Predicting California Wildfire Causes with Machine Learning: A Data-Driven Approach

Report

Annissa A. Pereira
apereira26@umassd.edu
Department of Data Science
University of Massachusetts, Dartmouth
Student ID: 02126803

DSC 550: Masters Project
Project Advisor: Dr. Amir Akhavan Masoumi

Spring 2025

Abstract:

The increasing severity of wildfires in California, particularly those caused by human activity, motivated the development of this project. With climate change, population growth, and vast human development near wildland areas, correctly identifying the cause of wildfire has become an important task for effective prevention, faster response and quick resource allocation [1][2][3].

The state of California in United States of America faces continuous wildfire crisis, with thousands of incidents happening each year. A vast portion of these fires are human caused, still distinguishing between human and natural causes remains challenging [4]. Recent strategies are limited in predictive capability, leaving authorities without actionable insights before fires escalate.

To address this underlying issue, this project influenced machine learning to build a classification model that can predict the cause of wildfire (human or natural). The dataset was constructed by joining wildfire incident data from the Fire and Resource Assessment Program (FRAP) with meteorological records from NOAA using spatial proximity [5][6]. Initially Logistic regression was considered as a baseline model and then ensemble models were trained as Random Forest, XGBoost and Light Gradient Boosting. After preprocessing, carrying out cross validation, engineering and tuning, best features were chosen automatically by SelectKBest with mutual information, the resulting features latitude, longitude, windspeed/direction, temperature extremes, and time of day were chosen as the most influential features. As the location of the place matters, the temporal features matters hereby giving us proper classification of the cause.

Different classification models were trained and evaluated, Logistic Regression, Random Forest, LightGBM, and XGBoost. Among all these models, the Random Forest Classifier model achieved accuracy of 94% which was the highest and it demonstrated strong performance in guessing the correct number of human-caused wildfire and natural wildfires. The model was then deployed in Flask, making it usable for real time inference.

This study highlights the power of data fusion and machine learning for addressing big environmental problems. Future work may involve handling missing causes and experimenting with advanced deep learning techniques.

Table of Contents

Chapter 1: Introduction

1.1 Background and Literature Review

Wildfires are a threat to the environment and human life, especially in regions like California that are prone to dry conditions, high winds, and dense vegetation. Understanding the cause behind these wildfires is crucial for the prevention, early intervention, and resource allocation. Wildfire can vastly be categorized as either human caused such as equipment and vehicle use, arson/incendiarism, debris and open burning or smoking or naturally caused, through lighting strikes. [7]

Recent advances in Machine Learning have opened new ways for predictive analytics in environmental sciences, including wildfire detection and forecasting. Prior research has focused primarily on predicting wildfire risk, spread, and intensity using vegetation, weather conditions, and topographic features [3], [8]. However, few studies have addressed the classification of wildfire causes, which is an important aspect for target policy interventions and law enforcement. [9]

Several datasets have been used in previous work, including satellite imagery (e.g., MODIS, VIIRS), weather station data (e.g., NOAA), and fire incident databases (e.g. FRAP). Researchers have applied classification models such as Decision Trees, Support Vector Machines, and Random Forest to model the wildfire behavior [10], [11]. The accuracy of these models often depends on the high-quality input features, robust preprocessing, and effective handling of missing or inconsistent data.

This project is built upon that foundation by developing a machine learning model capable of predicting the cause of wildfire (human or natural) using real-world data from California. The goal of this project is to improve classification accuracy using carefully engineered features derived from both the fire incidents and weather conditions [11].

1.2 Motivation:

Humans are responsible for approximately 85% of wildfires, whether through negligence, arson, or any other activities, such as open burning or vehicle use.[7] Given that many fires are preventable, identifying the cause of wildfire accurately is vital for targeting prevention efforts, improving response strategies, and reducing overall fire risk. [4] Most of the fires are caused by human activity and some by nature such as lightning. Finding the root cause of how these fires ignite what features contribute for their ignition, depending upon the features can we decide whether it was human caused, or nature wildfire is what this study focuses on.

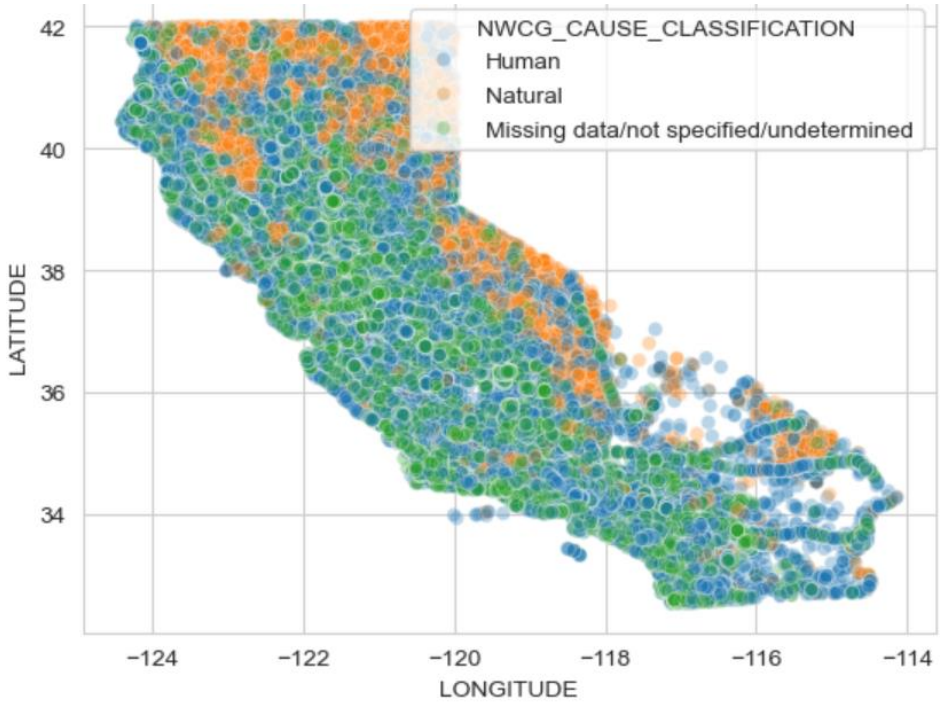

Figure 1

Distribution of wildfire causes across California is shown in fig. 1, based on NWCG cause classification. Majority of wildfires, approximately 76.6% are human caused, followed by 16.1% missing or undetermined causes and 7.3% natural wildfires.

California, with its vast landscape and environmental conditions, is vulnerable to wildfire. Human-caused fires present significant challenges for mitigation, as they often occur at high-risk areas where intervention could make vast difference. On the other hand, natural causes, such as lightning strikes, require different approaches, such as monitoring vulnerable regions and implementing early warning systems. [3]

By developing a trustworthy method to classify the cause of wildfires, this project aims to fill a critical gap in fire management practices. Correct classification will enable fire agencies to allocate resources, plan for prevention, and implement policies tailored to either human or naturally caused fires. With climate change contributing to increase in fire frequency and intensity, understanding these causes is more urgent than ever, allowing for more informed decisions in fire prevention. [14]

This project uses historical data on fire incidents and weather conditions to create a classification model, with the goal of improving the strategies of fire management, reducing the social, environmental, and economic impacts of wildfires. [16]

1.3 Existing Dataset

This section outlines the datasets utilized in this project, focusing on the fire incident data from the Fire Return Assessment Program (FRAP) and meteorological data from the National Oceanic and Atmospheric Administration (NOAA) [1][2] [14].

1.3.1 Fire Return Assessment Program (FRAP) Data

FRAP dataset is provided by the California Department of Forestry and Fire Protection (CAL FIRE) and contains historical wildfire incident data. While the original data spans across the United States, it was filtered specifically to consider only wildfire records from California for the purpose of this project. This ensures focused analysis relevant to the region of interest.

The dataset includes key information about the date of each incident, geographical coordinates, the reported cause of fire (classified as human-caused or natural), time the fire took place, and the size of fire. Covering the period from January 1, 1992, to December 31, 2020, these features enable detailed analysis of wildfire trends, spatial distribution, and underlying causes specific to the state of California over an extended period [1].

1.3.2 National Oceanic and Atmospheric Administration (NOAA) Data

Meteorological information was sourced from NOAA, providing data from multiple weather stations distributed across California. It includes environmental variables such as temperature (MAX, MIN), humidity, wind speed, and precipitation, all of which are important indicators of wildfire risk. These attributes help in assessing the environmental conditions present during wildfire incidents and are particularly useful in recognizing natural causes along with the ignition of human caused fires [2][14].

To integrate this data with FRAP wildfire data, spatial matching approach was used. Since NOAA data is associated with specific weather station locations, while the FRAP data contains geographic coordinates of fire incidents, the Haversine distance method was used to merge fire locations with their nearest weather station data.

The NOAA weather stations with highest number of wildfires were considered, this study involved: Merced Municipal Airport, Riverside Municipal Airport, Fresno Yosemite International, Los Angeles International Airport, San Diego International Airport.

This integration enriched the fired incident data with local weather conditions, making the combined dataset suitable for developing an effective wildfire cause classification model.

1.3.3 Relevance to the Study

Both datasets the FRAP and NOAA, are directly related to the goals of this study, which focuses on classifying the causes of wildfires as either human-caused or natural [1] [21]. The FRAP dataset provides a detailed comprehensive record of fire incidents, making it a necessary source for understanding wildfire distribution and cause. The NOAA dataset contributes in

meteorological data, which is important for understanding the environmental conditions that might lead to natural wildfires, such as those triggered by lightning.

Integrating weather data with fire incident data allows for a more complete model of wildfire causes, thereby improving the accuracy of the classification. It allows the machine learning model to make predictions based on both human and environmental factors, facilitating better wildfire prevention and resource allocation [20] [21].

Chapter 2: Methods

2.1 Tools and Libraries

For this study, the entire workflow was conducted using Python within Jupyter lab environment. This setup facilitated interactive data exploration, essential preprocessing, and integration of essential libraries for both machine learning and data visualization tasks. Key libraries utilized were:

- Pandas: For data manipulation and analysis.
- geopandas: For geospatial data preprocessing.
- Matplotlib and seaborn: For data visualization and exploratory analysis.
- Scikit-learn: For machine learning model development and evaluation.

These tools and libraries collectively supported the entire data pipeline, from initial data loading to model training.

2.2 Data Collection and Initial Preprocessing

2.2.1 FRAP Dataset

The FRAP dataset, by California Department of Forestry and Fire Protection (CAL FIRE), provides detailed records of wildfire incidents in California. For this project, the dataset included only state of California incidents. Key attributes for this analysis included:

- DISCOVERY_DATE: Date on which the fire was first reported.
- DISCOVERY_TIME: Time of day at which the fire was discovered.
- FIRE_YEAR: Year in which the fire occurred.
- LATITUDE: Geographic latitude of the fire's origin.
- LONGITUDE: Geographic longitude of the fire's origin.
- STATE: State in which the fire was recorded (filtered to California).
- FIRE_SIZE: Size of the fire in acres.
- FIRE_SIZE_CLASS: Classification of fire size.
- NWCG_CAUSE_CLASSIFICATION: Specific cause classification of the fire.
- NWCG_GENERAL_CAUSE: General cause of the fire (human or natural).
- COUNTY: Name of the county in which the fire occurred.

Data preprocessing for this dataset began with exploratory data analysis (EDA) using python libraries such as pandas and geopandas. Dimensions of dataset were examined with functions like .shape, and columns were checked with .column method. Initially at the beginning of exploration there were (2303566, 38) rows and columns but after filtering the data to just the California statement the dataset was cut down to (251881 rows x 3 columns). After carrying out a missing value analysis, total of 94,725 values from the County column were found missing. To solve this, a shapefile of California county boundaries was used, and a spatial join was performed using the method .sjoin() from geopandas package. This approach of geospatial join helped impute the missing values based on each incident's latitude and longitude.

Temporal fields such as DISCOVERY_DATE and DISCOVERY_TIME were transformed from Julian to standard datetime formats usings the .to_datetime() method from pandas library. This enabled the extraction of temporal features like week, month, and time duration between each fire discovery.

