# Integrated Wildfire Risk Prediction and Monitoring System Using Multi-Source Data and Advanced Machine Learning Techniques

Girish Jeswani
*University of Colorado, Boulder*
Colorado, USA
girish.jeswani@colorado.edu

Siddhant Kodolkar
*University of Colorado, Boulder*
Colorado, USA
siddhant.kodolkar@colorado.edu

Nishchal Shetty
*University of Colorado, Boulder*
Colorado, USA
Nishchal.Shetty@colorado.edu

Darshan Vijayaraghavan
*University of Colorado, Boulder*
Colorado, USA
darshan.vijayaraghavan@colorado.edu

## I. Introduction

This project focuses on developing a comprehensive wildfire risk prediction and monitoring system by integrating diverse data sources and leveraging advanced machine learning techniques. We have successfully collected wildfire data from NASA FIRMS and weather data using the Open Meteo API, establishing a foundation for robust analysis. Our team has conducted extensive Exploratory Data Analysis (EDA) on these datasets, uncovering valuable insights and visualizing patterns that influence wildfire risks.

Building on these findings, we have implemented our first predictive model using XGBoost, achieving a promising accuracy of 80.33%. This marks a significant milestone in our efforts to classify fire risk and predict future occurrences. The system is designed to incorporate additional data sources, such as satellite imagery, vegetation indices, topographical features, and human activity data, to further enhance its predictive capabilities. Future developments include real-time monitoring, predictive analytics, and a user-friendly dashboard for risk visualization and alerts, ensuring practical applications for wildfire management and prevention.

## II. Related Work

In this paper [1], a multi-modal wildfire prediction and early warning system has been developed based on a novel spatio-temporal machine learning architecture. A comprehensive wildfire database with over 37 million data points was created, including the historical wildfires, environmental and meteorological sensor data from the Environmental Protection Agency, and geological data. The data was augmented into $2.53\,\text{km} \times 2.53\,\text{km}$ square grids to overcome the sensor network coverage limitations. Leading and trailing indicators for the wildfires are proposed, classified, and tested. The leading indicators are correlated to the risks of wildfire conception, whereas the trailing indicators are correlated to the byproducts of the wildfires. Additionally, geological data was incorporated to provide additional information for better assessment on wildfire risks and propagation. Next, a novel U-Convolutional Long Short-Term Memory (ULSTM) neural network was developed to extract key spatial and temporal features of the dataset, specifically to address the spatial nature of the location of the wildfire and the time-progression temporal nature of the wildfire evolution. Through iterative improvements and optimization, the final ULSTM network architecture, trained with data from 2012 to 2017, achieved $> 97\%$ accuracy for predicting wildfires in 2018, as compared to $\sim 76\%$ using traditional Convolutional Neural Network (CNN) techniques. The final model was applied to conduct a retrospective study for the 2018–2022 wildfire seasons and successfully predicted $85.7\%$ of wildfires $> 300\,\text{K}$ acres in size. This technique could enable fire departments to anticipate and prevent wildfires before they strike and provide early warnings for at-risk individuals for better preparation, thereby saving lives, protecting the environment, and avoiding economic damages.

The method used in this paper [2] combines Big Data, Remote Sensing, and Data Mining algorithms (Artificial Neural Network and Support Vector Machine) to process data collected from satellite images over large areas and extract insights to predict the occurrence of wildfires and avoid such disasters. For this reason, a methodology was implemented by building a dataset based on Remote Sensing data related to the state of crops (NDVI), meteorological conditions (LST), as well as the fire indicator "Thermal Anomalies." These data were acquired from MODIS (Moderate Resolution Imaging Spectroradiometer), a key instrument aboard the Terra and Aqua satellites. Experiments were conducted using the big data platform "Databricks." The experimental results achieved high prediction accuracy of $98.32\%$. These results were assessed using several validation strategies (e.g., classification metrics, cross-validation, and regularization) and compared

with some wildfire early warning systems.

The paper [3] presents a method to predict large wild- fires using machine learning models, focusing on leveraging remote sensing and atmospheric reanalysis data across the United States. Wildfires, especially large ones, account for the major- ity of burned areas in the U.S., and predicting these events is crucial for miti- gation efforts. The research employs extensive data from NASA's MODIS satellite and the ERA5 atmospheric dataset, analyzing 2109 wildfire sites spanning two decades (1992-2020), representing over 14 million hectares burned.

The aim is to provide a scalable model for predicting wildfires while avoiding the computational intensity seen in traditional mod- els. The key environmental variables used in the study include Nor- malized Difference Vegetation Index (NDVI), Enhanced Vegetation Index (EVI), Leaf Area Index (LAI), Land Surface Temperature (LST) (both day and night), and Fraction of Photosynthetically Active Radiation (FPAR). These data, derived from MODIS satellite obser- vations, capture the vegetation health and surface temperature that contribute to fire risk. Atmospheric reanalysis data, such as wind components, relative humidity, and temperature, are integrated from the ERA5 dataset to assess weather conditions influencing wildfire spread. The study implements six machine learning mod- els: Logistic Regression, Decision Tree, Random Forest, XGBoost, K-Nearest Neighbors, and Support Vector Machine (SVM). Among these, XGBoost achieves the highest accuracy of 90.44%, with a true positive rate of 0.92 and a true negative rate of 0.88, making it the most effective at predicting large wildfires.

This research also emphasizes environmental justice by ana- lyzing the overlap between predicted wildfire occurrences and disadvantaged communities. Through spatial analysis, the study highlights regions where both large wildfires and vulnerable popu- lations coincide, such as Northern California and Oklahoma. This approach aligns with the Justice40 Initiative, which aims to direct 40% of federal benefits to underserved communities, making the model useful for prioritizing resource allocation and safeguarding those disproportionately impacted by wildfires. In conclusion, the paper demonstrates the potential of machine learning in wildfire prediction, particularly by leveraging widely available satellite and atmospheric data. The XGBoost model developed can be applied across the United States, offering a highly accurate and less compu- tationally demanding tool for wildfire prediction. This model holds promise for improving response times, directing federal resources, and aiding fire safety organizations in mitigating the impacts of large wildfires.

The paper [4] explores the use of deep learning to predict wildfire occurrences by leveraging historical satellite data. With wildfires increasing in frequency and severity due to climate change and human activity, accurate prediction models are crucial. The authors use advanced techniques such as convolutional autoencoders and U-Net architectures, which are particularly effective for image segmentation, and combine them with Long Short-Term Memory (LSTM) networks to capture both spatial and temporal features in satellite imagery.

The study utilizes data from the MODIS (Moderate Resolution Imaging Spectroradiometer) satellite, which provides valuable re- mote sensing observations, including vegetation health, land sur- face temperature, and historical fire activity. By applying convo- lutional layers, the models can extract spatial patterns from satel- lite images, while the LSTMs analyze time-based trends in fire occurrence. This combined model helps to predict where and when wildfires are likely to occur.

Training and evaluating the model posed challenges, particularly due to the imbalance in the dataset, where non-fire regions vastly outnumber fire-affected areas. To address this, the researchers im- plemented data augmentation techniques and worked on improving the sensitivity of the model to smaller fire events, which are often harder to detect. This approach allows the system to perform better in identifying smaller fires as well as larger fire risks, which is essential for timely wildfire prevention.

The paper [5] talks about predicting the fire arrival times at a given spatial location to calculate the fire perimeters which are an accurate representation of the history of wildfire movement. A Conditional Wasserstein Generative Adversarial Network (cWGAN) is trained with 20 ideal simulations of forest fires by the WRF-SFIRE models to predict the fire arrival times. The cWGAN contains a generater and a critic , the generator generates a list of possible fire arrival times, and the critic tries to modify this prediction to get the most accurate value. Samples produced by the cWGAN are further used to assess the uncertainty of predictions. The cWGAN is tested on four California wildfires occurring between 2020 and 2022, and predictions for re extent are compared against high-resolution airborne infrared measurements. Further, the predicted ignition times are compared with reported ignition times. The predicted values show an average Sørensen's coefficient of 0.81 for the fire perimeters which informs that the actual and the predicted fire arrival times have a strong agreement. Also the fire ignition times of the predicted and the actual values differ by an average of 32 mins.

This paper [6] presents a comparative study between four popular Machine Learning methods - decision tree, random forest, k-nearest neighbors (KNN), and support vector machine (SVM) - for forest fire prediction, in terms of accuracy, precision, recall, and F1 score. Their experiments show that the decision tree outperforms the other three algorithms, achieving an accuracy of 97.95%, a precision of 100%, a recall of 97.05%, and an F1 score of 98.5%. The study also compares the time taken by the models to produce the results.

Paper [7] emphasizes how climate change will intensify

the danger of wildfires, significantly impacting human life. Deep Learning (DL) has been extensively applied in wildfire prediction research. In wildfire prediction, previous deep-learning methods have overlooked the inherent differences between static positional information and dynamic variables. Additionally, most existing deep learning models have not integrated the global system characteristics of the Earth's features and teleconnection during the learning phase. Here, we propose a static location-aware ConvLSTM (SLA-ConvLSTM) model that is aware of static positional elements and interconnected with global information and teleconnection. The proposed model can discern dynamic variables' influence across various geographical locations on predictive outcomes. Compared with other deep learning models, their SLA-ConvLSTM model has achieved commendable performance. The outcomes indicate that the collaborative interplay of spatiotemporal features and the extraction of static positional information present a promising technique for wildfire prediction. Moreover, the incorporation of climate indices and global feature variables enhances the predictive capability of their model in wildfire prediction.

