Project Name: Pre-Hurricane Alarm

Project Description

Developed a machine learning model to predict topological disorder from Northern Atlantic waves, which is essential in detecting and forecasting hurricane activities. Utilizing data from the Google Earth API, the model analyzes key parameters such as wind speed, pressure, and proximity to land. This model is integrated into a real-time alert system, enhancing disaster preparedness for regions at risk of hurricanes. The project showcases advanced pattern recognition and predictive modeling, aiming to improve environmental safety through early detection and alerts.

Team Members:

Krushna Thakkar

Akshar Gothi

Naing Htet

Context:

Project Brief Description

1. Problem Statement
2. Past-Current Mid Affects

Data Planning and Data Engineering

1. Data Source
2. Data Description
3. Data Extraction

Exploratory Data Analysis

1. Data Filtering and Cleaning
2. Handling Missing Values
3. Feature Engineering
4. Outlier Detection
5. Data Visualization
6. Categorical Encoding
7. Preprocessing and Model Building

Machine Learning and Algorithms

1. Testing Models
2. Evaluation and Metrices

Summary

Project Brief Description

The Pre-Hurricane Alarm project is designed to improve disaster preparedness by developing a machine learning model capable of detecting and forecasting hurricane activities and other topological disorders. The system leverages 33 years of historical data and integrates real-time data tracking from sources like the Google Earth API, NOAA and data. Government websites ensuring up-to-the-minute analysis of critical weather parameters such as wind speed, pressure, and proximity to land.

In addition to hurricanes, the system can detect a wide range of other natural disturbances, including extratropical cyclones, gale winds, subtropical depressions, subtropical storms, tropical depressions, and more. These disturbances are often precursors to larger calamities, and the model helps identify patterns that contribute to these events.

The real-time data tracking feature allows the system to continuously monitor and update predictions, offering governments and communities immediate insights and alerts. To enhance user experience, the project also incorporates bot assistance to provide real-time notifications and personalized updates about potential risks directly to users, improving response times and decision-making.

By analyzing vast amounts of historical data alongside real-time information, this predictive system not only enhances hurricane forecasting but also offers broad disaster alert capabilities, aiming to safeguard lives and infrastructure.

Real-Life Impact

- **Hurricane Maria (2017)**
  Devastated Puerto Rico, with winds reaching up to 175 mph, causing a complete power grid collapse and over 3,000 deaths.
- **Hurricane Harvey (2017)**
  Led to unprecedented rainfall in Texas, causing severe flooding and damages surpassing $125 billion.
- **Hurricane Sandy (2012)**
  Struck the northeastern U.S., causing $70 billion in damages and flooding in New York City.
- **Hurricane Dorian (2019)**
  Struck the Bahamas with 185 mph winds as a Category 5 storm, highlighting the vulnerability of small island nations.

Data Planning and Data Engineering

Dataset

1. NOAA Atlantic Hurricane Dataset

- **Description:** This dataset is part of the NOAA Atlantic hurricane database (HURDAT) and includes the best track data for storms in the Atlantic basin. It covers storm positions and attributes from 1975 to 2021, with measurements recorded every six hours for storms from 1979 onward. Earlier data may have gaps, but it provides critical insight into storm tracks, intensity, and key weather attributes.
- **Link**: NOAA Atlantic Hurricane Dataset
- **Size**: The dataset includes 19,066 observations and 13 variables.

---

2. **NCDC Storm Events Database**

- **Description:** The NCDC (National Climatic Data Center) Storm Events Database provides comprehensive data on various weather phenomena in the United States, ranging from hurricanes to tornadoes, hailstorms, thunderstorms, floods, and more. It also includes personal injury data, property damage estimates, and storm-related statistics from 1950 to the present. The database is updated monthly and may have a delay of up to 120 days.
- **Link:** NCDC Storm Events Database
- **Timeframe:** 1950–present.
- **Coverage:** United States.