To streamline modeling, a subset of selected features was used in a new GeoDataFrame. Missing values were identified using .isnull().sum() again after the imputation of COUNTY columns.

Statistical summary of key numerical features is provided below in Table 1. Observations, FIRE_SIZE column is highly skewed, with a maximum of 589,368 acres and a mean of only 83.12 acres, indicating extreme outliers.

Table 1: Summary Statistics for Numerical Features

Feature	Mean	Std Dev	Min	25%	50%	75%	Max
FIRE_YEAR	2006.01	8.30	1992	1999	2006	2013	2020
LATITUDE	37.25	2.56	32.54	34.66	37.39	39.23	42.01
LONGITUDE	-120.12	2.13	-124.40	-121.68	-120.49	-118.35	-114.14
FIRE_SIZE	83.12	3040.67	0.001	0.10	0.20	1.00	589,368.0

Table 2: Summary Statistics for Categorical Features

Feature	Unique Values	Most Frequent Value	Frequency
STATE	1	CA (California)	251,678
FIRE_SIZE_CLASS	7	A (Smallest fires: < 0.25 acres)	138,375
NWCG_CAUSE_CLASSIFICATION	3	Human	183,055
NWCG_GENERAL_CAUSE	13	Missing data/not specified/undetermined	95,224

Dataset shows class imbalance in both fire cause and fire size. The large number of small fires and presence of extreme fire sizes calls for the need of transformations (e.g., logarithmic scaling, boxcox) in the future model processing steps. Additionally, the high proportion of missing_unspecified_causes in NWCG_GENERAL_CAUSE must be carefully addressed during feature engineering, either through exclusion, imputation or categorization as a distinct class.

This workflow of preprocessing ensures that the FRAP dataset is complete, consistent, and structured for effective integration with meteorological data and subsequent predictive modeling.

2.2.2 NOAA Dataset

NOAA daily weather dataset provides us temporal features with comprehensive historical weather data from multiple stations across California. Data was collected from the official site of NOAA (National Oceanic and Atmospheric Administration), it was filtered with the top 4 stations showcasing the maximum number of fire incidents and were downloaded for the analysis. The four stations filtered were

- MERCED MUNICIPAL AIRPORT
- RIVERSIDE MUNICIPAL AIRPORT
- FRESNO YOSEMITE INTERNATIONAL
- LOS ANGELES INTERNATIONAL AIRPORT
- SAN DIEGO INTERNATIONAL AIRPORT.

Each column was examined for missing values. Columns with more than 50% missing data were dropped to maintain dataset quality. The remaining columns were retained for further analysis. Columns missing with values below 50% threshold were imputed using median value. This approach is robust to outliers, which are common in weather data, and preserves the central tendency of the data.

After cleaning the data, the following columns were selected for data integration and modeling:

- STATION
- NAME
- LATITUDE
- LONGITUDE
- ELEVATION
- DATE
- AWND - Average Wind Speed (tenths of meters per second)
- PGTM - Peak Gust Time (HHMM)
- PRCP - Precipitation (tenths of mm)
- TMAX - Maximum Daily Temperature (degree Celsius)
- TMIN - Minimum Daily Temperature (degree Celsius)
- WDF2 - Direction of fastest 2-minute Wind (degrees)

- WDF5 - Direction of fastest 5-minute Wind (degrees)
- WSF2 - Speed of fastest 2-minute Wind (tenths of m/s)
- WSF5 - Speed of fastest 5-minute Wind (tenths of m/s)

Descriptive statistics of numerical features is provided in Table 3

Table 3: Summary Statistics for NOAA Dataset, Numerical Features

	LATITUDE	LONGITUDE	ELEVATION	AWND	PGTM	PRCP	TMAX	TMIN	WDF2	WDF5	WSF2	WSF5
count	55999.00000	55999.000000	55999.000000	55999.000000	55999.000000	55999.000000	55999.000000	55999.000000	55999.000000	55999.000000	55999.000000	55999.000000
mean	33.11905	-116.305593	267.830179	6.057037	1454.550046	0.035001	74.183646	52.840765	260.286612	258.936017	15.669105	19.616481
std	5.26842	6.669558	504.151440	2.417765	310.194704	0.161488	12.231321	9.999122	64.181458	65.392087	18.614000	7.455681
min	18.60000	-120.517880	4.600000	0.000000	0.000000	0.000000	34.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	32.73360	-119.720160	29.700000	4.700000	1449.000000	0.000000	66.000000	46.000000	260.000000	260.000000	13.000000	17.000000
50%	33.95282	-118.386600	46.500000	5.820000	1455.000000	0.000000	72.000000	53.000000	270.000000	270.000000	15.000000	19.000000
75%	36.77999	-117.183100	257.700000	7.160000	1501.000000	0.000000	80.000000	61.000000	300.000000	290.000000	16.100000	21.000000
max	37.28597	-97.266700	1692.900000	23.940000	2359.000000	4.800000	118.000000	90.000000	360.000000	360.000000	916.900000	293.000000

Summary Statistics show a need for outlier handling, identification and capping of extreme values, especially for wind speed and elevation. Label or one-hot encoding to be performed on categorical data, numerical features need to be converted into standard scalar for better modeling.

NOAA weather dataset, after thorough cleaning and preprocessing, is complete and ready to use for merging with California FRAP wildfire dataset. Its inclusion enhances the temporal and environmental granularity of the wildfires model processing, enabling them to provide accurate classification and risk assessment of wildfire causes.

2.3 Data Normalization

To ensure comparability among the features and to use effective model training, all numerical variables were normalized. The target variable was encoded using label encoder for better readability during model training. Human cause was encoded as 0 and Natural cause was encoded as 1. Standardization was performed using z-scores method, resulting in standard normal distribution of the numerical features with mean one and standard deviation one. For variables exhibiting significant skewness, such as

- FIRE_SIZE
- PRCP

Log and Box-Cox transformations were applied to reduce skewness and approximate normality. The normalization process is an important part for improving the performance and stability of

machine learning algorithms, most importantly for those sensitive to feature scales and distribution.

2.4 Data Exploration

Exploratory data analysis was conducted to understand feature distributions, relationships, and data quality. Summary statistics, including mean, median, and skewness, were computed for all numerical variables. Correlation matrix analysis was performed to identify linear relationships between features and target variable, resulting in features such as LATITUDE, TMIN, and cyclical encodings of MONTH to be strongly correlated with the wildfire causes. Visualizations, including histograms, boxplots, and heatmaps, were examined to determine the effect of normalization and transformation steps, as well as to detect outlier and guide into feature select.

2.5 Model Selection and Justification

In this study, four machine learning algorithms were used for the task of wildfire cause classification: Logistic Regression, Light Gradient Boosting Machine (Light GBM), Random Forest, and XGBoost. These models were chosen to represent a range of linear ad ensemble-based approaches, each with unique strengths in handling structured data and complex features relation.

- **Logistic Regression** served as a baseline linear model, prividng interpretability and a benchmark for comparison with more complex algorithms.
- **Random Forest and XGBoost (Extreme Gradient Boosting)** are ensemble tree-based models known for their ability to capture nonlinear relationships and interactions among features.
- **LightGBM (Light Gradient Boosting Machine)** was included for its computational efficiency and high predictive performance along with the speed of executing with large dataset particularly.

Initial model selection was followed by cross validation performance on training set and the dataset was examined with the primary evaluation metric like Accuracy, Recall, Precision, f1-score.

Confusion Matrix:

	PREDICTED POSITIVE	**PREDICTED NEGATIVE**
ACTUAL POSITIVE	True Positive (TP)	False Negative (FN)
ACTUAL NEGATIVE	False Positive (FP)	True Negative (TN)

Accuracy: Measures the proportion of correctly classified instances among all.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision: Measures, how many of the predicted positive cases were actual positive.

$$Precision = \frac{TP}{TP + FP}$$

Recall (Sensitivity or True Positive Rate): Measures how many of the actual positives were correctly predicted.

$$Recall = \frac{TP}{TP + FN}$$

F1-Score: Harmonic mean of precision and recall, balancing the two.

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Specificity (True Negative Rate): Measures how many of the actual negative cases were correctly predicted.

$$Specificity = \frac{TN}{TN + FP}$$

Automatic feature selection was incorporated within the cross-validation pipeline to identify the most informative subset of predictors and to prevent overfitting. This approach made sure that the performance of final model was both accurate and generalizable.

2.6 Model Training Workflow

The model training workflow was made to maximize predictive accuracy and ensure detailed evaluation. The process began by training the baseline model logistic regression, followed by the iterative training for LightGBM, Random Forest and XGBoost classifiers. For each model, stratified k-fold cross validation was used to provide reliable results of performance and to guide hyper parameters tuning.

Automatic feature selection was performed during cross validation within each fold to optimize the set of input variables for each algorithm.

After complete evaluation, the Random Forest Classifier emerged as the best performing model based on cross-validation accuracy and performance metric. Hyperparameter tuning was conducted using RandomizedSeacrhCV, optimizing the parameters such as number of tress, maximum depth,and minimum samples per split.

The final, tuned Random Forest model was retrained on full training set and was deployed as a RESTful API using FLASK, enabiling real-time inference and integration downstream applications.

Chapter 3: Implementation

3.1 Data Merging (NOAA and FRAP Datasets)

To carry out analysis of the relationship between weather conditions and wildfire occurrences, the NOAA dataset was merged with California wildfire records from the FRAP dataset. This combination allows for a more complete assessment of cause of wildfire risk.

The keys columns in merging include:

- LATITUDE
- LONGITUDE
- DATE

Geospatial Matching was used to merge these features, the nearest weather station to each wildfire record was identified using the BALL TREE Algorithm from sklearn.neighbors, using the Haversine distance metric. This approach accurately calculates distances on the Earth's surface, accounting for its spherical geometry. Coordinates were converted to radians to ensure compatibility with the Haversine formula [14]

The **Haversine Formula** is used in mathematical equation to calculate the great-circle distance between two points on the surface of a sphere, given the latitude and longitude coordinates.

$$d = 2r * \arcsin(\sqrt{sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos(\varphi_1) * \cos(\varphi_2) * sin^2\left(\frac{\Delta\lambda}{2}\right)})$$

Where, d = distance between the two points (in kilometers or meters, depending on the radius used)

r = radius of the Earth (mean radius ≈ 6,371km)

φ_1, φ_2 = latitudes of the two points in radians

λ_1, λ_2= longitudes of the two points in radians

$\Delta\varphi$ = difference in latitude

$\Delta\lambda$= difference in longitude

The DISCOVERY_DATE of each wildfire was matched to the corresponding weather data for the same day. Both data fields were standardized to a common (date-only) format to ensure accurate alignment of data.

After the merger was done, missing values were identified in weather-related columns. The cause was after carrying out the merging, some fire records had no NOAA station within the specified

radius. Hence, no rows with missing weather data were dropped so that the dataset only contains records with both fire and weather information.

3.2 Exploratory Data Analysis

3.2.1 Target and Feature Variable Selection

The main objective of this project is to predict the cause of wildfire using a combination of fire incident, geographical, and weather-related features. The **target variable** (dependent variable) selected for classification is NWCG_GENERAL_CAUSE, which provides categorization of human, missing and natural causes.

The independent variables (features) were grouped as follows:

- Temporal Features: YEAR, MONTH, DAY, HOUR.
- Geographical Features: LATITUDE, LONGITUDE.
- Fire Behavior Features: FIRE_SIZE.
- Weather Features: AWND, PGTM, TMAX, TMIN, WDF2, WDF5, WSF2, WSF5.

3.2.2 Key observations and Data Quality

Summary statistics of the final data frame showed key observations. Most environmental variables (e.g., precipitation, wind speed) were skewed toward lower b=values, reflecting typical dry, calm wildfire conditions. Several features contain outliers (e.g., extremely high wind speeds), requiring further cleaning. The distribution of FIRE_SIZE is highly skewed, necessitating log transformation or a need for normalization. High rates of missing or unspecified values in certain Target column must be addressed.