Paper [8] looks to move away from the traditional prediction methods in the case of wildfires. Wildfires result in billions of dollars in damage each year and are expected to increase in frequency, duration, and severity due to climate change. The current state-of-the-art wildfire spread models rely on mathematical growth predictions and physics-based models, which are difficult and computationally expensive to run. They present and evaluate a novel system, FireCast. FireCast combines artificial intelligence (AI) techniques with data collection strategies from geographic information systems (GIS). FireCast predicts which areas surrounding a burning wildfire have high-risk of near-future wildfire spread, based on historical fire data and using modest computational resources. FireCast is compared to a random prediction model and a commonly used wildfire spread model, Farsite, outperforming both with respect to total accuracy, recall, and F-score.

While existing research has laid a strong foundation for wildfire prediction and monitoring, our study addresses several gaps and introduces unique contributions compared to prior work. Unlike the multi-modal approach in [1], which primarily focuses on spatio-temporal data from EPA sensors and large-scale grids, our work integrates high-resolution fire data from NASA FIRMS and meteorological data from Open Meteo with a regional focus on Queensland, Australia. This regional specificity enables us to capture environmental and historical fire patterns unique to this fire-prone area, which contrasts with the generalized approaches in [1] and [3] that model wildfire risks at national or global scales. Further, while [2] utilizes vegetation indices such as NDVI and thermal anomalies from MODIS data, our work extends beyond remote sensing by incorporating additional environmental factors such as soil moisture, soil temperature, and evapotranspiration. These features are critical for understanding wildfire risks in regions with distinct soil and vegetation profiles.

In comparison to studies like [4] and [7], which rely heavily on deep learning architectures such as U-Net and ConvLSTM, our study begins with simpler, interpretable models such as XGBoost and Random Forest to establish a baseline before progressing to advanced architectures like LSTM. This gradual approach not only allows us to validate the effectiveness of traditional models but also ensures computational efficiency and interpretability, which are critical for operational use by fire management agencies. Moreover, while the generative models used in [5] focus on predicting fire arrival times and perimeters, our study prioritizes predicting the likelihood of fire occurrences based on weather and environmental conditions. This predictive approach, combined with a balanced dataset of fire and non-fire events, provides a robust framework for early warning systems, distinct from the retrospective analyses seen in studies like [5].

While [6] evaluates the comparative performance of lightweight models like Decision Tree and Random Forest, achieving impressive metrics, our project focuses on combining these models with high-dimensional, multi-source data to create an integrated pipeline. Our approach ensures that the predictions are not only accurate but also robust across diverse scenarios, a contrast to the single-data-source reliance in [6].

Finally, [8] introduces FireCast for wildfire spread prediction, focusing on modest computational resources and comparisons with physics-based models. In contrast, our study emphasizes wildfire occurrence prediction rather than spread, integrating satellite, meteorological, and environmental data. This focus allows us to develop early warning systems with immediate applicability to proactive fire management in Queensland. Finally, [8] introduces FireCast, which focuses on predicting wildfire spread based on historical fire data and GIS techniques, emphasizing modest computational resources and comparisons with physics-based models. In contrast, our study focuses on predicting the likelihood of wildfire occurrences rather than spread. By integrating satellite, meteorological, and environmental data with high-resolution regional specificity, our approach enables the creation of early warning systems tailored to the unique conditions of Queensland.

In summary, our study differentiates itself by combining regional specificity, multi-source data integration, practical implementation, and a gradual progression from traditional machine learning models to advanced deep learning architectures. These aspects not only enhance prediction accuracy but ensure the usability and scalability of wildfire management in Queensland and similar regions worldwide.

## III. Methodology

### A. Objectives

This study plans to develop an integrated data pipeline to gather wildfire-related information from multiple sources, enabling efficient data management and analysis. It aims to create advanced machine learning models for accurate wildfire risk prediction. Additionally, it will implement a real-time monitoring and alert system to track fire activity and provide timely warnings. The study will also provide interpretable insights to support fire management decision-making. Furthermore, it intends to assess long-term wildfire risks under different climate change scenarios to aid in future mitigation and preparedness efforts.

### B. Data Collection/Integration

This study has successfully implemented APIs for NASA FIRMS [9] [10] and Open Meteo [11] to collect wildfire-related and weather data, respectively. The data is stored in a structured format to facilitate analysis and modeling. Exploratory Data Analysis (EDA) has been performed on these datasets to extract insights and visualize patterns that influence wildfire risks. Future plans include incorporating data from additional sources, such as USGS Earth Explorer for satellite imagery and OpenStreetMap for human activity data. A unified schema will be established to merge these datasets, ensuring seamless integration, consistent data structure, and efficient processing for advanced wildfire risk prediction and monitoring.

### C. Datasets

Tables [1] and [2] give the detailed descriptions of both the datsets that were used in our project. Fire data was collected from NASA FIRMS for Australia from years 2000 through 2024, which resulted in over 15,000,000 rows. Keeping training times in mind the dataset was reduced to fire data from Queensland for the years 2021 through 2024, and the resulting dataset was just over 100,000 rows.

The coordinate attributes were used to fetch weather data from the resulting data. Finally, both the datasets were combined to make our training data which is described in Table 2.

### D. Training

This study began its training process by implementing and optimizing an XGBoost model for wildfire risk prediction, achieving an initial accuracy of 80.33%. Building on this foundation, the study will further implement and optimize a Random Forest Classifier to refine prediction accuracy. Additionally, an LSTM network will be developed for sequence-to-sequence prediction of time-series data, capturing temporal dependencies in wildfire occurrences. If time permits, The study also plans to design a CNN architecture tailored for processing satellite imagery. Ultimately, the outputs from these models will be combined to enhance prediction accuracy and provide comprehensive insights for wildfire risk assessment and management.

### E. Feature Engineering

The current dataset used to train the XGBoost model includes a wide range of features: temperature, humidity, dew point, precipitation, wind speed, wind direction, cloud cover, pressure, soil temperature, soil moisture, and other weather-related variables. Additionally, historical fire data for the past three years in Queensland, Australia, has been integrated into the dataset, providing critical insights into fire patterns and occurrences. These features collectively establish a strong foundation for predicting wildfire risks based on environmental, meteorological, and historical conditions.

Future feature engineering efforts will expand this dataset by calculating vegetation indices, such as the Normalized Difference Vegetation Index (NDVI) and Enhanced Vegetation Index (EVI), from satellite imagery to assess vegetation health and fire susceptibility. Topographical features like elevation and slope will be generated to account for terrain-related fire risks. Additional weather-derived features, such as the Fire Weather Index, will be developed to incorporate more sophisticated meteorological risk factors.

Furthermore, human activity indices derived from OpenStreetMap data will be included to account for anthropogenic factors influencing wildfire risks. These enhancements aim to create a more robust and comprehensive predictive model for wildfire risk assessment.

## IV. Implementation

### A. Exploratory Data Analysis

The exploratory data analysis (EDA) for NASA FIRMS data began by consolidating datasets from three different satellites into a single unified dataset. Key preprocessing steps included addressing inconsistencies in the confidence column, which contains both categorical and numeric values. A standardization function was implemented to classify all values into low (l), nominal (n), or high (h) categories, ensuring consistency across the dataset.

To analyze spatiotemporal patterns, records were filtered by year for visualization. Using geospatial libraries like Cartopy and GeoPandas, annual scatter plots of fire brightness were generated overlaid on a map of Australia, providing a clear visual representation of fire activity distribution across the region. Additionally, a bar chart was created to display the yearly frequency of fire occurrences, offering insights into trends over time. The dataset was further analyzed for monthly wildfire frequency. A bar chart with labeled months revealed seasonal trends in wildfire activity, highlighting peak months for fire occurrences.

These visualizations and preprocessing steps lay the foundation for integrating FIRMS data into the wildfire risk prediction model, enabling robust feature engineering and improved data interpretation.

The OpenMeteo Weather API was used to collect meteorological data for Queensland. To overcome the API's strict call

| Attribute | Description (Instrument MODIS) | Description (Instrument VIIRS) |
|---|---|---|
| Latitude | Center of 1 km fire pixel, but not necessarily the actual location of the fire as one or more fires can be detected within the 1 km pixel. | Center of nominal 375 m fire pixel. |
| Longitude | Center of 1 km fire pixel, but not necessarily the actual location of the fire as one or more fires can be detected within the 1 km pixel. | Center of nominal 375 m fire pixel. |
| Brightness | Channel 21/22 brightness temperature of the fire pixel measured in Kelvin. | VIIRS I-4 channel brightness temperature of the fire pixel measured in Kelvin. |
| Scan and Track | The algorithm produces 1 km fire pixels, but MODIS pixels get bigger toward the edge of scan. Scan and track reflect actual pixel size. | The algorithm produces approximately 375 m pixels at nadir. Scan and track reflect actual pixel size. |
| Acq_Date (Acquisition Date) | Date of MODIS acquisition. | Date of VIIRS acquisition. |
| Acq_Time (Acquisition Time) | Time of acquisition/overpass of the satellite (in UTC). | Time of acquisition/overpass of the satellite (in UTC). |
| Satellite | A = Aqua and T = Terra. | N = Suomi National Polar-orbiting Partnership (Suomi NPP), N20 = NOAA-20 (designated JPSS-1 prior to launch), N21 = NOAA-21 (designated JPSS-2 prior to launch). |
| Confidence (0 - 100%) | This value is based on a collection of intermediate algorithm quantities used in the detection process. Confidence estimates range between 0 and 100% and are assigned one of the three fire classes (low-confidence fire, nominal-confidence fire, or high-confidence fire). | This value is based on a collection of intermediate algorithm quantities used in the detection process. Low confidence daytime fire pixels are typically associated with areas of sun glint and lower relative temperature anomaly ($<$15K) in the mid-infrared channel I4. High confidence fire pixels are associated with saturated pixels. |
| Version | Version identifies the collection (e.g., MODIS Collection 6.1) and source of data processing (Ultra Real-Time (URT suffix), Real-Time (RT suffix), Near Real-Time (NRT suffix), or Standard Processing). Example: "6.1URT" - Collection 6.1 Ultra Real-Time processing. | Version identifies the collection (e.g., VIIRS Collection 1) and source of data processing. Example: "1.0NRT" - Collection 1 NRT processing, "1.0" - Collection 1 Standard processing. |
| Bright_T31 for MODIS / Bright_ti5 for VIIRS | Channel 31 brightness temperature of the fire pixel measured in Kelvin. | I-5 Channel brightness temperature of the fire pixel measured in Kelvin. |
| FRP (Fire Radiative Power in megawatts) | Depicts the pixel-integrated fire radiative power in MW (megawatts). | FRP depicts the pixel-integrated fire radiative power in MW (megawatts). The VIIRS 375 m fire detection algorithm was customized to optimize its response over small fires while balancing false alarms. |
| DayNight (Day or Night) | D = Daytime fire, N = Nighttime fire. | D = Daytime fire, N = Nighttime fire. |
| Type (Inferred hot spot type) | 0 = presumed vegetation fire, 1 = active volcano, 2 = other static land source, 3 = offshore. | 0 = presumed vegetation fire, 1 = active volcano, 2 = other static land source, 3 = offshore. |