---

3. International Best Track Archive for Climate Stewardship (IBTrACS)

- Description: The IBTrACS dataset is a comprehensive global tropical cyclone database, providing information on the location and intensity of cyclones from as far back as the 1840s up to the present day. It offers 3-hour interval data on storm tracks, with additional parameters such as maximum sustained wind speed, minimum central pressure, and environmental conditions. It allows for filtering by basin or time and includes regions like the East Pacific, North Atlantic, South Atlantic, and others.
- Link: IBTrACS on Google Earth Engine
- Dataset Provider: NOAA NCEI (National Centers for Environmental Information).
- Timeframe: 1842 to 2024 (ongoing).
- Earth Engine Snippet:

- ee.FeatureCollection("NOAA/IBTrACS/v4")
- FeatureView for visualization: ui.Map.FeatureViewLayer("NOAA/IBTrACS/v4_FeatureView

These datases provide critical information on hurricane activity, storm-related events, and global tropical cyclones, offering the data needed to build the predictive models used in the Pre-Hurricane Alarm system. They enable analysis of past storms and real-time tracking of weather phenomena, enhancing the system's ability to forecast hurricane activities and other weather-related disturbances.

Risks Involved

1. Data Inconsistency
    - Risk: Since the datasets come from various sources (e.g., NOAA, Google Earth, NCDC), there may be inconsistencies in data formats, units, or measurement intervals (e.g., different time intervals between storm data).
    - Mitigation: Consistent unit conversion, timestamp alignment, and clear documentation of dataset specifics. Ensuring proper handling of time zone differences and daylight-saving time changes.
2. Missing Data
    - Risk: Historical datasets, especially older ones, may have missing or incomplete records, such as gaps in storm track positions or missing wind speed measurements.
    - Mitigation: Use interpolation techniques to fill missing values or imputation for non-time series data. Alternatively, omit incomplete rows if necessary, or use advanced imputation techniques like K-Nearest Neighbors (KNN) or multiple imputations.
3. Data Leakage
    - Risk: Data from the future may inadvertently be used to predict outcomes, leading to overly optimistic model performance (e.g., using data points recorded after the storm has ended in the training set).
    - Mitigation: Strict separation of training and testing data based on time, ensuring no future data is included in model training.
4. Overfitting due to Feature Engineering

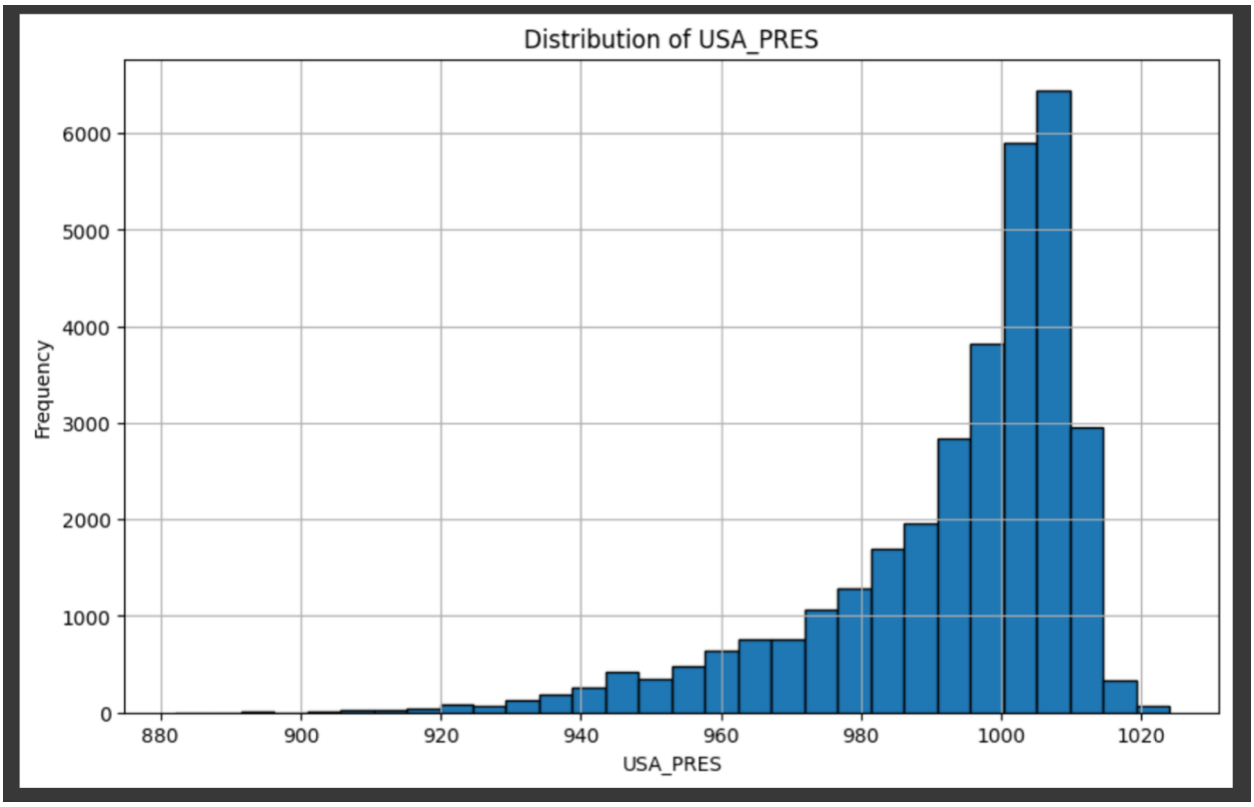
- Risk: Excessive feature engineering can create models that fit the training data too well, leading to poor generalization to new data.
- Mitigation: Apply regularization techniques and cross-validation to ensure that features do not introduce overfitting. Keep features interpretable and directly related to the physical phenomena being modeled.