3.2.3 Feature Selecting and Processing

The final set of features were selected based on their relevance for modeling. These features were selected based on the exploratory data analysis, showing the correlations between the target variable and independent variables. Going forward we will need to engineer columns for better modeling.

Non- informative fields such as station, names, stat/county, elevation, duplicate coordinates were excluded to reduce noise and overfitting.

3.2.4 Feature Engineering

Features like MONTH and HOUR were encoded using sine and cosine transformations to capture cyclical nature, improving the model's ability to learn time-based patterns. The target variable NWCG_CAUSE_CLASSIFICATION was label encoded for binary classification tasks. All numerical features were standardized using StandardScalar to ensure a mean of 0 and standard deviation of 1.

3.2.5 Handling Missing Cause

Rows with missing or unspecified cause classification were removed to ensure a clean target variable for binary classification. For weather features, missing values were imputed using median, which is robust to outliers.

3.3 Data Visualization

3.3.1 Correlation Heat Map

A **correlation matrix** was computed to assess the linear relationship between numerical features and target variable, NWCG_CAUSE_CLASSIFICATION, where 0=Human and 1=Natural. Both LATITUDE and TMIN exhibited the strongest positive correlations with cause classification about 0.27, suggesting that geographic locations and temperature play a significant role in distinguishing human-caused and natural wildfires. TMAX = 0.21 and AWND = 0.08 had positive correlations with the target variable, suggesting that higher temperatures and wind speeds could be associated in predicting the cause. On the other hand features like MONTH_sin and MONTH_cos were negatively correlated with the target, impacting the seasonality part on wildfire ignition sources.

Further examinations of feature relations show strong internal connections with weather and spatial data. TMAX and TMIN were highly correlated (0.71) and WSF2 and WSF5 (0.72), reflecting the expected consistency between these features.

These findings show that LATITUDE, TONGITUDE, TMAX and cyclical encodings for month are more likely valuable predictors, while other features such as PRCP, FIRE_SIZE and HOUR_cos, despite weak correlations with target, might contribute to capturing the non linear pattern in advanced models.

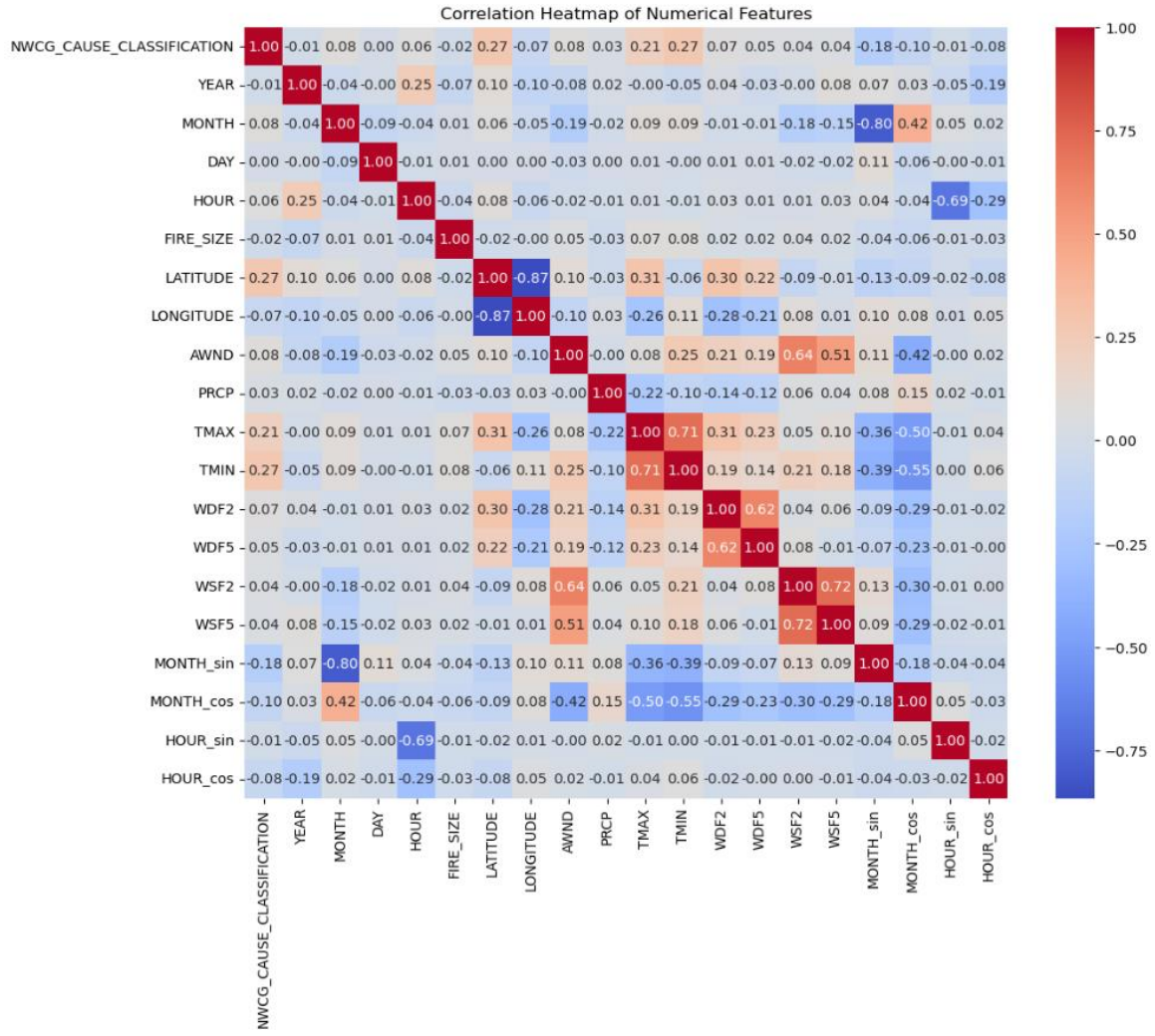

Figure 2

Correlation heatmap in fig. 2 showcasing the relationship between the wildfire cause, which is the target variable and different input features.

3.3.2 Handling Outliers

IQR test was conducted to check the outliers in the dataset after the test around 60979 values were outliers. Capping was used to the extreme values at 1st and 99th percentiles to reduce influence without removing data points entirely. After capping the number of outliers significantly decreased in the dataset it went down to about 27480.

3.3.3 Skewness and Feature Transformation

Summary statistics along with KDE plots showed features like FIRE_SIZE (7.470407), PRCP to have high skewness (20.165), along with WSF2 (38.751) and WSF5 (7.071). Log transformation was applied to features that still showed high skewness. After log transformation the skewness of these features went down to about

- FIRE_SIZE = 7.394864
- PRCP = 8.947162
- WSF2 = 1.402297
- WSF5 = -0.125461

FIRE_SIZE and PRCP still show high skewness.

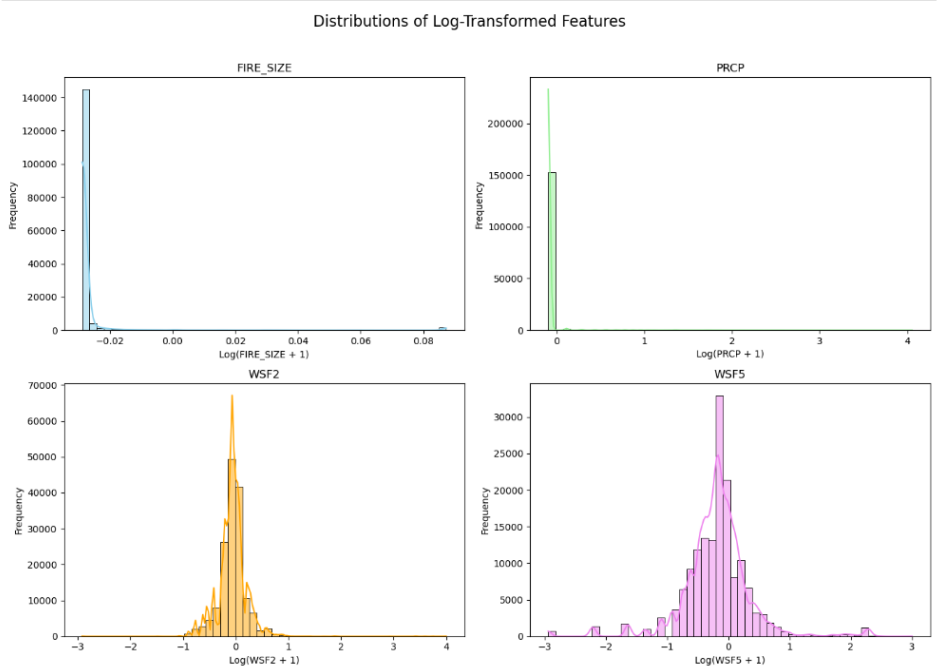

Figure 3

Skewness, KDE Plot of log transformed attributes in fig. 3, showing high positive skewness in two features FIRE_SIZE and PRCP - precipitation

To handle this skewness Box-Cox transformation was applied. This method dynamically selects the optimal power transformation for each feature.

The Box-Cox transformation:

$$y(\lambda) = \begin{cases} \dfrac{y^{\lambda} - 1}{\lambda}, & if\ \lambda \neq 0 \\ \ln(y), & if\ \lambda \neq 0 \end{cases}$$

Where y is the original FIRE_SIZE, PRCP value, and λ is the transformation parameter determined from the data.The below figure shows the skewness after box-cox transformation, the skewness of FIRE_SIZE became perfect symmetrical with 0.000 skewness score, but PRCP skewness was still right skewed with values of 5.38.

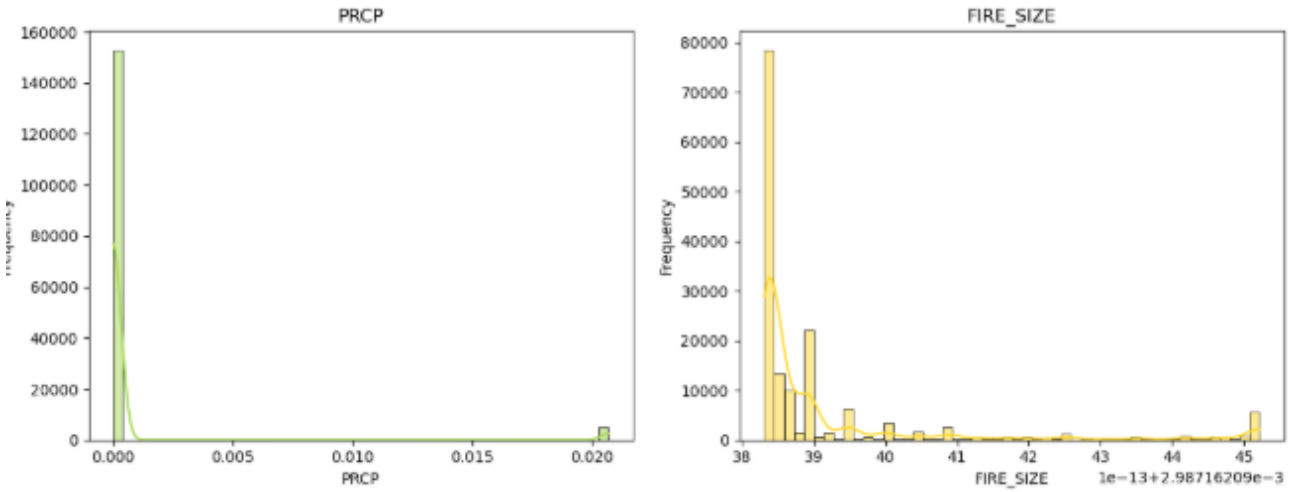

Figure 4

KDE plot in fig. 4, showcasing the skewness after applying Box-Cox transformation, the attribute FIRE_SIZE has perfect symmetric skewness, but Precipitation still struggles.

After transformation was done on specific features, some features (e.g., WSF5) exhibited missing values due to invalid log operations. These issues were addressed by removing the affected rows. Rows with missing or unspecified cause classification were removed to ensure clean binary classification dataset.

The distribution of NWCG_CAUSE_CLASSIFICATION variable showed significant class imbalance.