TABLE I: Details of Attributes in MODIS [9] and VIIRS [10] Data

limit, multiple VPN profiles were created and cycled through, enabling continuous data collection. The timestamps of the weather data were adjusted to align with fire occurrence data obtained from NASA's FIRMS dataset. A new binary column, fire, was added to the dataset, where a value of 1 indicated the presence of fire and 0 denoted its absence. To balance the dataset, additional non-fire data points were sampled from the weather data, labeled as 0, and combined with the fire data.

This consolidated dataset was used for correlation analysis to understand patterns and relationships between features. A five-number summary (minimum, first quartile, median, third quartile, maximum) was computed for all variables, followed by visual exploration through box plots, line graphs, pair plots, and a correlation heat-map. The analysis revealed that forest fires were more likely to occur when temperatures exceeded 30°C, relative humidity was above 50%, and precipitation was near 0 mm. These conditions reflect environmental patterns commonly associated with fire-prone areas, validating the pre-processing and data integration efforts.

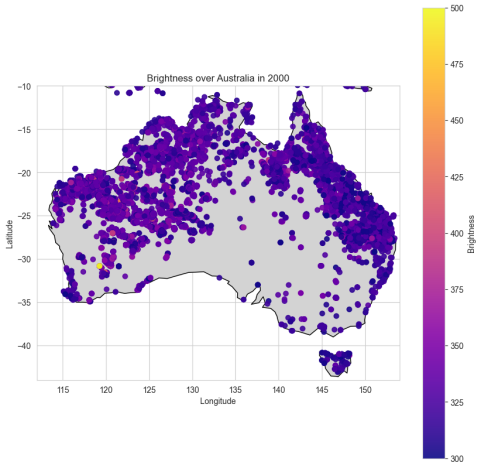


Fig. 1: Wildfire density in year 2000

| Variable | Unit | Description |
|---|---|---|
| temperature_2m | °C (°F) | Air temperature at 2 meters above ground |
| Air temperature at 2 meters above ground | % | Relative humidity at 2 meters above ground |
| dew_point_2m | °C (°F) | Dew point temperature at 2 meters above ground |
| apparent_temperature | °C (°F) | Apparent temperature is the perceived feels-like temperature combining wind chill factor, relative humidity and solar radiation |
| precipitation | mm (inch) | Total precipitation (rain, showers, snow) sum of the preceding hour |
| rain | mm (inch) | Only liquid precipitation of the preceding hour |
| snowfall | cm (inch) | Snowfall amount of the preceding hour in centimeters. For the water equivalent in millimeter, divide by 7. E.g. 7 cm snow = 10 mm precipitation water equivalent |
| snow_depth | meters | Snow depth on the ground |
| weather_code | WMO code | Weather condition as a numeric code. Follow WMO weather interpretation codes. See table below for details |
| pressure_msl surface_pressure | hPa | Atmospheric air pressure reduced to mean sea level (msl) or pressure at surface. Typically pressure on mean sea level is used in meteorology. Surface pressure gets lower with increasing elevation |
| cloud_cover | % | Total cloud cover as an area fraction |
| cloud_cover_low | % | Low level clouds and fog up to 3 km altitude |
| cloud_cover_mid | 0% | Mid level clouds from 3 to 8 km altitude |
| cloud_cover_high | % | High level clouds from 8 km altitude |
| et0_fao_evapotranspiration | mm (inch) | ET Reference Evapotranspiration of a well watered grass field. Based on FAO-56 Penman-Monteith equations ET is calculated from temperature, wind speed, humidity and solar radiation. Unlimited soil water is assumed. ET is commonly used to estimate the required irrigation for plants |
| vapour_pressure_deficit | kPa | Vapour Pressure Deficit (VPD) in kilopascal (kPa). For high VPD (¿1.6), water transpiration of plants increases. For low VPD (¡0.4), transpiration decreases |
| wind_speed_10m<br>wind_speed_100m | km/h (mph, m/s, knots) | Wind speed at 10 and 100 meters above ground. Wind speed on 10 meters is the standard level. |
| wind_direction_10m<br>wind_direction_100m | ° | Wind direction at 10 and 100 meters above ground |
| wind_gusts_10m | km/h (mph, m/s, knots) | Gusts at 10 meters above ground as a maximum of the preceding hour |
| soil_temperature_0_to_7cm<br>soil_temperature_7_to_28cm<br>soil_temperature_28_to_100cm<br>soil_temperature_100_to_255cm | °C (°F) | Temperature in the soil |
| soil_moisture_0_to_7cm<br>soil_moisture_7_to_28cm | $m^3/m^3$ | Average soil water content as volumetric mixing ratio |

TABLE II: Description of the Features present in the training data [**?**]