5. Bias in Historical Data
    - Risk: Older data may reflect outdated measurement techniques or biases in how storms were recorded and classified (e.g., underreporting of storms in earlier decades).
    - Mitigation: Use modern data wherever possible for model training, or account for biases by applying correction factors when dealing with historical records.
6. Merging Errors
    - Risk: Incorrect joining of datasets (e.g., mismatching time intervals, misaligning storm IDs, or geographic coordinates) can lead to errors in the final dataset, affecting model accuracy.
    - Mitigation: Use careful inspection and validation of joins. Test on a small sample of data before performing large-scale joins. Ensure common keys are accurate and properly aligned.
7. Performance and Memory Constraints
    - Risk: Large datasets may cause memory overflow or slow down processes during manipulation, particularly when applying complex transformations or feature engineering.
    - Mitigation: Use efficient data structures (e.g., pandas) to handle large datasets, and apply memory optimization techniques like chunking and parallel processing.

Exploratory Data Analysis (EDA)

1. **Data Filtering and Cleaning**

- Columns Selection: The code starts by selecting specific columns (columns_to_keep) from the hurricane dataset to focus only on the relevant features (e.g., wind speed, pressure, coordinates, etc.).
- Missing Values: It checks the dataset for missing values using .isnull().sum(). The results are sorted and displayed to identify which columns have the most missing values.
- Column Dropping: Columns with high numbers of missing values that are deemed unnecessary (USA_RECORD, SUBBASIN) are dropped to clean the data.



2. **Handling Missing Values**

- Missing values in critical columns like USA_PRES (pressure) are handled by filling them with the mean value of that column. This prevents losing rows in the dataset that have NaN values.

- The column USA_STATUS is filled with the value 'Unknown' where the data is missing to ensure consistency for classification.

3. **Feature Engineering**

- Datetime Extraction: The ISO_TIME column, which contains timestamps, is broken down into Year, Month, Date, Hour, Minute, and Second to allow more granular analysis and potentially uncover time-based patterns.
- Day/Night Classification: A new column, Day_Night, is created based on the hour value. It assigns 'Day' for hours between 6 AM and 6 PM, and 'Night' for the remaining hours. This feature might influence hurricane behavior.
- Wind-Pressure Ratio: A new feature wind_pressure_ratio is created by dividing wind speed by pressure, which could be useful in understanding the relationship between these two variables during hurricane events.

4. **Outlier Detection**

- The Interquartile Range (IQR) method is used to detect outliers in numerical columns (USA_WIND, USA_PRES, etc.). Outliers are data points that fall outside 1.5 times the IQR above the third quartile or below the first quartile. These can impact model performance and understanding their presence is important.

5. **Data Visualization**

- Histogram of USA_PRES: A histogram is plotted to visualize the distribution of pressure values (USA_PRES), which helps to understand the central tendency, spread, and skewness of the data.
- Bar Plot for USA_PRES: A bar plot is created for unique values in USA_PRES to observe the frequency distribution of pressure values.

