** for class 0 balances its precision and recall performance.
* F1-Score is a good indicator of overall model performance per class.

**5. Support:**

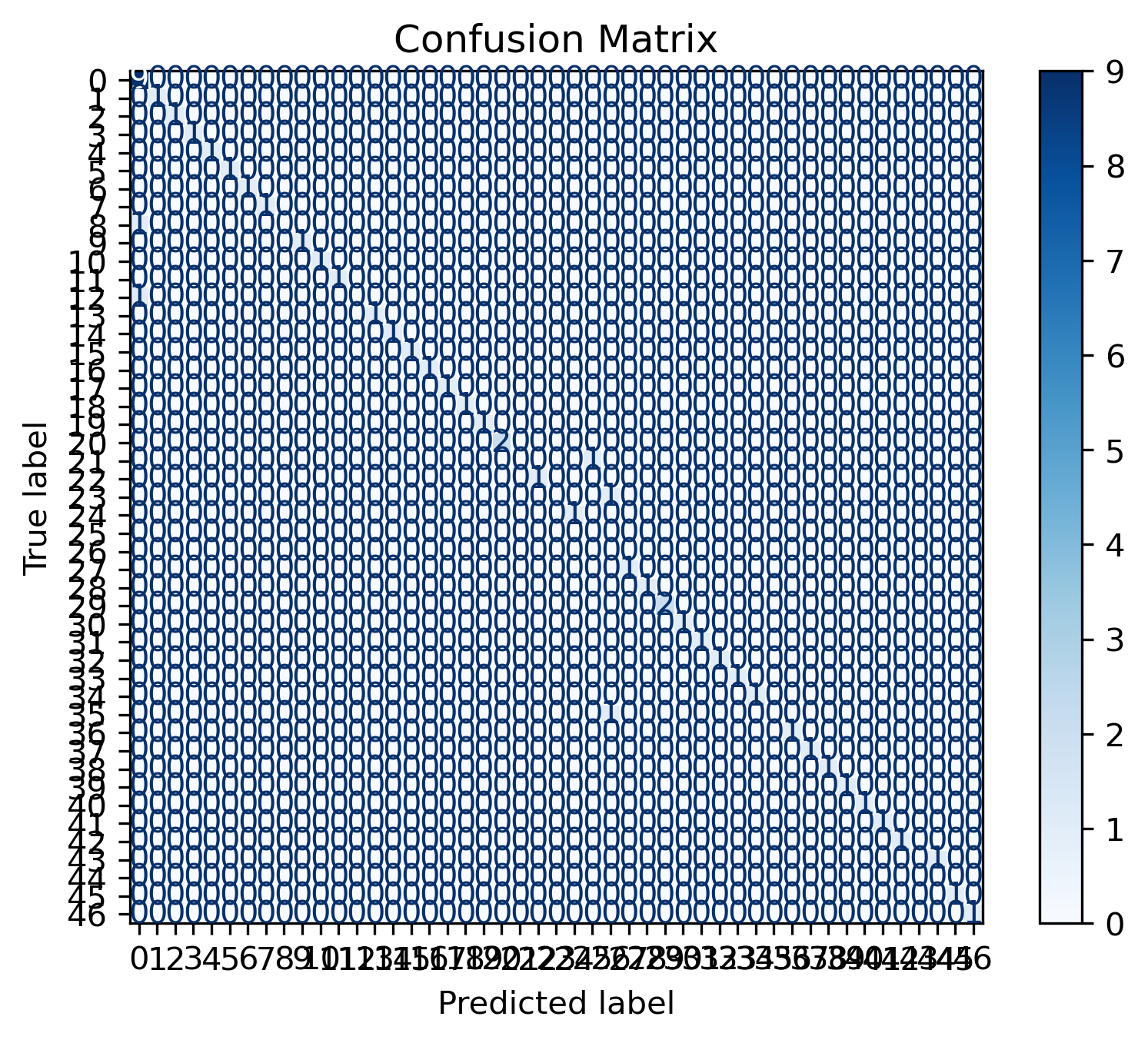
* This indicates the number of true instances for each class in the dataset.
* Example: For class 0, the model evaluated **9** instances.

Visual Analysis of Decision Tree

To further understand the decisions made by the model, a single decision tree was visualized from the Random Forest ensemble. The tree displays key decision points, showing how different features such as 'Confirmed cases', 'Recovered cases', and '1 week % increase' contribute to the model’s classification process.  
- Root Node: The root node uses 'Active cases' to split the data, indicating its significance in the classification process.  
- Decision Nodes: Each decision node provides a splitting criterion that maximizes the separation between classes, while leaf nodes at the bottom contain the final classification outcome for each branch.  
- Gini Impurity: Gini impurity values indicate the purity of the nodes, where lower values mean that nodes contain mostly samples from a single class.  
This tree helps identify important factors influencing classifications, making it easier to interpret the model's behavior.



**Confusion Matrix**



**Explanation of the Confusion Matrix:**

1. **True Labels (Rows)**: The vertical axis (rows) represents the true class labels.
2. **Predicted Labels (Columns)**: The horizontal axis (columns) represents the predicted class labels.
3. **Diagonal Cells**: These represent the correct predictions (true positives). A higher intensity along the diagonal indicates better model performance.
4. **Off-Diagonal Cells**: These are the incorrect predictions (false positives and false negatives). The lighter color here suggests fewer misclassifications.

**Key Points**

1. **Dominant Diagonal**:
   * The confusion matrix shows a strong diagonal line, which indicates that the majority of the predictions are correct.
   * The model is effective at distinguishing between classes.
2. **Error Analysis**:
   * The off-diagonal values (non-zero values away from the diagonal) represent misclassifications. These areas can help identify problematic classes where the model struggles.
   * This insight can guide improvements or adjustments, such as adding more data or tweaking model parameters.
3. **Class Imbalance**:
   * Some rows or columns may have very few values. This could indicate class imbalance, which might need to be addressed if it affects important categories.

The confusion matrix indicates that the model performs well, with most predictions being accurate as seen along the diagonal. Any misclassifications are sparse, suggesting the model has minimal errors. We can analyze misclassified cases further to see if improvements are necessary for specific classes.

**Conclusion**

Our classification model achieves an overall accuracy of **88.8%**, with strong precision, recall, and F1-scores across most classes. For example, class 0 has a precision of **82%**, a recall of **100%**, and an F1-score of **90%**. These metrics show the model’s reliability in identifying and correctly classifying data, especially for smaller classes.