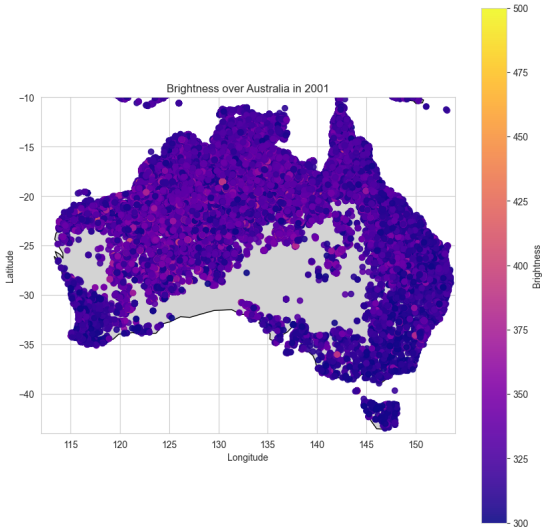


Fig. 2: Wildfire density in year 2001

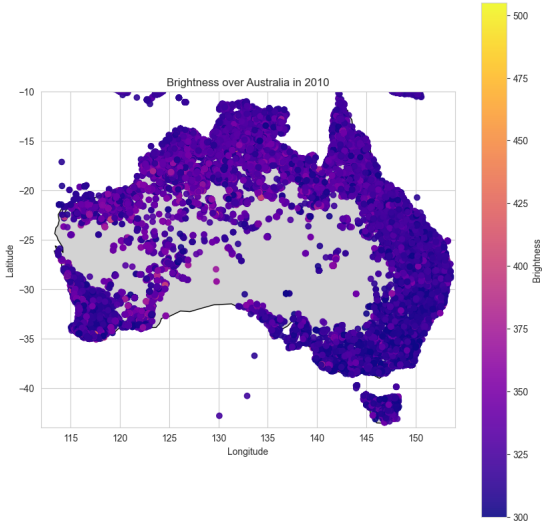


Fig. 3: Wildfire density in year 2010

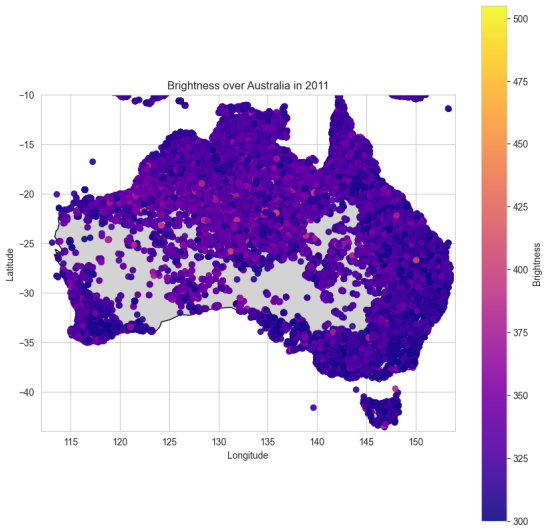


Fig. 4: Wildfire density in year 2011

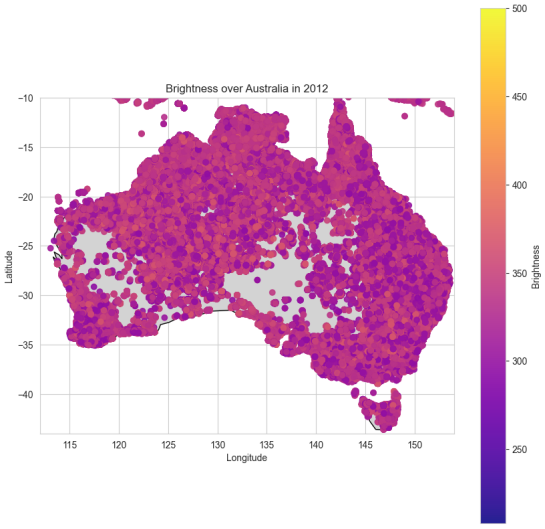


Fig. 5: Wildfire density in year 2012

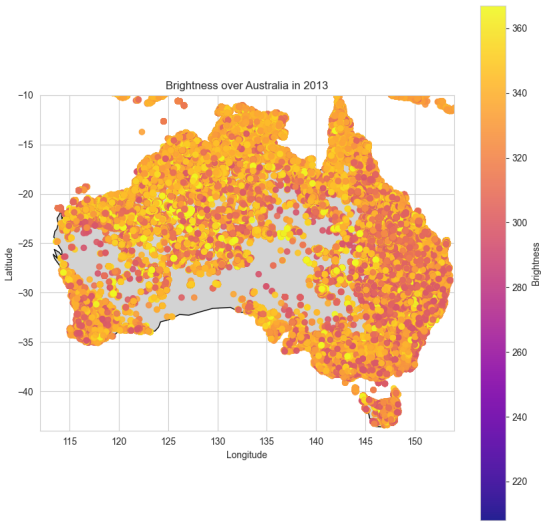


Fig. 6: Wildfire density in year 2013

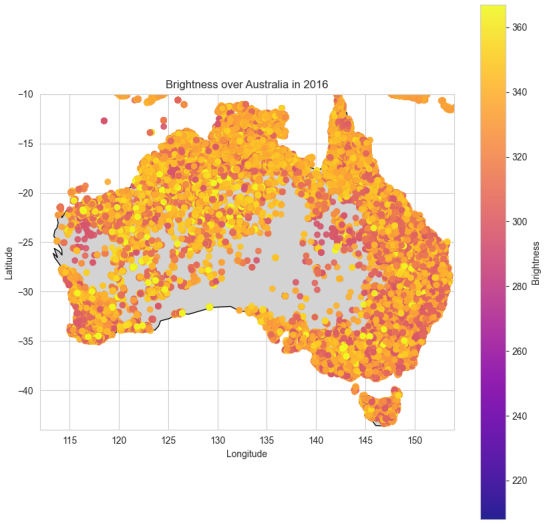


Fig. 7: Wildfire density in year 2016

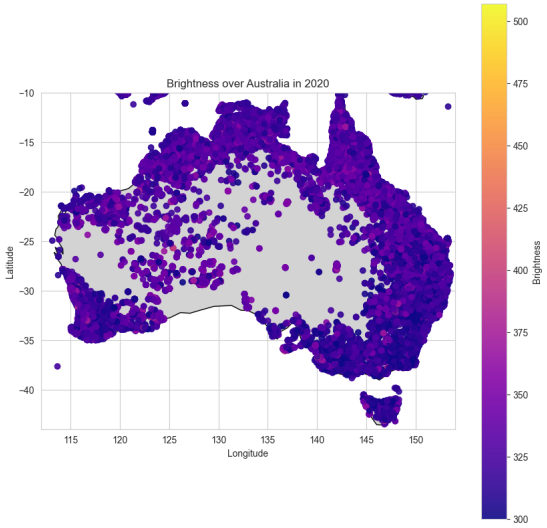


Fig. 8: Wildfire density in year 2020

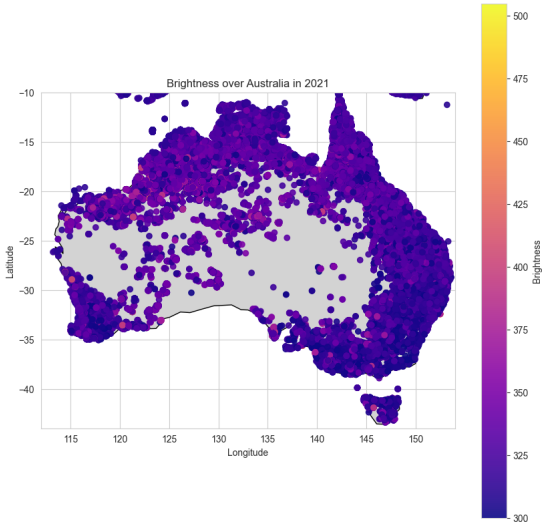


Fig. 9: Wildfire density in year 2021

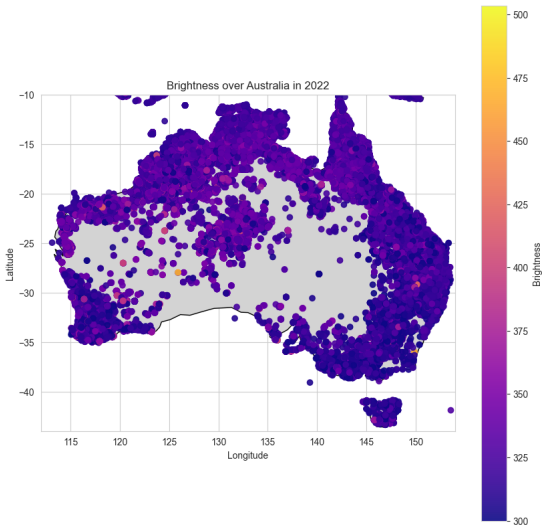


Fig. 10: Wildfire density in year 2022

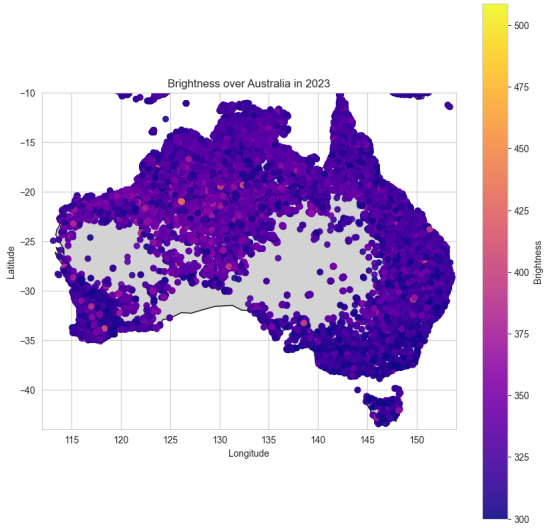


Fig. 11: Wildfire density in year 2023

The Geo-spatial analysis above shows the density of fires in Australia every year.

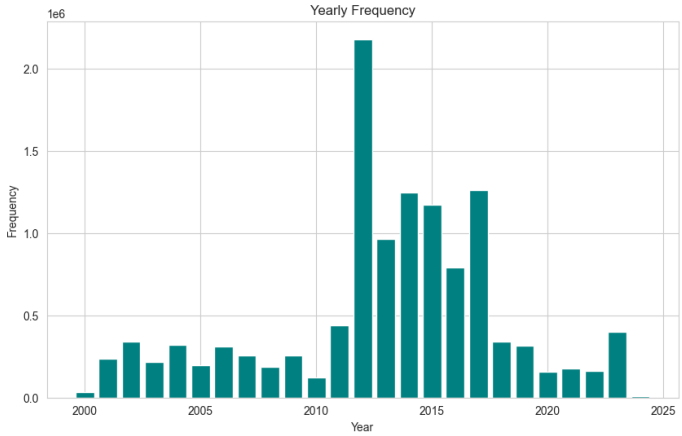


Fig. 12: Temporal Analysis of wildfires in Australia

It was quite interesting that more than 60% of the wildfires after 2000 occurred after the year 2012.

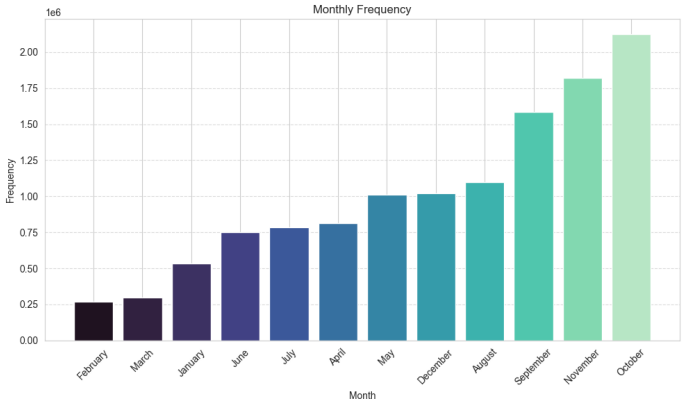


Fig. 13: Wildfire frequency by month

Most of the fires occurred in the summer months in Australia.

```
In [41]: balanced_df.isna().sum()
Out[41]: date                          193
         temperature_2m                193
         relative_humidity_2m          193
         dew_point_2m                  193
         apparent_temperature          193
         precipitation                 193
         rain                          193
         snowfall                      193
         snow_depth                    193
         weather_code                  193
         pressure_msl                  193
         surface_pressure              193
         cloud_cover                   193
         cloud_cover_low               193
         cloud_cover_mid               193
         cloud_cover_high              193
         et0_fao_evapotranspiration    193
         vapour_pressure_deficit       193
         wind_speed_10m                193
         wind_speed_100m               193
         wind_direction_10m            193
         wind_direction_100m           193
         wind_gusts_10m                193
         soil_temperature_0_to_7cm     193
         soil_temperature_7_to_28cm    193
         soil_temperature_28_to_100cm  193
         soil_temperature_100_to_255cm 193
         soil_moisture_0_to_7cm        193
         soil_moisture_7_to_28cm       193
         soil_moisture_28_to_100cm     193
         soil_moisture_100_to_255cm    193
         fire                            0
         dtype: int64

In [42]: balanced_df = balanced_df.dropna()
```

Fig. 14: Null handling in weather data

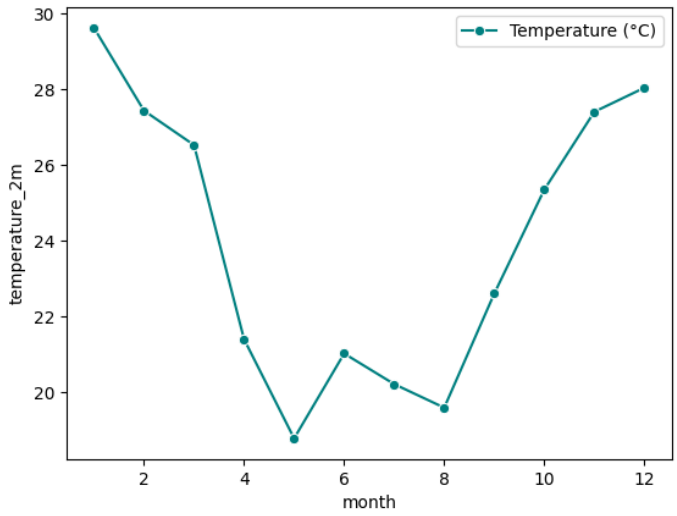


Fig. 15: Temperature trends over a year

Months September through March had the highest temperatures over a year, which correlates with the wildfire trends we observed.