6. **Categorical Encoding**

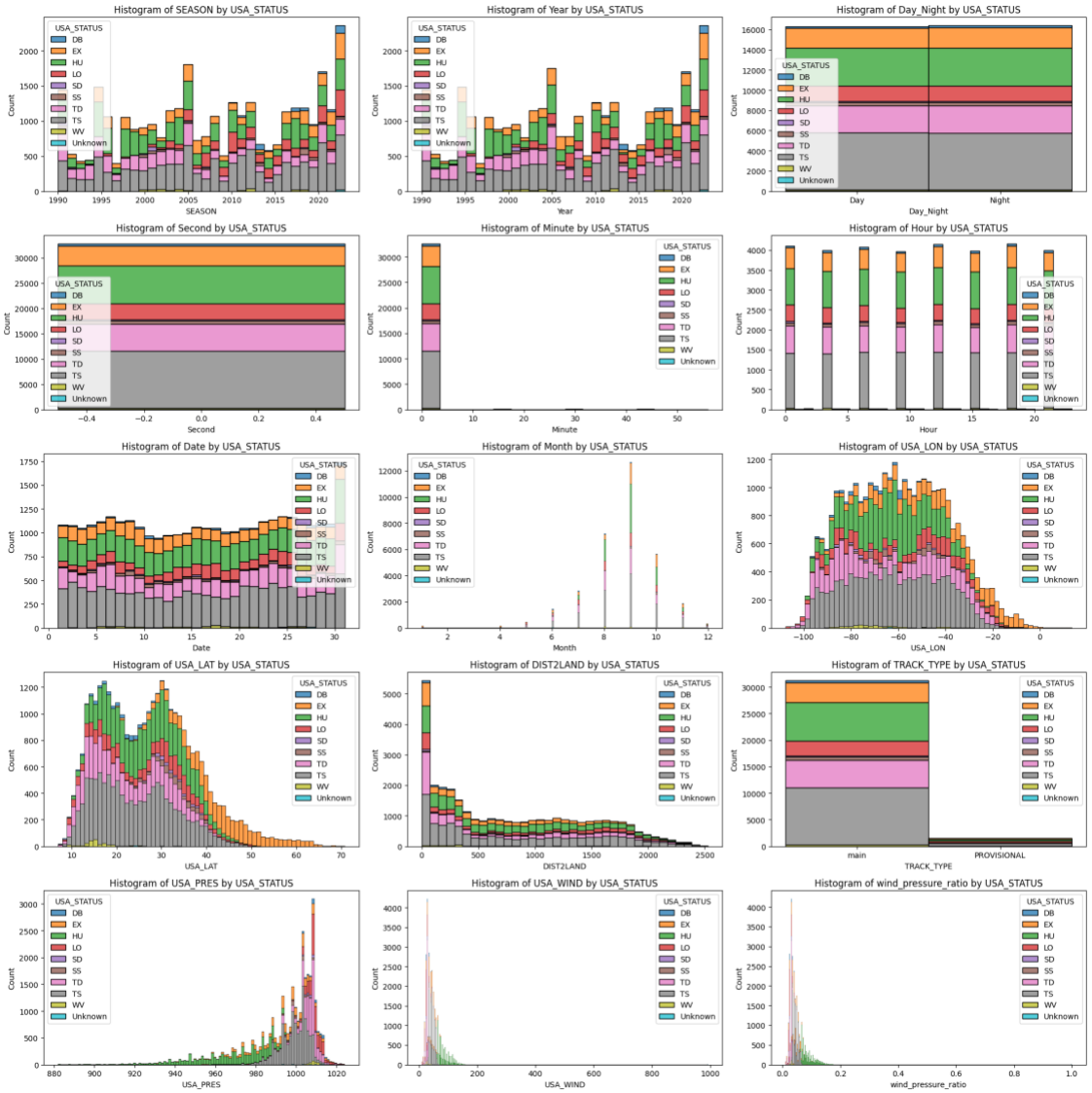
- The Day_Night column is converted into numerical values (0 for 'Day' and 1 for 'Night') using label encoding to prepare it for machine learning models.