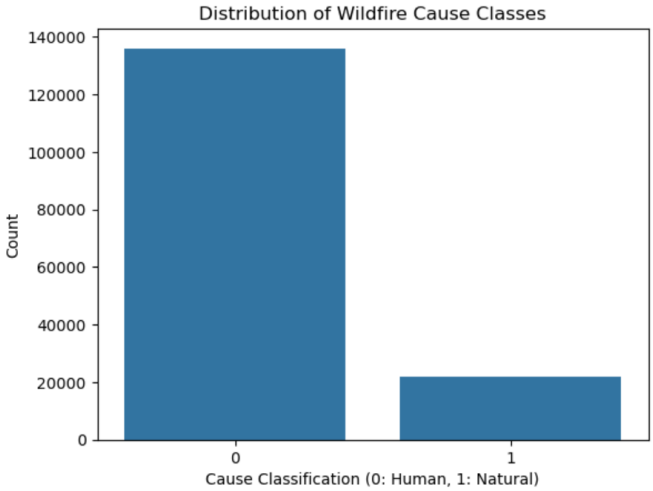

Figure 5

Distribution of Target variable shown in fig. 5, addressing class imbalance isse with Class 0 (Human Caused) having 135,995 instances and class 1 (Natural Wildfire)having 21,809 instances

This imbalance indicates that the fires caused by humans dominate the dataset, which could bias the model toward predicting the majority class. Techniques such as resampling were used before model training to resolve this issue.

The exploratory data analysis revealed important relationships and distributional properties in the dataset, guiding the selection and transformation of features for modeling. The use of correlation analysis, skewness reduction and robust feature engineering ensures that the dataset is well-prepared for predictive modeling of wildfire causes.

3.4 Model Training:

3.4.1 Data Splitting and Class Imbalance Handling

For models evaluation and addressing the issue of class imbalance in the target variable, the dataset was first split into training and testing subsets using 80/20 split ratio, with 20% of the data stored for final model evaluation. Stratified sampling was used to preserve the original distribution of target variable, NWCG_CAUSE_CLASSIFICATION, across both distributions.

Following the split, the training data set had a significant class imbalance, with human-caused wildfires having 108,503 samples, outnumbering the natural wildfire which had about 17,422 samples. To solve this imbalance and improve the model's ability to learn from both classes, the Synthetic Minority Over-Sampling Technique was applied to the training data. SMOTE generates synthetic instances of the minority class by interpolating between existing class samples, resulting in balanced training with equal samples in both classes.

3.4.2 Baseline Model

A baseline model was set up to provide a performance benchmark against evaluating the performance of more advanced machine learning models before applying complex models and feature selection techniques.

Logistic Regression Classifier was selected as a baseline model due to its simplicity and efficiency in binary classification. The model was trained using inbuild logistic regression function with default hyperparameters, without any feature selection aside from standard scalar and handling missing values.

Logistic regression served benchmark while Random Forest, LightGBM, and XGBOOST provided ensemble-based approaches capable of capturing complex, non-linear relationships in data.

The features used to train this baseline model:

Temporal	YEAR, MONTH, DAY, HOUR, MONTH_sin, MONTH_cos, HOUR_sin, HOUR_cos
Spatial	LATITUDE, LONGITUDE
Fire Characteristics	FIRE_SIZE
Meteorological Data	AWND, PRCP, TMAX, TMIN, WDF2, WDF5, WSF2, WSF5

The model's performance was evaluated with performance metrics to assess predictive performance on both classes. This baseline model established the minimum standard for future ensemble models.

3.4.3 Cross-Validation and Feature Engineering

To evaluate the models performance, Stratified K-Fold Cross-Validation with five folds was used. This step was chosen to store the proportion of classes (human caused , natural caused) in each fold for validation, making sure to keep balanced representation during bot training and validation process.

Each model Pipeline was composed of two main components:

1. **Feature Section** using SelectKBest with mutual information classification to get 10 most informative features.
2. The **classifier**, which included Random Forest, Light GBM, XGBoost, and Logistic Regression.

These pipelines were applied to resample training data, which had undergone SMOTE for solving the class imbalance problem. For each model, cross-validation was performed by training and validating across five folds i.e. five different data splits, with the accuracy metric used to assess performance in each fold. Mean accuracy and standard deviation were calculated across folds to summarize the performance of model.

After cross-validation, each pipeline was trained again on the full resampled training set and evaluated on unseen test set. This helped with the comparison of both validation and real-world test performance, ensuring consistency between cross validation results that generalized well with new data.

3.4.4 Hyperparameter Tuning

After carrying out cross validation along with feature selection, hyperparameter tuning using RandomizedSearchCV, an efficient method for searching across predefined parameter grid was done on the best model. Here we selected Random Forest to be our best model.

Initially gridcv was used but because of its complexity and memory usage of Random model along with high time complexity made it not a good choice for model tuning.

Hence, RandomizedSearchCV was used which balances the optimization quality and computational cost by randomly sampling a fixed number of parameters.

A pipeline was constructed which consisted of feature selection with SelectKBest (using mutual information criterion) and the Random Forest Classifier. The RandomizedSearchCV included hyperparameters like splitting a node, minimum samples at leaf nodes, and the use of bootstrapping.

Five-fold cross validation was applied during the randomized search to make sure that model evaluation was efficient across different data subsets and to prevent overfitting.

After the hyperparameter tuning was done, the best hyperparameter combination was selected based on cross-validation accuracy.

The optimized model was then trained again on the full resampled training dataset and was evaluated in test set.

The trained model was then saved using model serialization techniques, enabling future reuse or integration into applications without retraining the entire model.

Chapter 4: Results and Analysis

4.1 Baseline Model Results

4.1.1. Logistic Regression:

The performance of baseline model used before applying advanced machine learning models and techniques such as cross-validation, hyper-parameter tuning. In this project, the baseline model used was Logistic Regression classifier using default settings.

Performance Metrics:

Model	Precision	Recall	F1-score	Support
0 (Human)	0.97	0.83	0.89	27127
1 (Natural)	0.43	0.82	0.57	4355
Accuracy			0.83	0.83
Macro avg	0.70	0.82	0.73	31482
Weighted avg	0.89	0.83	0.85	31482

Table 4: Showing performance metric of Baseline Model (Logistic regression)

The baseline model's performance, in the above table, shows an imbalance in predictive ability across the two classes. While it achieved good accuracy of 83%, model was more effective at predicting human-caused wildfires. This is visible from the high precision and f1-score for class 0, compared to much lower precision and f1-score for class 1. The recall for class 1 was strong at

0.82, suggesting the model correctly identified most human-caused wildfires, but low precision shows frequent false positives. This is the case because of data imbalances. Overall, the baseline model sets a reasonable benchmark.

Confusion Matrix:

The fig below shows the confusion matrix of baseline model, just considering the accuracy in a dataset which has major imbalance is not good practice. Evaluating the performance metric along with confusion table gives us a clear view of how exactly the model works.

The model performs better at identifying human-caused wildfires, with a high number of true negatives. However, it struggles with natural wildfires, making false positive predictions (4673), indicating lower precision for class1.

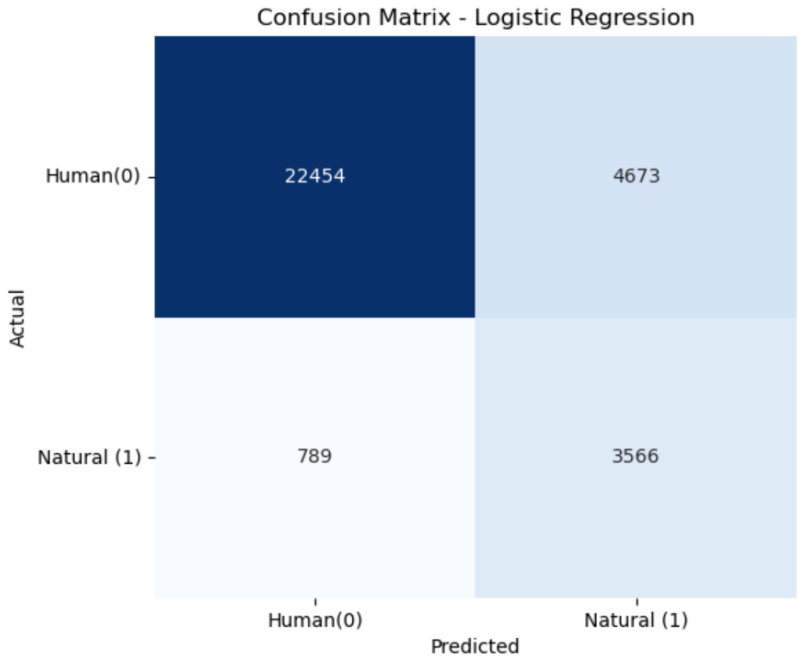

Figure 6

Confusion Matrix of Baseline Model (Logistic Regression)

4.1.2 Ensemble Model Performance before optimization:

Following the training of baseline Logistic Regression Model, more advanced ensemble learning algorithms were used to evaluate improvement in classification. The ensemble models selected for this purpose included:

- Random Forest Classifier
- Light Gradient Boosting (LightGBM)
- Extreme Gradient Boosting (XGBoost)

These models were chosen for their advanced properties in handling complex , non-linear relationships and for overcoming overfitting. After training each models performance was evaluated using key classification metrics, allowing for comparison with baseline model.

Performance Metric:

Model	Accuracy	Precision (Class 0)	Precision (Class 1)	Recall (Class 0)	Recall (Class 1)	F1 Score (Class 0)	F1 Score (Class 1)
Random Forest	0.95	0.97	0.81	0.97	0.83	0.97	0.82
LightGBM	0.91	0.97	0.62	0.92	0.84	0.94	0.72
XGBoost	0.90	0.98	0.60	0.91	0.87	0.94	0.71

Table 5: Showing Ensemble model performance and choosing the best performing model.

In the above table, among the ensemble models, Random Forest performed best with accuracy of 95% and strong F1-scores for both classes. LightGBM and XGBoost followed with a little lower accuracy of 91% and 90% and weaker precision for the minority class, but these models recall was high. Over, ensemble model outperformed the baseline model.

4.2 Cross Validation Results:

After training baseline and ensemble models without any tuning, cross validation was performed alongside feature engineering using SelectKBest to select the most informative features. Cross validation was applied to prevent overfitting and for generalization, the final model performance was evaluated on unseen data.

Here are the resulting Performance metrics:

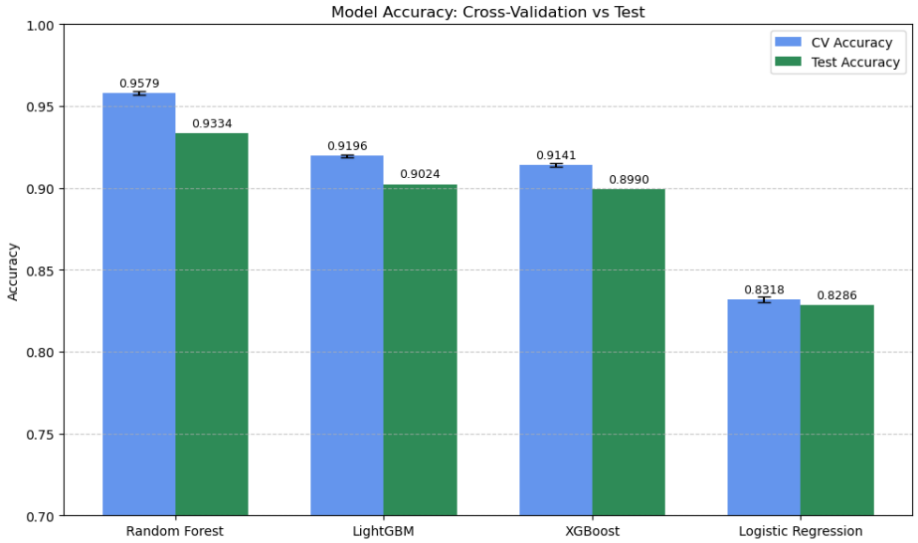

Figure 6

Fig. 6 shows the comparison of cv and test accuracy, Random forest achieved the highest cross-validation of 95% and testing accuracy 93.3% showing strong results.