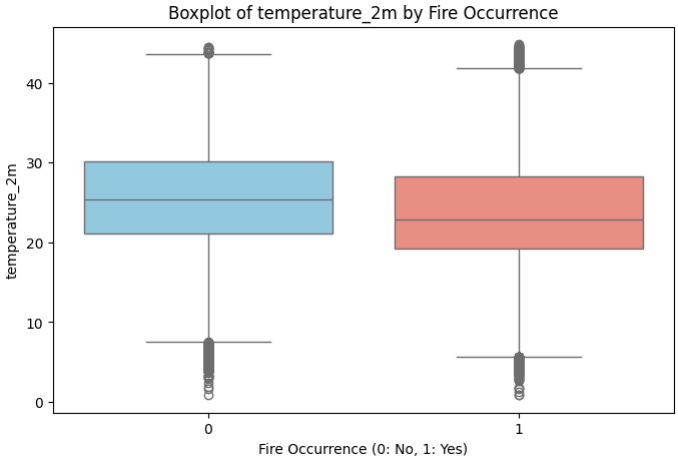


Fig. 16: Difference in fire occurrence for temperature

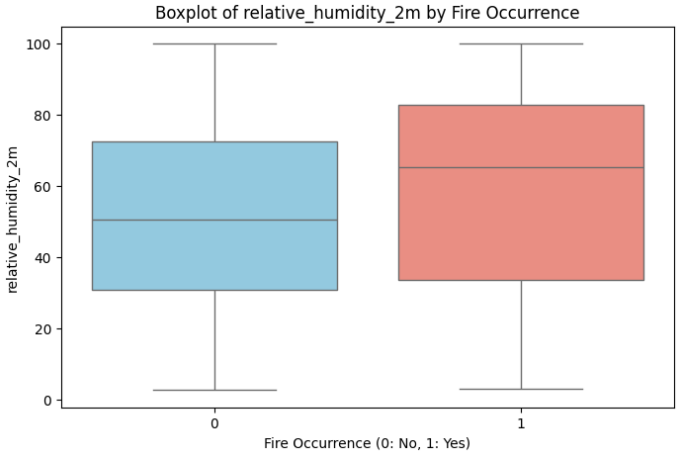


Fig. 17: Difference in fire occurrence for relative$_h umidity$

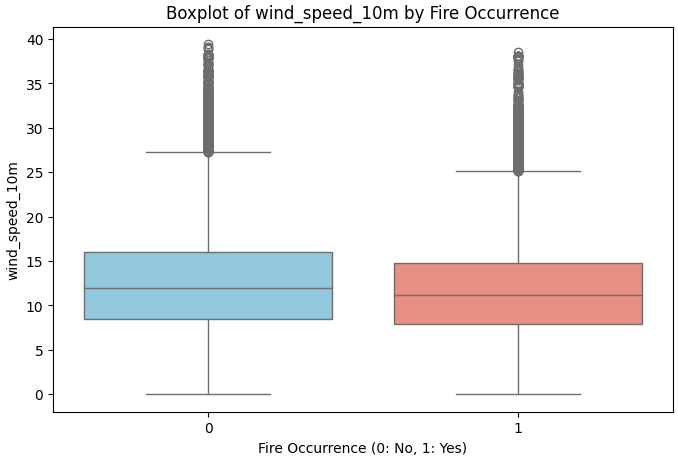


Fig. 18: Difference in fire occurrence for wind$_s peed$

There was a marginal difference in the number of fire occurrences for the attributes ”temperature” and ”wind_speed”.

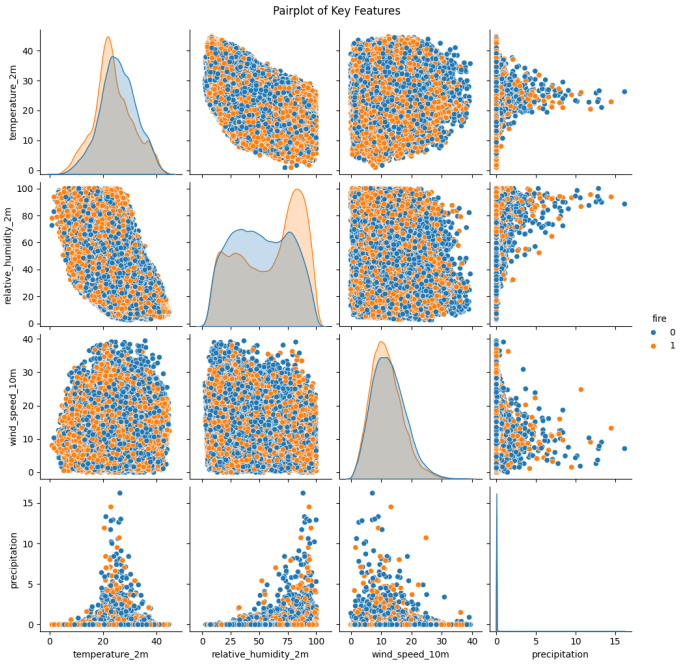


Fig. 19: Pairplot for final training data

There was an apparent negative correlation between temperature and relative humidity. The other attributes were not correlated.

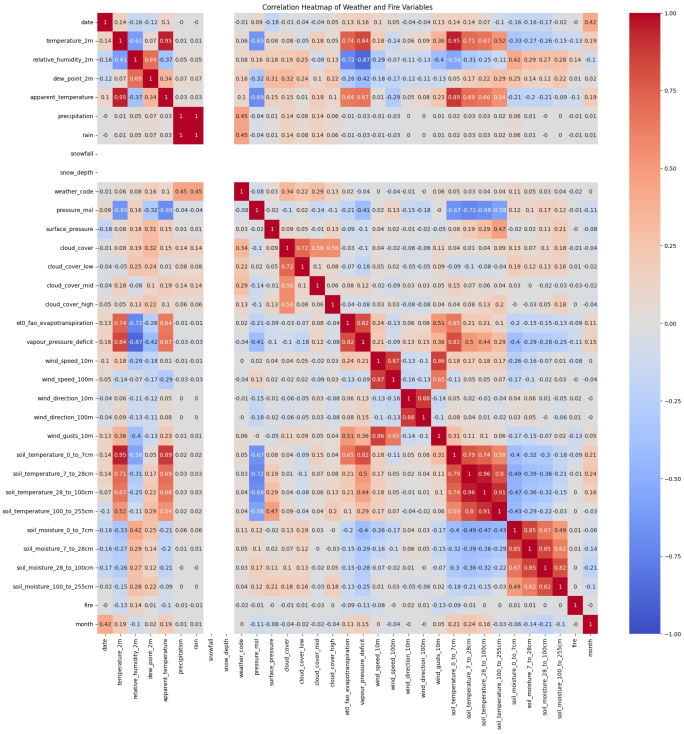


Fig. 20: Heatmap of final training data

## B. Model Building

The dataset is preprocessed by handling the missing values, removing the fire binary column and scaling the features for uniformity. The data is split into training and testing sets to ensure a fair evaluation.

Following is a brief discussion of each model based on its performance metrics and general characteristics:

- **XGBoost**: A high-performance gradient boosting model achieving balanced precision (0.81), recall (0.80), and an impressive AUC-ROC (0.876). It is robust for handling complex datasets and excels in feature importance analysis.
- **Random Forest**: Offers reliable accuracy (0.79) and precision (0.80) with a slightly lower recall (0.79). This ensemble learning method is effective for reducing overfitting and handling missing data.
- **KNN (K-Nearest Neighbors)**: Achieves moderate accuracy (0.76) and recall (0.76) but struggles with scalability in large datasets. Its simplicity makes it effective for small-scale problems.
- **SVM (Support Vector Machine)**: With lower metrics like accuracy (0.69) and AUC-ROC (0.75), SVM is better suited for linearly separable data but struggles with complex, non-linear datasets.
- **Decision Tree**: Provides a decent balance of accuracy (0.74) and recall (0.74), but its tendency to overfit can limit performance on unseen data. It is interpretable and useful for feature analysis.
- **Logistic Regression**: The lowest-performing model (accuracy, precision, and recall at 0.64), it works best with linearly separable data and struggles with non-linear relationships.
- **LSTM (Long Short-Term Memory)**: Outperforms other models in recall (0.90) and overall metrics, making it ideal for sequential data tasks. It leverages memory mechanisms to capture temporal patterns effectively.