7. **Preprocessing and Model Building**

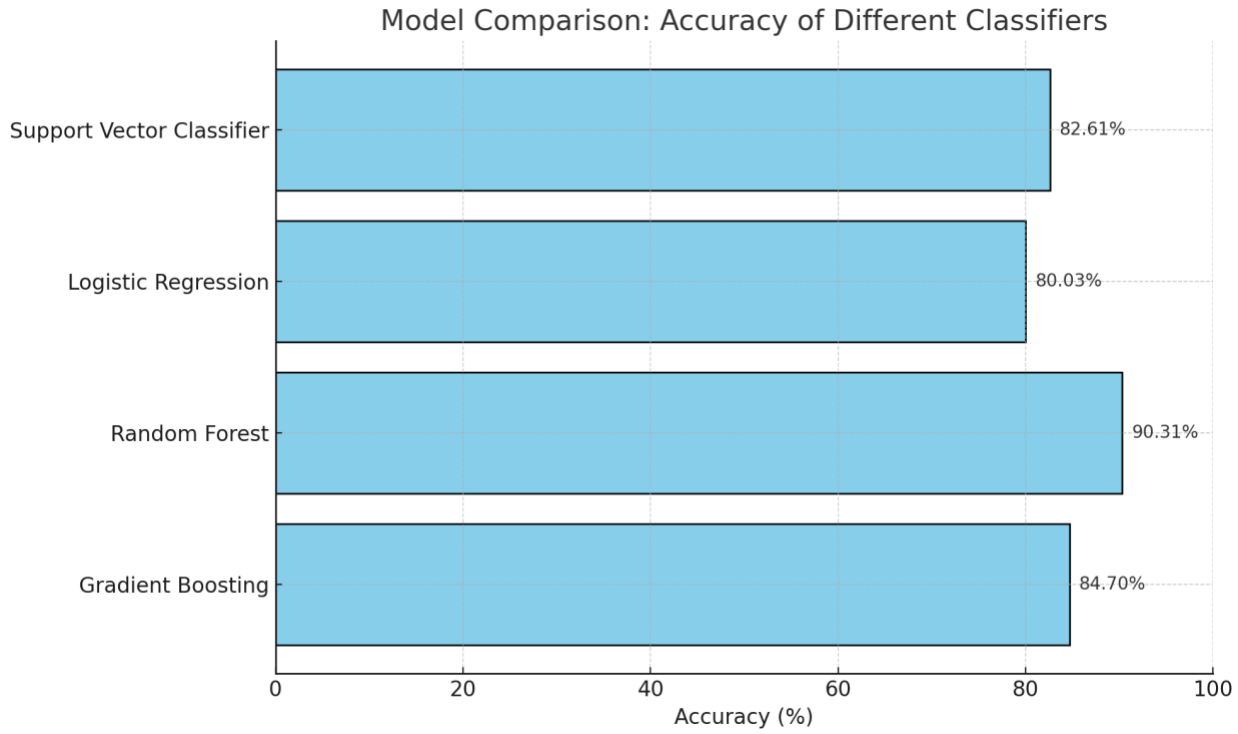
- Standardization: The numerical features (USA_WIND, USA_PRES, etc.) are standardized using StandardScaler to ensure that all features contribute equally to the model performance by having a mean of 0 and a standard deviation of 1.
- Data Splitting: The dataset is split into training and testing sets with 80% of the data used for training and 20% for testing.

Summary of EDA Steps:

- Data Cleaning: Handling missing values and dropping irrelevant columns.
- Feature Engineering: Creating new columns like Day_Night and wind_pressure_ratio to enhance the dataset.
- Outlier Detection: Identifying extreme values that might skew the model.
- Visualization: Understanding data distribution through visualizations.
- Encoding: Converting categorical data into a numerical format.



Machine Learning and Algorithms



Gradient Boosting Classifier:

- **Accuracy:** 84.70%
- **How it works**: Gradient Boosting is an ensemble technique that builds multiple weak learners (typically decision trees) in a sequential manner, where each new model attempts to correct errors made by the previous models.
- **Confusion Matrix Analysis:**
    - The model struggles in certain classes (e.g., class 0 and class 4), as seen in their lower recall and precision. For instance, class 0 (precision of 0.77 and recall of 0.32) is poorly predicted, with many instances being misclassified into other categories.
    - Class 2, which has the most data, performs very well with high precision and recall (~98% for both).
- **Overall:** Although the accuracy is relatively high, the confusion matrix shows that the model tends to overfit the most common classes while underperforming on the rarer ones.

Random Forest Classifier:

- **Accuracy:** 90.31%
- **How it works**: Random Forest is an ensemble method that builds multiple decision trees, each trained on a random subset of the data and features. The final prediction is made by averaging the predictions of all trees.
- **Confusion Matrix Analysis:**
    - This model provides more balanced predictions across most classes. For example, in class 0, it achieved a precision of 83% and a recall of 60%, which is a significant improvement over Gradient Boosting.
    - The most populous class (class 2) performs exceptionally well with precision and recall near 98%.
    - Even the underrepresented classes (like class 4 and class 9) perform relatively well compared to the other models.

Why Choose Random Forest: Random Forest tends to provide higher accuracy and more robust results across different classes. It mitigates overfitting and tends to generalize better. It is more accurate, particularly with hard-to-predict and minority classes, as shown by its confusion matrix.