Performance Metric:

Model	Accuracy	Precision (Class 0)	Precision (Class 1)	Recall (Class 0)	Recall (Class 1)	F1 Score (Class 0)	F1 Score (Class 1)
Random Forest	0.93	0.97	0.71	0.94	0.85	0.96	0.77
LightGBM	0.90	0.97	0.60	0.91	0.84	0.94	0.70
XGBoost	0.89	0.97	0.59	0.90	0.85	0.93	0.70
Logistic Regression	0.82	0.96	0.43	0.83	0.80	0.89	0.56

Table 6: Performance metric showing precision, recall, accuracy and comparing the models

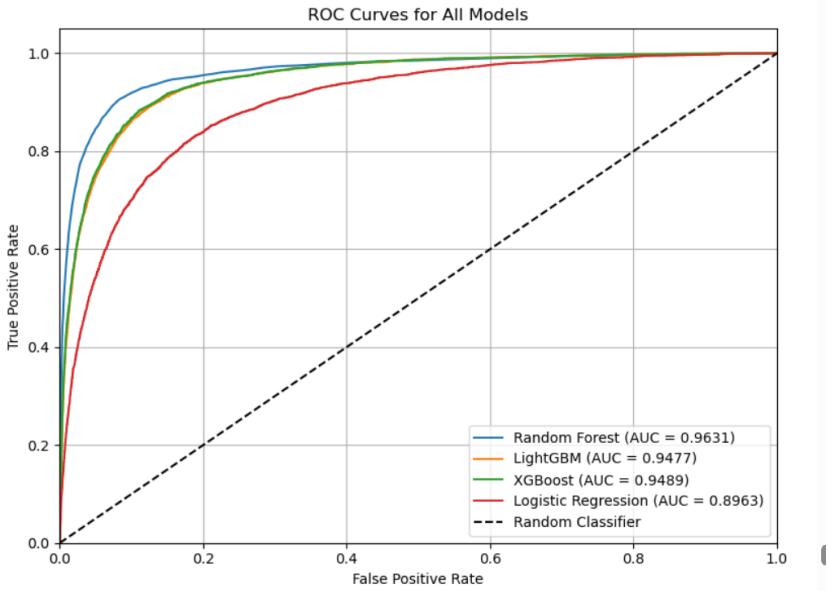

Figure 7

ROC- AUC CURVE fig. 6, showing the classification performance metric to have better understanding of model. Random forest being the best performed model

The performance metric shows that Random Forest Model outperformed all the others, achieving the highest accuracy (93.3%) and strong, balanced precision and recall for both human and natural causes. LightGBM and XGBoost also performed well, with good recall for natural wildfires but lower precision, indicating more false positives.

4.3 Hyperparameter Tuning Results and Analysis of Final Model

Hyperparameter tuning significantly improved the Random Forest models performance by advancing decision tree behavior. Setting the parameters as min_sample_split = 5, disabling bootstrap helped clearer splits, using n_estimators = 100 balanced the performance and computational accuracy.

Random forest was the only model that was tuned because of its best performance in the previously trained models with the performance metric being the strong indicator of the model performing well.

Performance metric:

Model	Precision	Recall	F1-score	Support
0 (Human)	0.97	0.96	0.97	27127
1 (Natural)	0.76	0.84	0.80	4355
Accuracy			0.94	0.83
Macro avg	0.87	0.90	0.88	31482
Weighted avg	0.95	0.94	0.94	31482

Table 7: Performance matrix of Final Model- Random Forest Classifier

After hyperparameter tuning, the model showed strong performance with accuracy upto 94%. It maintained high precision and recall for most of the class, Human, with f1-score of 0.97, while also improving detection of minority class, Natural, achieving a precision of 0.76, recall of 0.84, and f1 score of 0.80. The balanced macro and weighted averages indicate the model's ability to generalize well for both classes.

Feature Importance of Final Model:

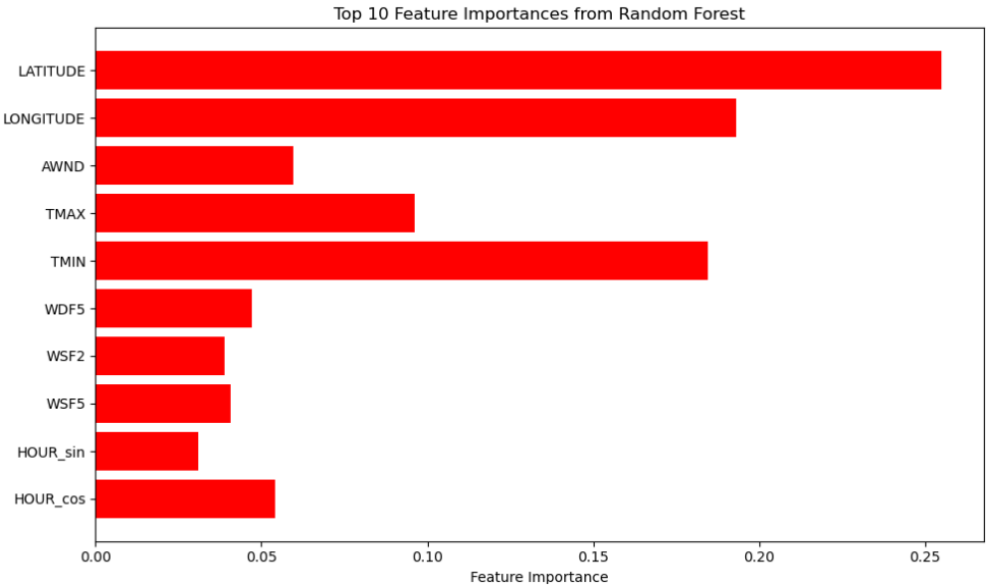

Figure 8

Feature importance is shown in fig. 8, extracted automatically by selectKBest and used for hyperparameter tuning

These were the features that were used for hyperparameter tuning model, it was done automatically using SelectKBest with mutual information classifier, getting the 10 best feature and training the model on it for better classification.

4.4 Comparison of Models:

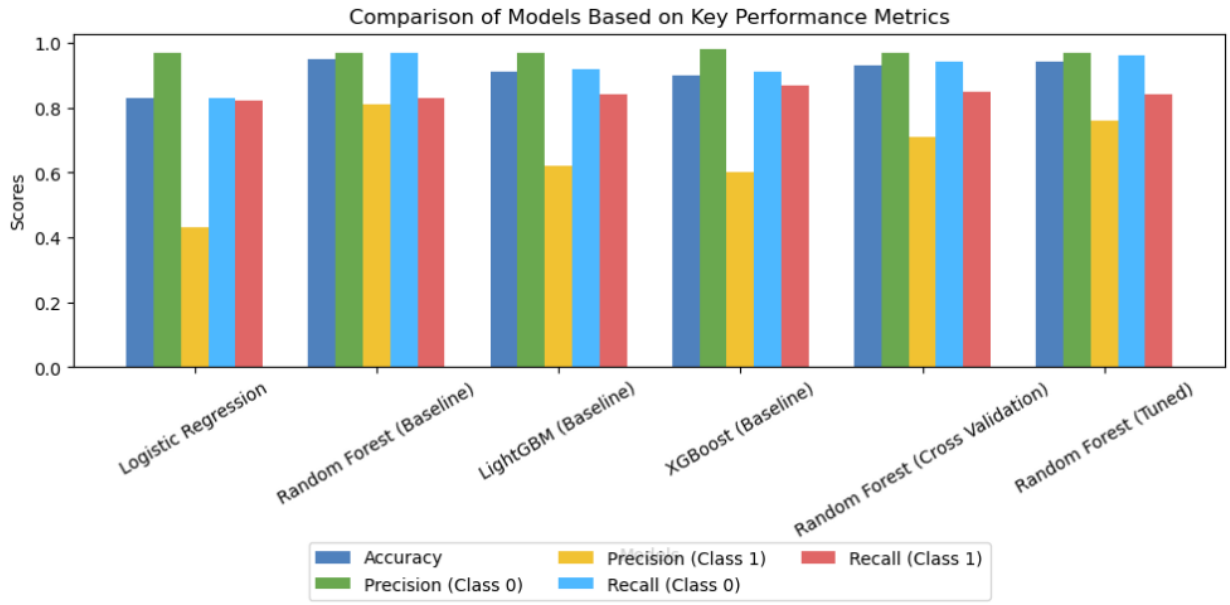

Figure 9

Fig. 9 is a visualization of the comparison between all the models trained and why Random forest was picked as the best model.

The above figure shows the clear performance from the baseline model to advanced ensemble problems. The baseline Logistic Regression, while achieving an accuracy of 83%, showed major imbalance in predicting minority class (Natural), with low precision and F1- Score. Ensemble models like Random Forest, LightGBM, and XGBoost significantly outperformed baseline, with Random Forest showing best accuracy, precision, recall and F1-score. After cross validation and feature selection, Random Forest maintained the high accuracy and improved minority class detection. Finally, after hyperparameter tuning the random forest model performed better, yielding a balanced performance with 94% accuracy and got better class wise metrics, making the most reliable and efficient model for classifying.

Chapter 5: Model Deployment

The final model which was used for deployment was the Random Forest Classiifer obtained after hyperparameter tuning. The model was part of a pipeline that first applied feature selection and the RadomizedSearchCV. With tuned parameters (bootstrap=False, min_samples_split-5, random_state=42), the model showed the best overall performance across all performance metrics. The pipeline was then saved into 'best_rf_model_fm.pkl' and used for final deployment.

Once the optimization of the Random Forest was done using hyperparameter tuning, the final model was deployed using Flask for a real-time prediction. The models efficiency in classification and accuracy on unseen data made sure that it would function effectively in a production environment. Flask was chosen as the deployment framework due to its simplicity and scalability, making it an ideal solution for training machine learning models in real-world applications. The deployment process involved exposing an API endpoint, where users could submit data and receive predictions on the cause of wildfires, whether human or natural.

Chapter 6: Conclusion

This project was focused on predicting the cause of California Wildfires using different machine learning models. After predicting the results of several algorithms, including the baseline model logistic regression and ensemble models like Random Forest, Light Gradient Boosting and XGBoost. Random Forest Classifier was the best performing model. With hyperparameter tuning, it achieved the highest testing accuracy of 94% along with balanced performance metric for both classes of human-caused wildfires and natural wildfires. It did well even when the data was imbalanced. Particularly, the recall for minority class which is natural wildfires showed better improvement, indicating the model's ability to reduce false negative in critical cases which the baseline model and cross validation couldn't resolve.

Chapter 7: Future Scope

- To work more in depth of this project, future work could explore:
- Including wildfire records with missing causes, which was dropped in this project; this can be achieved either by doing imputation or estimating their labels through semi-supervised learning or domain knowledge.
- Examining the use of deep learning models such as neural networks, which may capture more complex non-linear patterns in the data.
- Adding additional spatial-temporal or satellite-based features for rich context.

Chapter 8: References

[1] California Department of Forestry and Fire Protection, "Fire and Resource Assessment Program (FRAP) Wildfire Incident Data," [Online]. Available: https://frap.fire.ca.gov/. [Accessed: May 8, 2025].

[2] NOAA National Centers for Environmental Information, "Global Historical Climatology Network (GHCN) Daily Data," [Online]. Available: https://www.ncdc.noaa.gov/ghcn-daily-description. [Accessed: May 8, 2025].

[3] Wikipedia contributors, "January 2025 Southern California wildfires," *Wikipedia*, May 5, 2025. [Online]. Available: https://en.wikipedia.org/wiki/January_2025_Southern_California_wildfires. [Accessed: May 8, 2025].

[4] California Department of Forestry and Fire Protection, "Statistics - Cal Fire," *CA.gov*, 2025. [Online]. Available: https://www.fire.ca.gov/our-impact/statistics. [Accessed: May 8, 2025].