# V. Evaluation

### *Classification Metrics*

- **AUROC**: Measures the model's ability to distinguish between classes, with higher values indicating better performance.
- **AUPRC**: Evaluates the balance between precision and recall, especially useful for imbalanced datasets.
- **F1 Score**: Combines precision and recall into a single metric, offering a balanced evaluation of false positives and false negatives.
- **Cohen's Kappa**: Assesses the agreement between predicted and actual classifications, adjusting for random chance.

### *Regression Metrics*

- **MAE**: Represents the average magnitude of errors between predicted and actual values, indicating model accuracy.
- **RMSE**: Penalizes larger errors more than MAE, providing insight into how well the model handles extreme values.

- **R-squared ($R^2$)**: Explains the proportion of variance in the dependent variable that can be predicted by the model, indicating overall fit.

## VI. Discussions

### A. Model Results

TABLE III: Comparison of Model Performance Metrics

| Model | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|
| LSTM | 0.82 | 0.77 | 0.90 | 0.83 |
| XGBoost | 0.80 | 0.81 | 0.80 | 0.80 |
| Random Forest | 0.79 | 0.80 | 0.79 | 0.79 |
| KNN | 0.76 | 0.77 | 0.76 | 0.76 |
| SVM | 0.69 | 0.72 | 0.70 | 0.71 |
| Decision Tree | 0.74 | 0.75 | 0.74 | 0.74 |
| Logistic Regression | 0.64 | 0.64 | 0.64 | 0.64 |

### B. LSTM

An LSTM classifier is built, and the baseline performance is evaluated which had an accuracy of 82.58% .The Gini Index is used to determine the important feature to be evapotranspiration, soil temperature between 0 cm to 7 cm, relative humidity and wind speed.

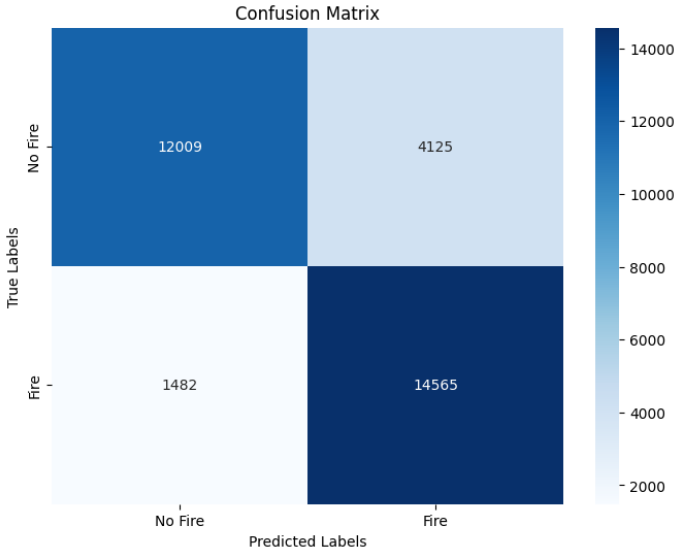


Fig. 21: Confusion matrix for the LSTM Model

A Receiver Operating Characteristic (ROC) curve closer to the top-left corner indicates better model performance, as it represents a higher true positive rate (TPR) and a lower false positive rate (FPR). For this model, the Area Under the Curve (AUC) is 0.9, which reflects excellent discriminatory ability and strong overall performance.

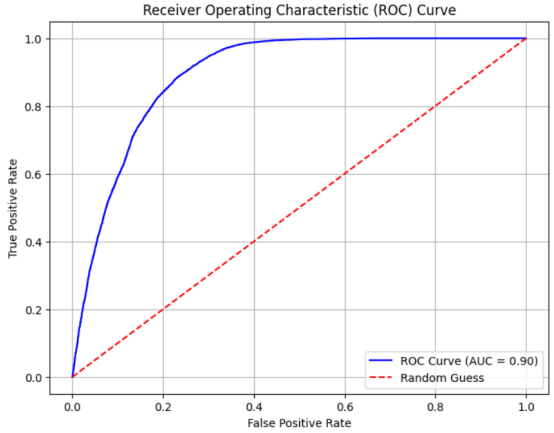


Fig. 22: AUC-ROC Curve for the LSTM Model

### C. Extreme Gradient Boosting - XGBoost

XGBoost is a machine learning algorithm based on gradient boosting, designed for both regression and classification. It builds decision trees sequentially, where each new tree tries to correct the errors made by previous ones. The final prediction is a combination of all the trees, helping provide high accuracy and robustness.

**Key Features**

1) **Regularization**: XGBoost supports L1 and L2 regularization to control the complexity of the model and reduce overfitting.
2) **Pruning**: XGBoost uses tree pruning to stop tree splitting when further splits do not improve performance. This improves the computational efficiency

For our baseline model, we performed 5-fold cross validation, achieving a mean accuracy of 75.2% accuracy. On the training data, the baseline model achieved an accuracy of 78.68%, but false positives were high on the test data, where the model incorrectly predicted wildfire occurrences in non-fire scenarios.

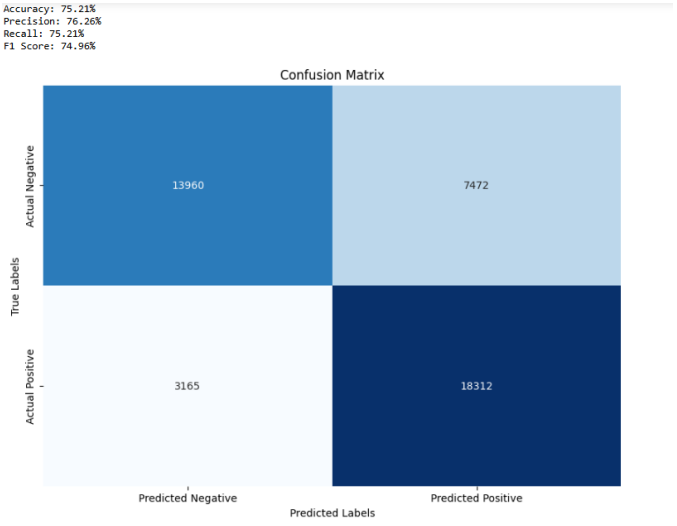


Fig. 23: XGB Baseline Metrics

**Hyperparameter Tuning with Bayesian Optimization**

To optimize the XGBoost model we used Bayesian Optimization, a hyperparameter tuning technique that efficiently explores the hyperparameter space by modeling the relationship between hyperparameters and the model's performance using a probabilistic surrogate function.

Unlike grid or random search Bayesian Optimization:

1) **Learns from the previous evaluations**: It identifies the most promising regions of the search space and focuses exploration there.
2) **Balances exploration and exploitation**: It iteratively evaluates hyperparameters to maximize performance while minimizing many unnecessary valuations

We utilized BayesSearchCV from the scikit-optimize[12] library, which combines Bayesian Optimization with cross-validation for hyperparameter tuning. The process started with random samples, and subsequent iterations used prior results to intelligently select new hyperparameters.

1) **learning_rate**: Determines the size of the steps during each boosting round. Lower values lead to slower learning, often leading to a more accurate model and higher values speed up the training but risk overfitting or missing the most optimal solution.
2) **n_estimators**: Specifies the number of boosting rounds (trees) in the model. More estimators allow the model to learn more but increase training time and risk overfitting. Fewer estimators reduce training time but might underfit the data.
3) **max_depth**: Maximum depth of each decision tree. Deeper trees (higher max_depth) allow the model to learn more complex patterns but can lead to overfitting. Shallower trees (lower max_depth) help prevent overfitting but might miss complex patterns.
4) **subsample**: Determines the fraction of training samples used to grow each tree. Lower values add randomness, reducing overfitting but possibly can underfit. A value of 1.0 uses all data, which can increase overfitting for smaller datasets.
5) **colsample_bytree**: Specifies the fraction of features or columns used when constructing each tree. Lower values reduce feature dependency which adds randomness and prevents overfitting. A value of 1.0 means all features will be considered, which might overfit the data.
6) **gamma**: Sets the minimum reduction in loss required to make a further split on a node. Higher values make the model more conservative, requiring larger reductions in loss before splitting. Lower values allow more splits, increasing flexibility but potentially leading to overfitting

After tuning, the model achieved a training accuracy of 95.49%. On the test dataset, the results demonstrated significant improvement over the baseline model:

1) **Accuracy**: Increased, reflecting better overall performance.
2) **True Positives**: Improved, indicating better detection of actual wildfire occurrences
3) **False Positives**: Still present but reduced, showing a better balance in predictions compared to the baseline

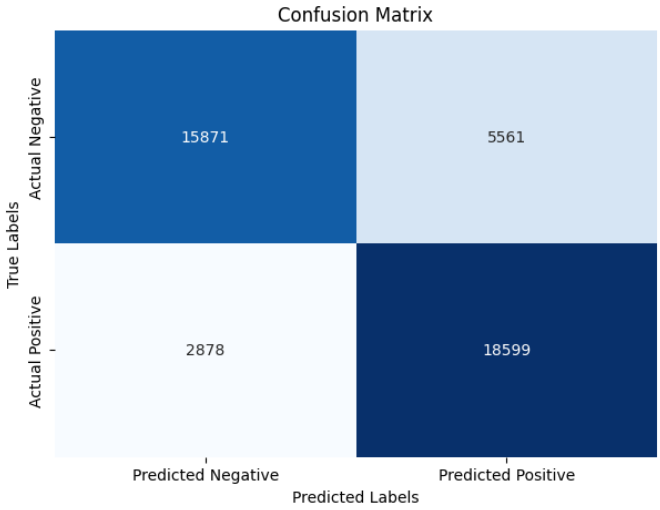


Fig. 24: XGB Model Confusion Matrix

However, the model is still slightly biased toward predicting wildfire occurrences, leading to occasional false positives. This behavior highlights a trade-off in the model's ability to avoid missing true positives versus minimizing false positives. We used the AUC-ROC score and the ROC curve to assess the model's performance. These metrics provide insights into the trade-offs between sensitivity (recall) and specificity across different classification thresholds, enabling a comprehensive evaluation of the model's predictive capability.