Logistic Regression:

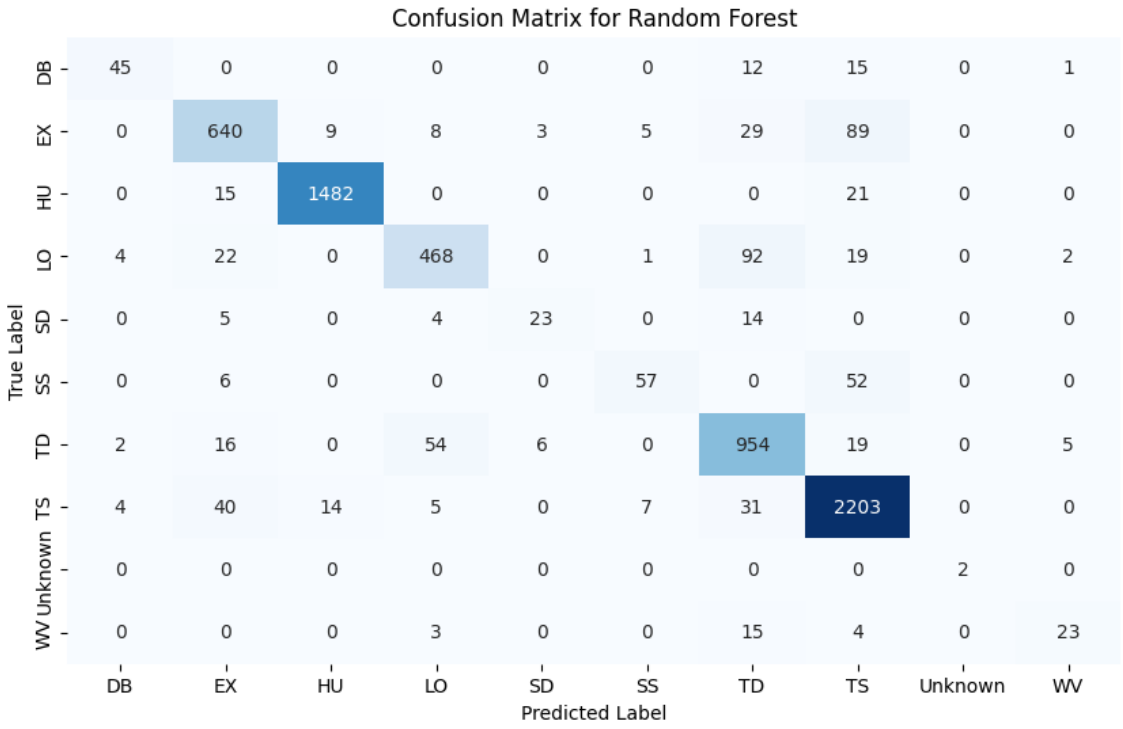
- **Accuracy:** 80.03%
- **How it works:** Logistic Regression is a simple linear model used for classification. It estimates the probability of a class using a logistic function and classifies based on the highest probability.
- **Confusion Matrix Analysis**:
    - The model fails to predict several classes entirely (e.g., class 0 and class 4 show precision and recall of 0), indicating its weakness in handling complex, non-linear relationships in the data.
    - The confusion matrix shows that it tends to misclassify minority classes as larger classes. For example, class 1 has reasonable performance, but rarer classes like 0 and 4 are poorly predicted.
- **Overall:** Logistic Regression, despite being simple, is not suitable for this task, as evidenced by its lower accuracy and poor handling of multi-class imbalances

Support Vector Classifier (SVC):