[5] World Vision, "California fires: Facts, FAQs, and how to help," 2025. [Online]. Available: https://www.worldvision.org/disaster-relief-news-stories/california-fires-facts-faqs-how-to-help. [Accessed: May 8, 2025].

[6] Wikipedia contributors, "2025 California wildfires," *Wikipedia*, Apr. 22, 2025. [Online]. Available: https://en.wikipedia.org/wiki/2025_California_wildfires. [Accessed: May 8, 2025].

[7] U.S. Forest Service, "Inference of wildfire causes from their physical, biological, social and management attributes," 2025. [Online]. Available: https://www.fs.usda.gov/rm/pubs_journals/2025/rmrs_2025_pourmohamad_y001.pdf. [Accessed: May 8, 2025].

[8] ECMWF, "2025 California wildfires: insights from ECMWF forecasts," Feb. 5, 2025. [Online]. Available: https://www.ecmwf.int/en/about/media-centre/science-blog/2025/2025-california-wildfires-insights-ecmwf-forecasts. [Accessed: May 8, 2025].

[9] IRJET, "Wildfire Prediction and Detection using Random Forest and ...," *International Research Journal of Engineering and Technology*, vol. 7, no. 6, pp. 1356–1361, 2020. [Online]. Available: https://www.irjet.net/archives/V7/i6/IRJET-V7I61356.pdf. [Accessed: May 8, 2025].

[10] Boise State University, "Computing student identifies wildfire ignition sources," Feb. 24, 2025. [Online]. Available: https://www.boisestate.edu/news/2025/02/24/machine-learning-sparks-insight-computing-student-identifies-wildfire-ignition-sources/. [Accessed: May 8, 2025].

[11] *Nature*, "Global lightning-ignited wildfires prediction and climate change," Mar. 6, 2025. [Online]. Available: https://www.nature.com/articles/s41598-025-92171-w. [Accessed: May 8, 2025].

[12] California Department of Forestry and Fire Protection, "2025 Fire Season Incident Archive," *CAL FIRE*, 2025. [Online]. Available: https://www.fire.ca.gov/incidents/2025. [Accessed: May 8, 2025].

[13] Bankrate, "2025 U.S. Wildfire Statistics and Facts," Jan. 24, 2025. [Online]. Available: https://www.bankrate.com/insurance/homeowners-insurance/wildfire-statistics/. [Accessed: May 8, 2025].

[14] NOAA Climate.gov, "The weather and climate influences on the January 2025 fires around Los Angeles," Feb. 19, 2025. [Online]. Available: https://content-drupal.climate.gov/news-features/event-tracker/weather-and-climate-influences-january-2025-fires-around-los-angeles. [Accessed: May 8, 2025].

[15] S. G. Shulgina *et al.*, "Exploration of geo-spatial data and machine learning algorithms for wildfire occurrence prediction," *Sci. Rep.*, vol. 15, no. 1, 2025. [Online]. Available: https://www.nature.com/articles/s41598-025-94002-4. [Accessed: May 8, 2025].

[16] The Pennsylvania State University, "Q&A: Causes, spread and solutions for California's wildfire crisis," Jan. 16, 2025. [Online]. Available: https://www.psu.edu/news/research/story/qa-causes-spread-and-solutions-californias-wildfire-crisis. [Accessed: May 8, 2025].

[17] NHSJS, "A Comparative Analysis of Machine Learning Models for Wildfire Prediction," Jun. 22, 2024. [Online]. Available: https://nhsjs.com/2024/a-comparative-analysis-of-machine-learning-models-for-wildfire-prediction/. [Accessed: May 8, 2025].

[18] M. B. Zermani *et al.*, "Predicting wildfires in Algerian forests using machine learning models," *PLoS One*, vol. 18, no. 7, 2023. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC10372657/. [Accessed: May 8, 2025].

[19] CAL FIRE, "Fire and Resource Assessment Program (FRAP) Historical Fire Perimeters," California Department of Forestry and Fire Protection, Apr. 22, 2025. [Online]. Available: https://www.fire.ca.gov/what-we-do/fire-resource-assessment-program/fire-perimeters. [Accessed: May 8, 2025].

[20] Cal OES GIS Data Hub, "USA Current Wildfires - California," 2025. [Online]. Available: https://gis-calema.opendata.arcgis.com/maps/5adf4fafcfdd4cb28a0510f6d9fab122. [Accessed: May 8, 2025].

[21] ESRI, "CAL FIRE Historical Fire Perimeters Available in ArcGIS Living Atlas," Feb. 12, 2024. [Online]. Available: https://www.esri.com/arcgis-blog/products/arcgis-living-atlas/decision-support/cal-fire-historical-fire-perimeters-available-in-arcgis-living-atlas. [Accessed: May 8, 2025].

[22] "haversine - PyPI," 2025. [Online]. Available: https://pypi.org/project/haversine/. [Accessed: May 8, 2025].

[23] N. Rooy, "The Haversine formula," 2016. [Online]. Available: https://nathan.fun/posts/2016-09-07/haversine-with-python/. [Accessed: May 8, 2025].

[24] scikit-learn developers, "haversine_distances - scikit-learn 1.6.1 documentation," 2025. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.haversine_distances.html. [Accessed: May 8, 2025].

[25] scikit-learn developers, "SelectKBest - scikit-learn 1.6.1 documentation," 2025. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html. [Accessed: May 8, 2025].

[26] J. Jordahl *et al.*, "GeoPandas: Python tools for geographic data," 2025. [Online]. Available: https://geopandas.org/. [Accessed: May 8, 2025].

[27] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011. [Online]. Available: https://jmlr.org/papers/v12/pedregosa11a.html. [Accessed: May 8, 2025].

Chapter 9: Appendix

9.1 Data Visualization Plots

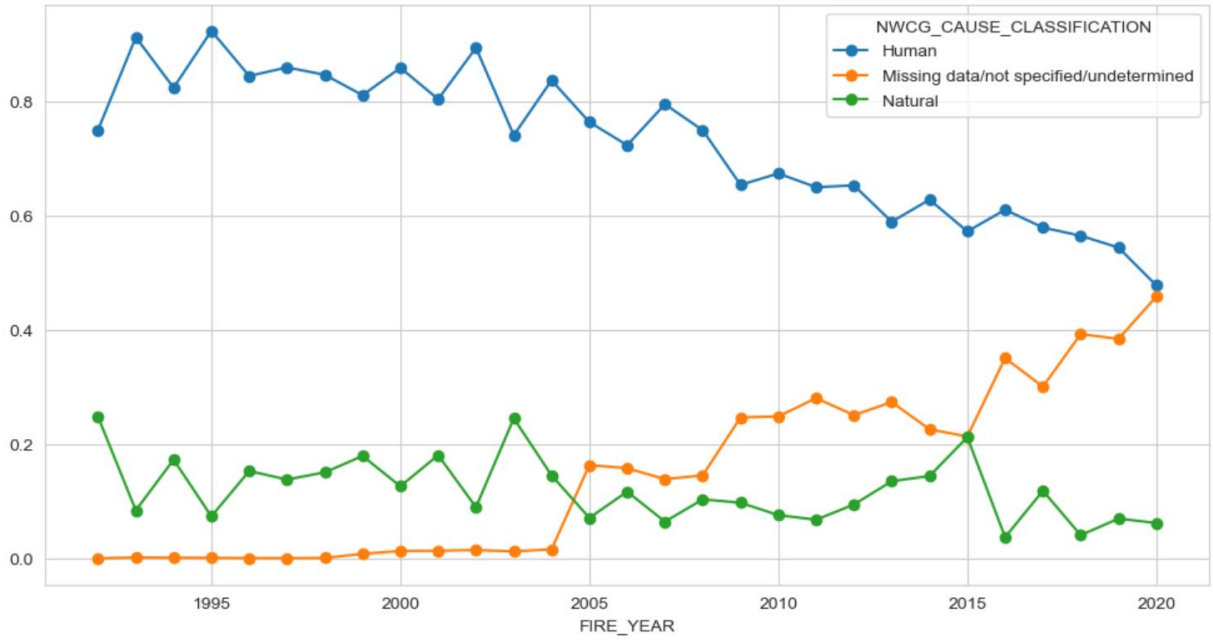

Figure 11: Yearly Proportion of Wildfire Cause

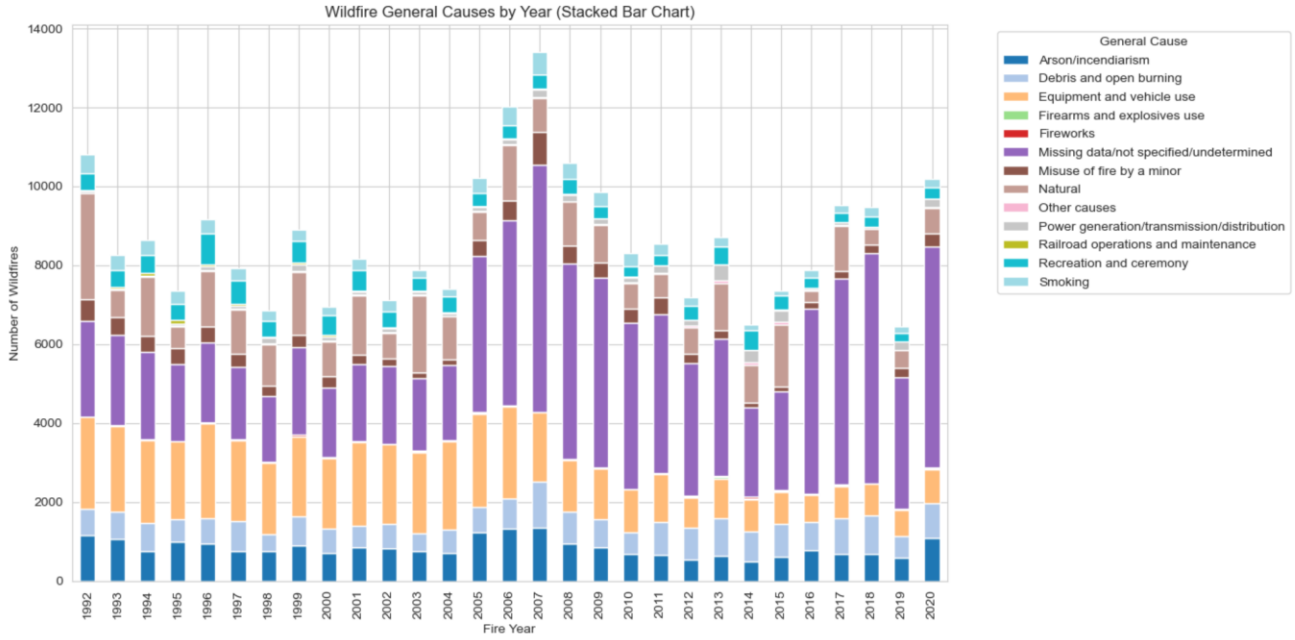

Figure 12: Stacked Bar Chart Showing Annual Distribution of Cause of Wildfire Per Year