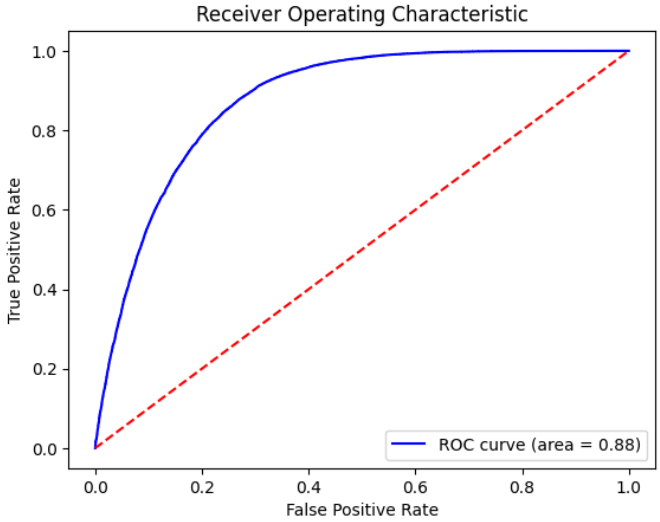


Fig. 25: XGB Model AUC ROC Curve

### *D. Random Forest*

The Random Forest Classifier was implemented to predict wildfire occurrences by leveraging an ensemble of decision trees. This model was selected for its robustness, ability to handle complex datasets, and resistance to overfitting. To further optimize its performance, hyperparameter tuning was conducted using Optuna[13], ensuring the best possible configuration.

**Key Steps and Features:**

- **Feature Engineering:** All features were scaled using *Standard Scaling* to ensure uniformity across the dataset, as variables like temperature, humidity, and precipitation had different units of measurement.
- **Feature Importance:** Gini Importance was calculated to evaluate the contribution of each feature to the model. The analysis revealed that key features such as evapotranspiration, soil temperature between 0 to 7 cm, and relative humidity were highly influential in predicting fire events. Features with lower importance, including cloud cover, weather code, rain, and snow, were identified but retained to observe their impact.
- **Cross-Validation:** A 5-fold cross-validation process was applied, achieving a mean accuracy of **79.3%**, reflecting strong generalization performance.
- **Hyperparameter Tuning:** The model was optimized using Optuna[13] to determine the best combination of parameters such as: n_estimators (number of trees in the forest), max_depth (maximum depth of the trees), min_samples_split (minimum samples required to split a node), min_samples_leaf (minimum samples required at a leaf node), and max_features (maximum features considered for splitting). The optimized model parameters improved the trade-off between bias and variance, ensuring better performance on unseen data.

**Model Performance:**

- The final model achieved an **accuracy** of **79.2%** on the test dataset.
- The Area Under the ROC Curve (AUC-ROC) was **0.87**, indicating strong discriminative ability between fire and non-fire events.
- The confusion matrix highlighted that true positives (actual fire events) were well-detected, although false positives persisted, likely due to similarities in environmental conditions between fire and non-fire records.

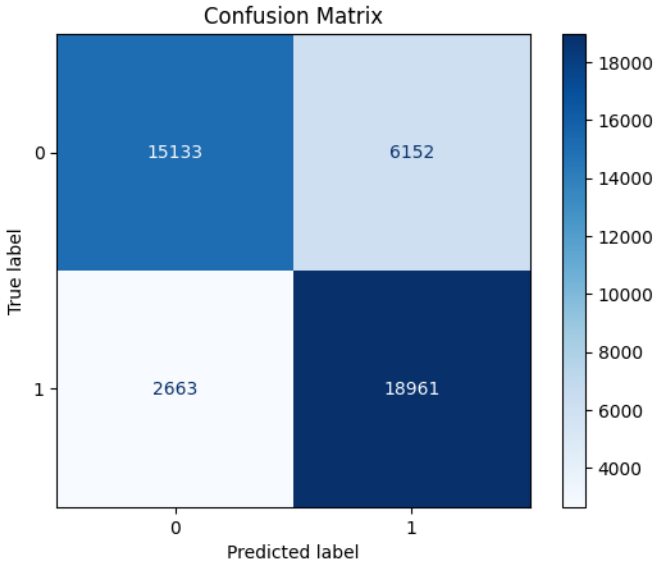


Fig. 26: Random Forest Model Confusion Matrix

**Visualizations:**

- A *Feature Importance* plot was generated, showcasing that evapotranspiration, soil temperature (0–7 cm), and relative humidity were the most significant predictors.
- The *Confusion Matrix* provided insights into the model's classification performance, showing the distribution of correct and incorrect predictions for fire and non-fire classes.
- The *ROC Curve* demonstrated the trade-off between the true positive rate (recall) and the false positive rate, with the curve leaning closer to the top-left corner, indicating strong performance.
- Although the model achieved high accuracy and AUC-ROC scores, a slight false-positive bias remained, which can be mitigated through further feature selection or threshold adjustments.

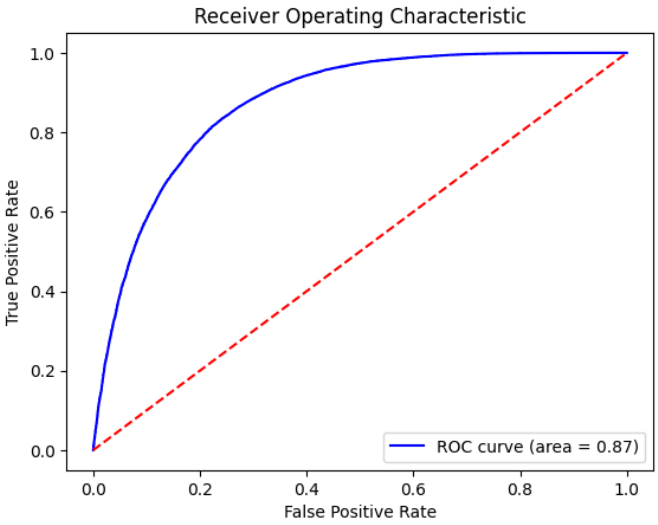


Fig. 27: Random Forest Model AUC ROC Curve

### *E. Support Vector Machine (SVM)*

The Support Vector Machine (SVM) model was implemented with the *Radial Basis Function (RBF)* kernel to classify wildfire occurrences. The data was preprocessed by handling missing values and standardizing the features using the `StandardScaler`. The training and test sets were created with an 80-20 split.

**Model Performance:** The model achieved a test accuracy of 69.96%, a precision of 64.85%, a recall of 86.61%, and an F1 score of 73.64%. This highlights the model's strong recall, which is crucial for wildfire detection tasks where minimizing false negatives is a priority.

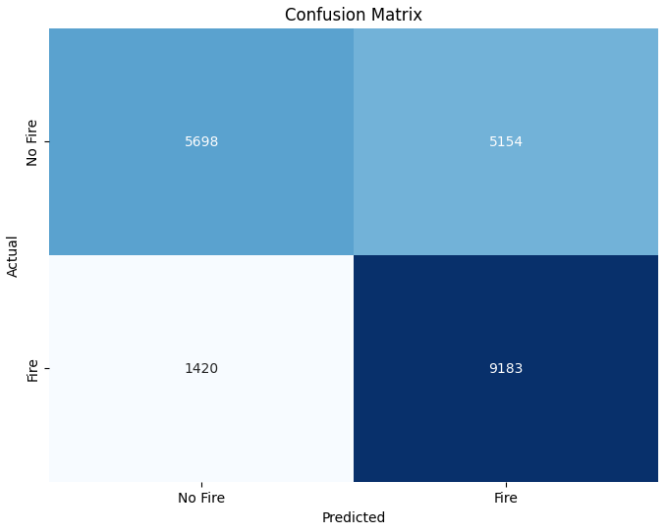


Fig. 28: Confusion Matrix for the SVM Model

The confusion matrix indicates that the model correctly predicted 9,183 instances of actual wildfires but misclassified 1,420 cases as "No Fire". Similarly, 5,154 instances were incorrectly predicted as "Fire" when no fire occurred.

**ROC Curve and AUC:** The Receiver Operating Characteristic (ROC) curve for the SVM model was plotted to evaluate its performance across varying thresholds. The Area Under the Curve (AUC) was found to be 0.75, demonstrating moderate discriminative power.

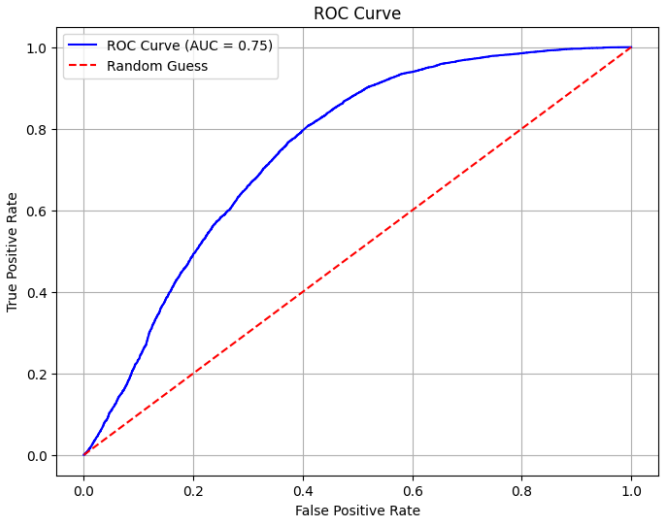


Fig. 29: ROC Curve for the SVM Model

**Key Observations:**

- The SVM model achieved a strong recall of 86.61%, making it effective at detecting wildfire occurrences.
- The model struggled with precision (64.85%), indicating a higher number of false positives, which may lead to unnecessary alerts.
- A trade-off between precision and recall is evident, as the model prioritizes sensitivity over specificity.
- The AUC-ROC score of 0.75 reflects that the model performs moderately well compared to other techniques.
- SVM's performance is sensitive to feature scaling, making preprocessing a critical step.