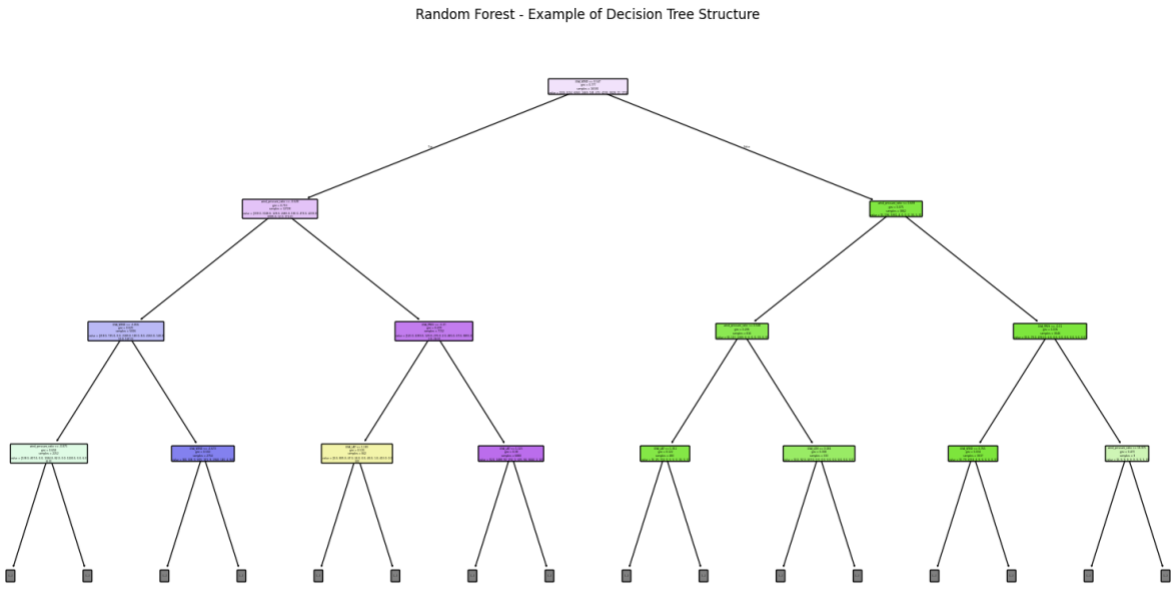
- **Accuracy:** 82.61%
- **How it works**: SVC is a classifier that finds the optimal hyperplane to separate different classes in the feature space. It can perform well in both linear and non-linear separations.
- **Confusion Matrix Analysis:**
    - The model performs similarly to Gradient Boosting, with solid performance for large classes like class 2 (precision and recall near 99%).
    - However, it performs poorly for minority classes (e.g., class 0, class 5), where precision and recall are zero, indicating significant misclassification for rare categories.
- **Overall:** SVC performs reasonably well, but its inability to handle minority classes leads to lower effectiveness in this dataset, where class imbalance is a concern.

Why We Choose Random Forest:

Random Forest outperforms other models by delivering balanced performance across both common and rare classes, with higher precision and recall for problematic classes like 0 and 4. Its accuracy of 90.31% highlights its strong generalization ability. Unlike models such as Logistic Regression and SVC, Random Forest handles class imbalances effectively by averaging multiple decision trees, preventing overfitting to the majority class. It also shows greater robustness and versatility compared to Gradient Boosting, offering consistent results for multi-class problems. The confusion matrix further confirms its superior recall and lower misclassifications, making it the best choice for this dataset.