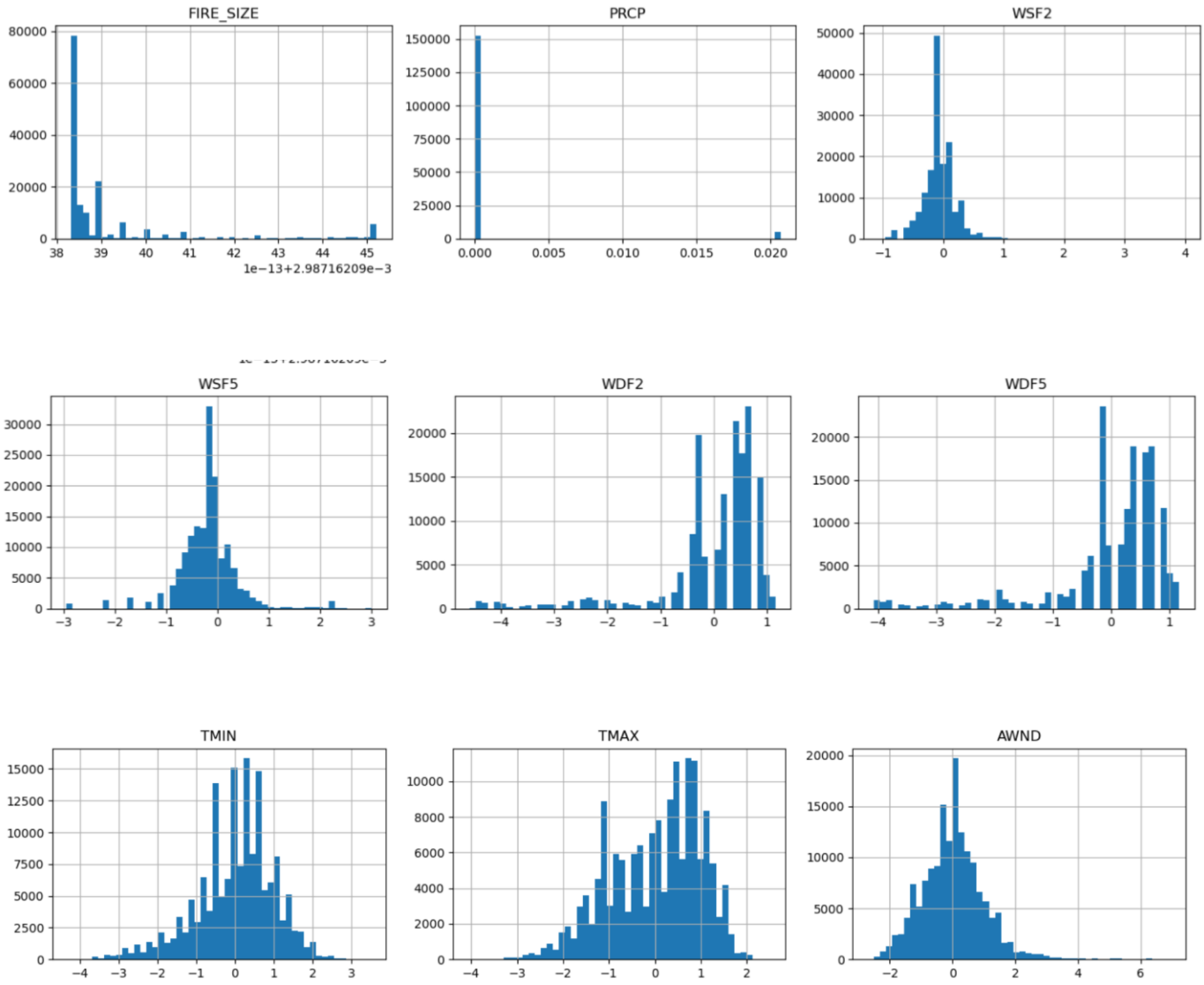

Figure 13: Histograms of Selected Numeric Features From the merged Wildfire and Weather Dataset

9.2 Baseline Model

Model	Accuracy	Class 0 (Precision / Recall / F1)	Class 1 (Precision / Recall / F1)
Random Forest	0.95	0.97 / 0.97 / 0.97	0.81 / 0.83 / 0.82
LightGBM	0.91	0.97 / 0.92 / 0.94	0.62 / 0.84 / 0.72
XGBoost	0.90	0.98 / 0.91 / 0.94	0.60 / 0.87 / 0.71
Logistic Regression	0.83	0.97 / 0.83 / 0.89	0.43 / 0.82 / 0.57

Table 8: Performance Comparison of Classification Models

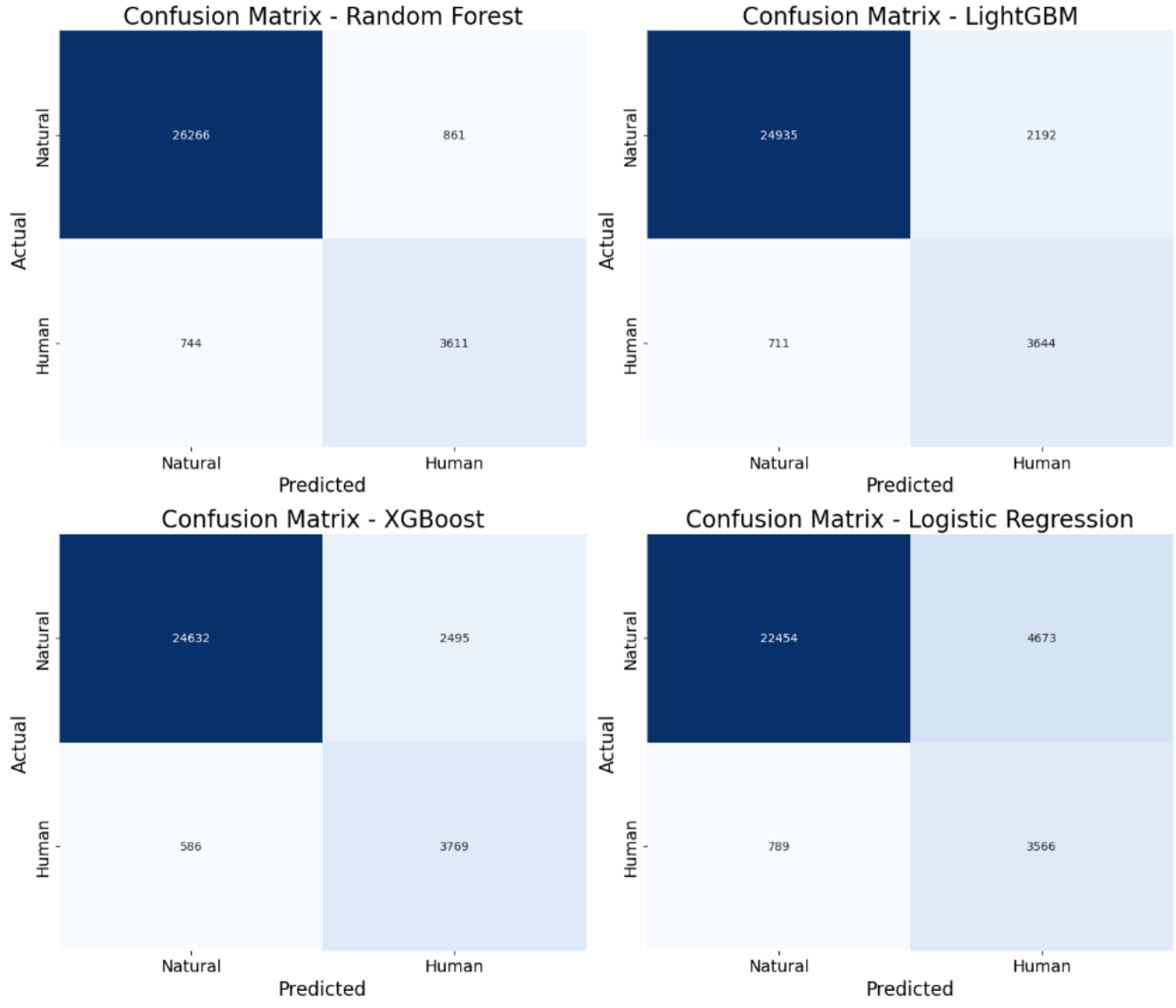

Figure 14: Confusion Matrix of Classifier Models for Wildfire Cause Prediction

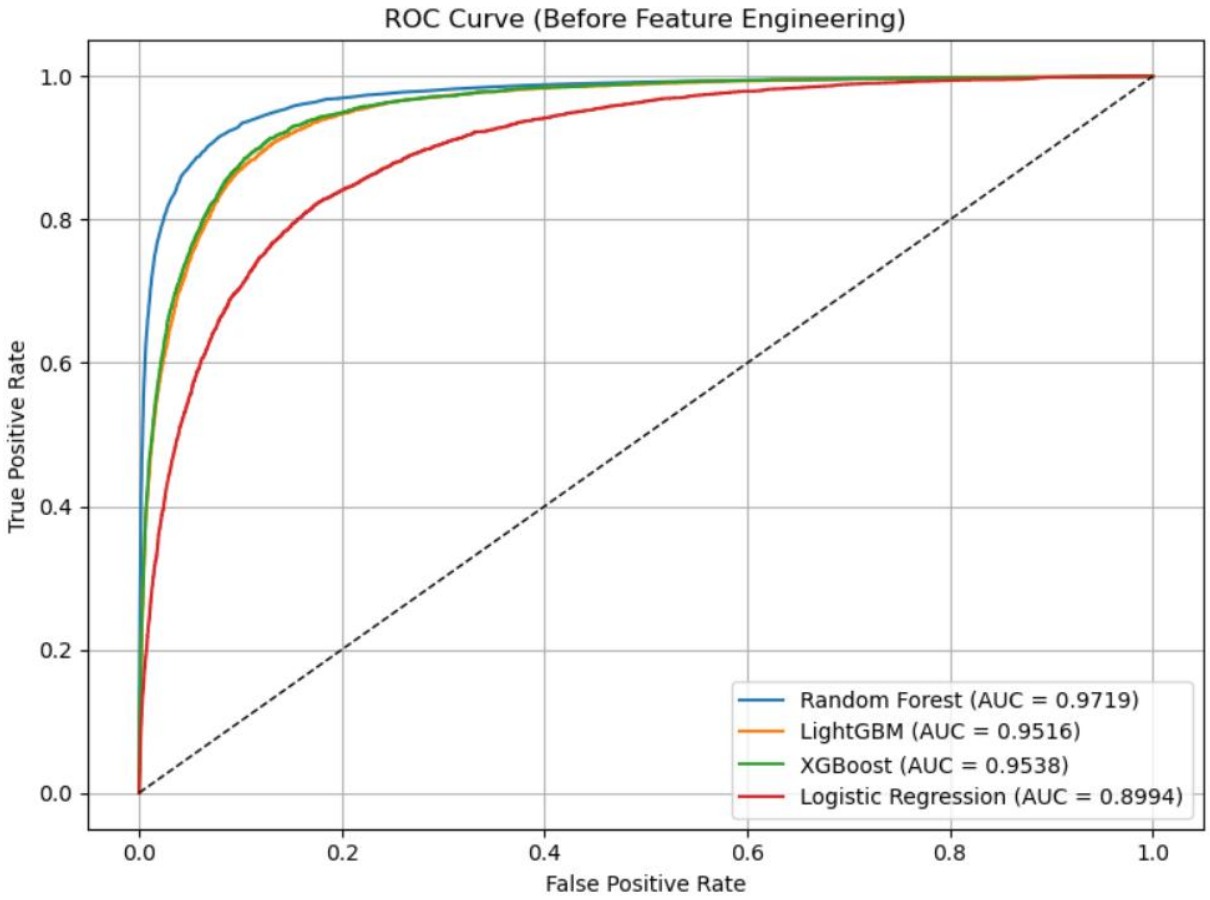

Figure 15: ROC-CURVE Of Classification Models before tuning or feature engineering

9.3 Cross Validation along with Feature Engineering

Model	CV Accuracy (mean ± std)	Test Accuracy
Random Forest	~0.958 ± 0.001	~0.933
LightGBM	~0.920 ± 0.001	~0.902
XGBoost	~0.914 ± 0.001	~0.899
Logistic Regression	~0.832 ± 0.002	~0.829

Table 9: Cross-Validation and Test Accuracy of Classifier Models

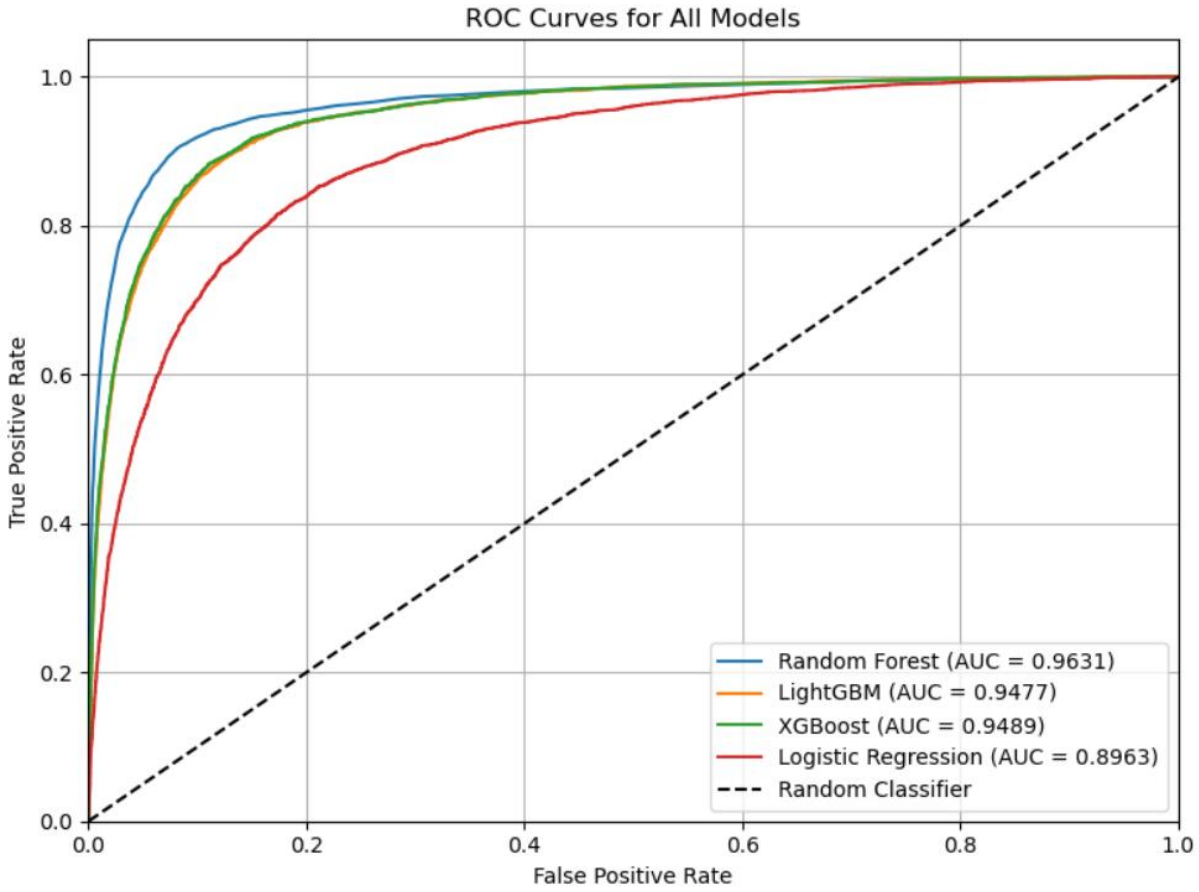

Figure 16: ROC-CURVE Of Classification Models with Cross validation and Feature Engineering

9.4 Hyperparameter Tuning with Final Random Forest Classifier

Metric	Value
Test Accuracy	0.9420
Precision (Class 0)	0.97
Recall (Class 0)	0.96
F1-Score (Class 0)	0.97
Precision (Class 1)	0.76
Recall (Class 1)	0.84
F1-Score (Class 1)	0.80
Macro Average	0.87 / 0.90 / 0.88 (Precision / Recall / F1)
Weighted Average	0.95 / 0.94 / 0.94 (Precision / Recall / F1)
Confusion Matrix	[[25979, 1148], [677, 3678]]
ROC-AUC Score	0.9011

Table 10: Random Forest Performance Summary

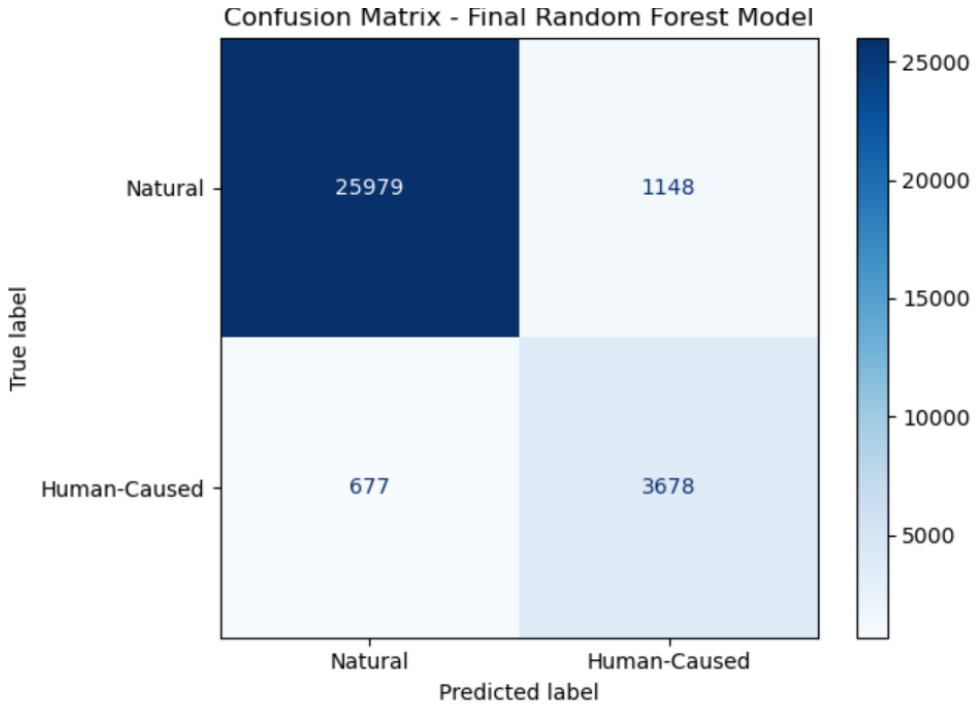

Figure 17: Confusion Matrix of Final Model (Random Forest)

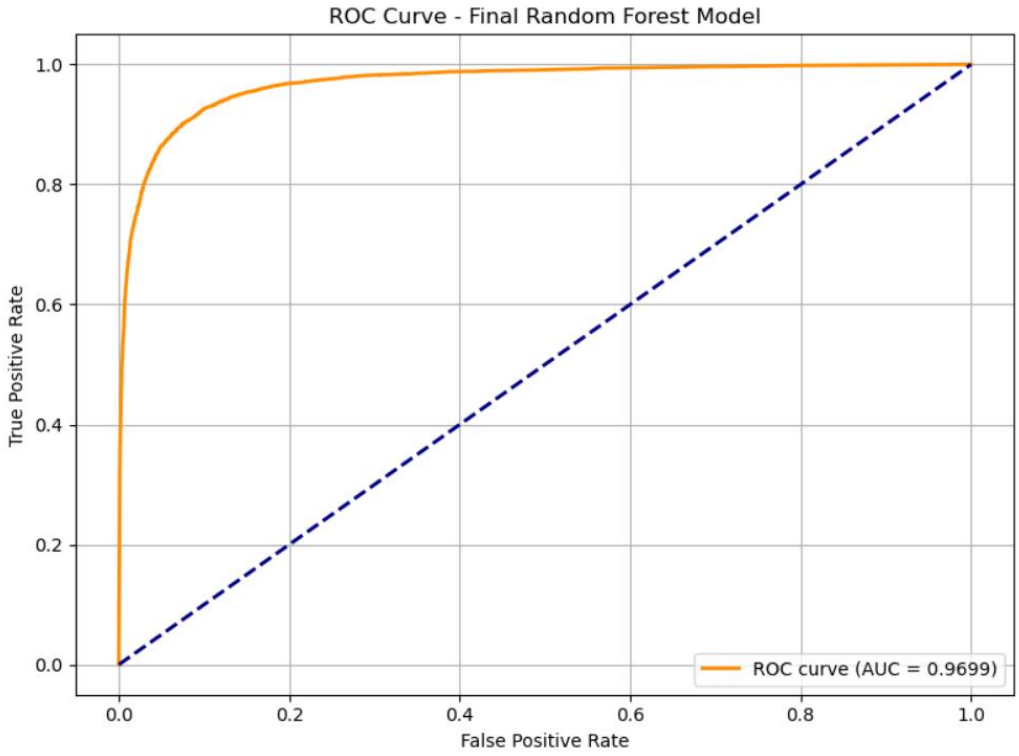

Figure 18: ROC-AUC of Final Model

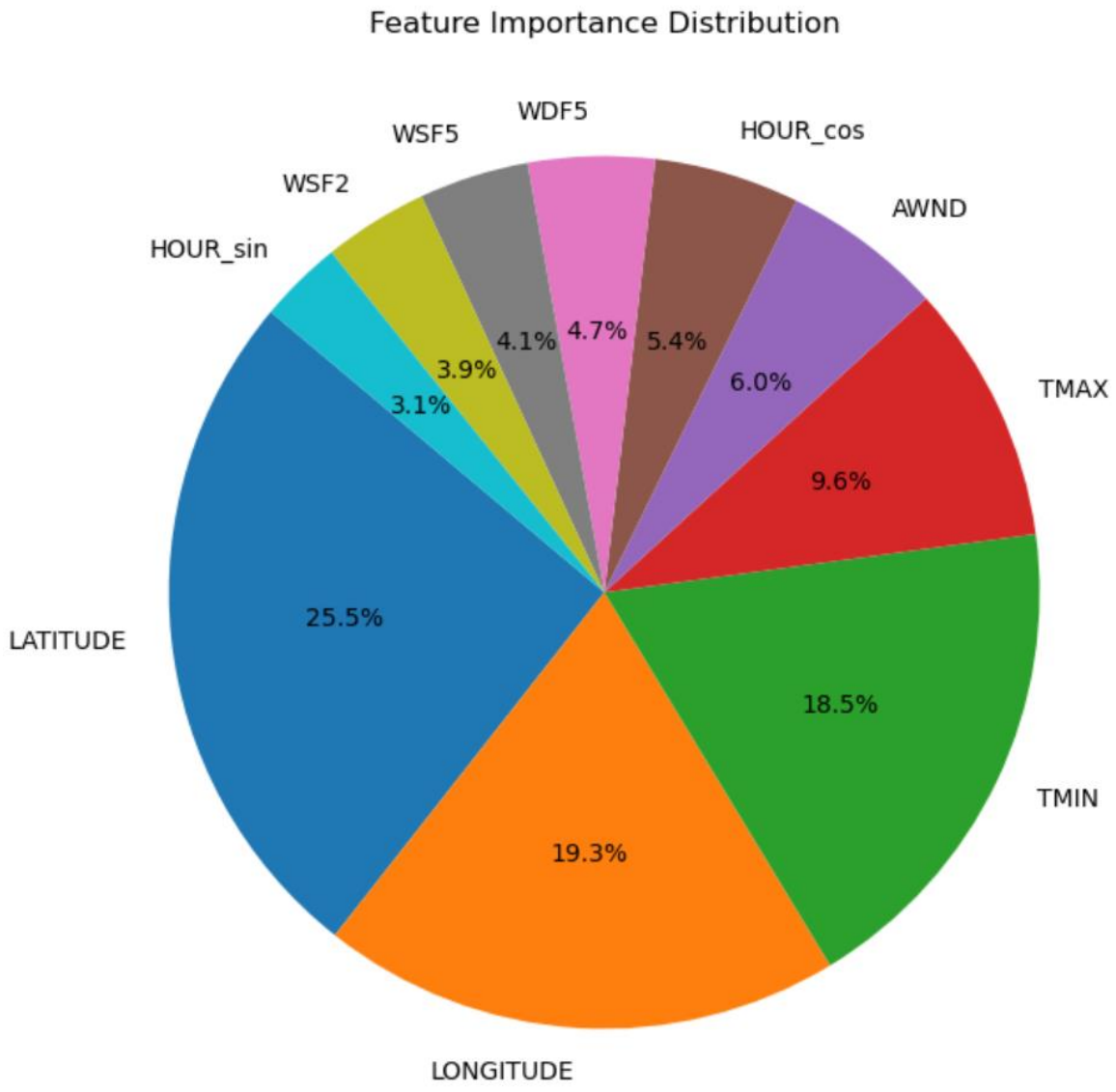

Figure 19: Feature Importance Distribution

9.5 Code Reference

https://github.com/annissapereira/fire-cause-predictor the full code for this work can be found here