Random Forest Classifier



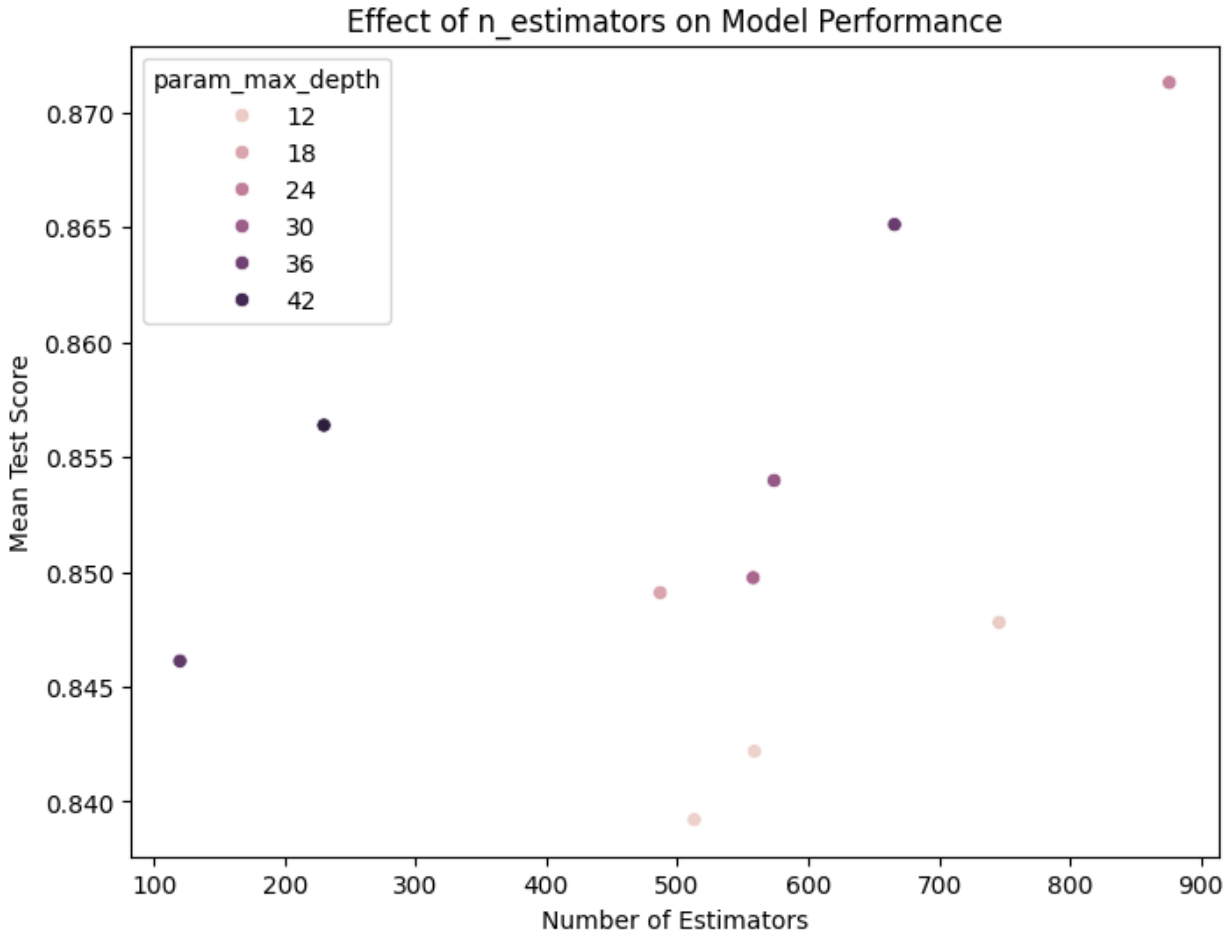
Hyper-Parameter Tuning

Hyperparameter tuning is the process of selecting the best set of hyperparameters for a machine learning model.

In Random Forest models, for instance, hyperparameters such as the number of trees (n_estimators), the maximum depth of each tree (max_depth), and the minimum number of samples required to split a node (min_samples_split) affect the model's performance. Tuning these values is essential for getting the best performance.

RandomizedSearchCV:

RandomizedSearchCV is a method that searches through a predefined grid of hyperparameters and tries random combinations of the values to find the best ones. It's faster than GridSearchCV, which exhaustively tests all possible parameter combinations, because it randomly selects a subset of the combinations to try.



Summary

The **Pre-Hurricane Alarm** project is focused on developing a machine learning model that predicts topological disorder in Northern Atlantic waves, helping detect and forecast hurricane activities. The system integrates data from the Google Earth API, NOAA datasets, and real-time tracking to analyze critical weather parameters such as wind speed, pressure, and proximity to land. This project aims to enhance disaster preparedness by providing early detection and alert systems for hurricane-prone regions.

The model is trained on 33 years of historical data and includes real-time tracking from sources like NOAA, ensuring accurate and up-to-date analysis. It also extends its capabilities beyond hurricanes to detect other natural disturbances like extratropical cyclones, gale winds, and tropical depressions. The system offers real-time notifications and alerts to assist government bodies and communities in making informed decisions quickly, minimizing the impact of natural disasters.

Data Engineering & Data Sources

- **NOAA Atlantic Hurricane Dataset** (1975–2021): Tracks storm positions and intensity.
- **NCDC Storm Events Database** (1950–present): Provides comprehensive weather event data, including hurricanes.
- **IBTrACS Dataset**: A global tropical cyclone database providing detailed storm information from the 1840s onward.

Machine Learning Models

- The project tested several machines learning algorithms, including **Random Forest**, **Gradient Boosting**, **Logistic Regression**, and **Support Vector Classifier (SVC)**. After evaluating performance, **Random Forest Classifier** was chosen due to its balanced performance across both common and rare hurricane classes, achieving an accuracy of **90.31%**. The model was hyperparameter-tuned using **RandomizedSearchCV** for optimal performance.
- **Impact**
- The Pre-Hurricane Alarm system offers significant benefits, such as early detection and alert notifications for hurricanes and other storm-related disasters. By combining historical data with real-time analysis, this system provides essential insights to safeguard lives and infrastructure.
- Real-life case studies like **Hurricane Maria** (2017) and **Hurricane Dorian** (2019) demonstrate the potential of this system to mitigate disaster impacts and enhance response time.