### *F. Decision Tree*

**Model Implementation:** A DecisionTreeClassifier was implemented and hyperparameter optimization was performed using GridSearchCV. The following hyperparameter ranges were explored:

- `criterion`: Gini, Entropy
- `max_depth`: [None, 10, 20, 30, 40, 50]
- `min_samples_split`: [2, 5, 10]
- `min_samples_leaf`: [1, 2, 4]

The optimal parameters found were: criterion being 'gini', max_depth of 20, min_samples_leaf of 1, min_samples_split of 2.

**Model Performance:**

- **Test Accuracy:** 74.19%
- **Precision:** 75%
- **Recall:** 74%
- **F1 Score:** 74%

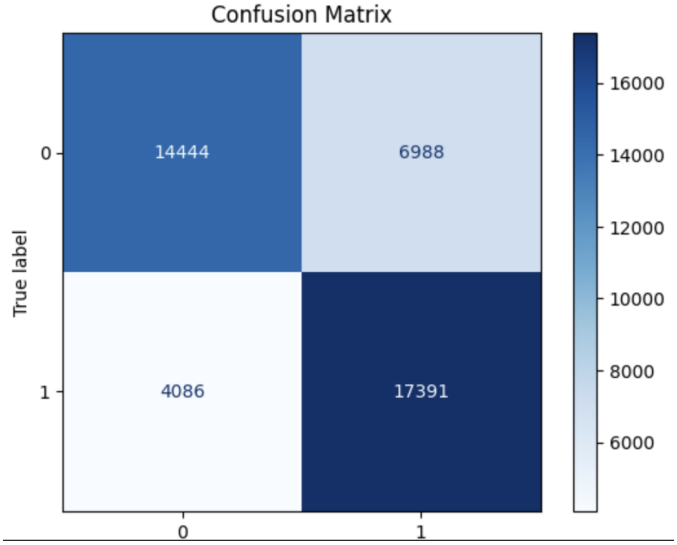


Fig. 30: Confusion Matrix for Decision Tree Model

**Feature Importance:** Gini importance scores revealed that features such as *soil temperature* and *soil moisture* were highly influential. Features contributing less than 2% to Gini importance, such as *cloud cover*, *rain*, and *snow*, were dropped during optional model optimization.

**Key Observations:**

- The decision tree achieved balanced performance with moderate precision and recall for both classes.
- Recall for class 1 (*fire*) was particularly strong (81%), making the model effective at minimizing false negatives.

- The model exhibited susceptibility to overfitting due to its reliance on specific training data splits.
- Gini importance scores aligned with intuitive expectations, although some features with apparent relevance (e.g., *rain* and *snow*) contributed minimally to the model.
- Hyperparameter tuning improved accuracy but had limited impact on the model's ability to differentiate between *fire* and *no-fire* instances.

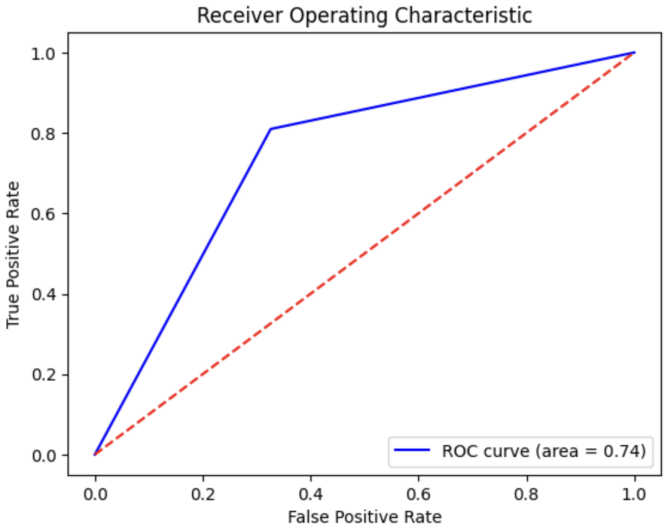


Fig. 31: AUC ROC Curve for Decision Tree Model

### G. K-Nearest Neighbors (KNN)

**Model Implementation:** A KNeighborsClassifier was implemented with the following configuration:

- `metric`: Manhattan
- `n_neighbors`: 1
- `weights`: Uniform

The model was trained on transformed features and evaluated on both training and testing datasets.

**Model Performance:**

- **Test Accuracy:** 75.98%
- **Precision:** 74.00%
- **Recall:** 81.00%
- **F1 Score:** 77.00%

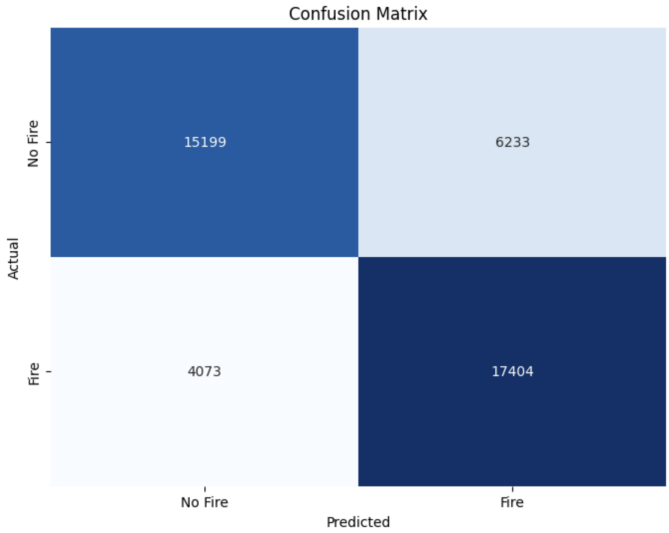


Fig. 32: Confusion Matrix for KNN Model

**Key Observations:**

- The model exhibited strong recall (81%) for class 1, minimizing false negatives and making it effective in scenarios where identifying positives is critical.
- The balanced accuracy, F1 score, and classification metrics suggest that the model performs well across both classes.
- The reliance on Manhattan distance and a single nearest neighbor could lead to computational inefficiencies for larger datasets.

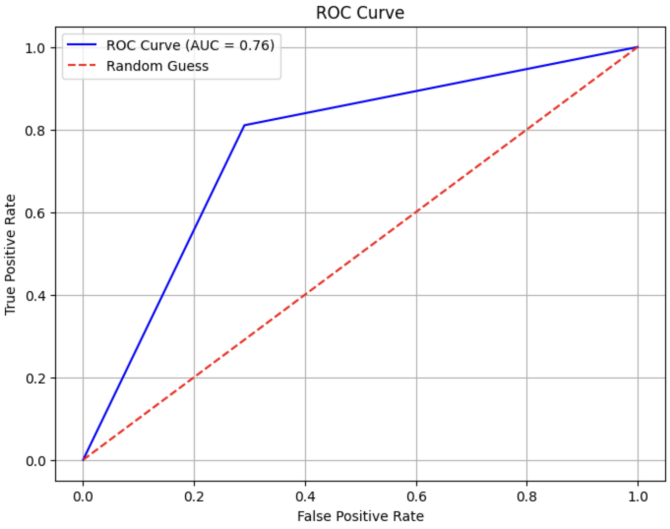


Fig. 33: ROC Curve for the KNN Model

### H. Logistic Regression

**Model Implementation:** A `LogisticRegression` model was implemented, and hyperparameter optimization was performed using `GridSearchCV` with the following parameter grid:

- `C`: [1, 10, 100]
- `penalty`: ['l1', 'l2']
- `solver`: ['liblinear']

The optimal parameters were: `C = 100`, `penalty = 'l2'`, and `solver = 'liblinear'`.

**Model Performance:** The logistic regression model produced results that were just slightly better than random guessing, with a test accuracy of 64%. The precision, recall, and F1 score were all the same, each at 64%, indicating a balanced but mediocre performance. These results suggest that the model is not effectively distinguishing between the two classes, and its performance is likely limited by the chosen features or the linear nature of the logistic regression algorithm in this case.

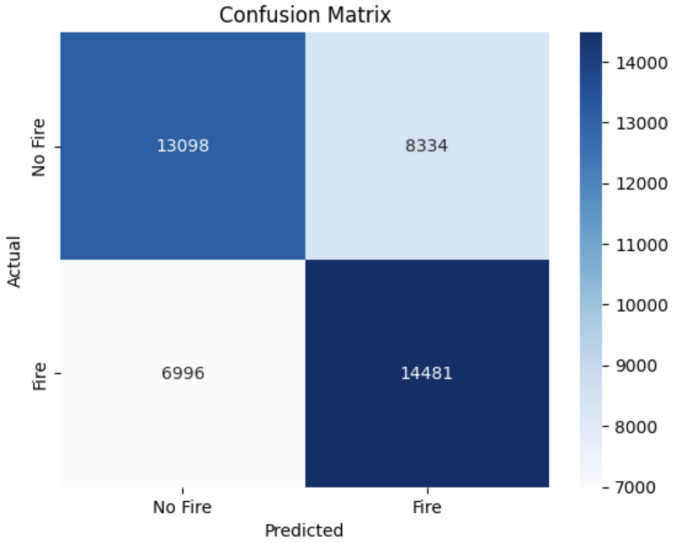


Fig. 34: Confusion matrix for Logistic Regression Model

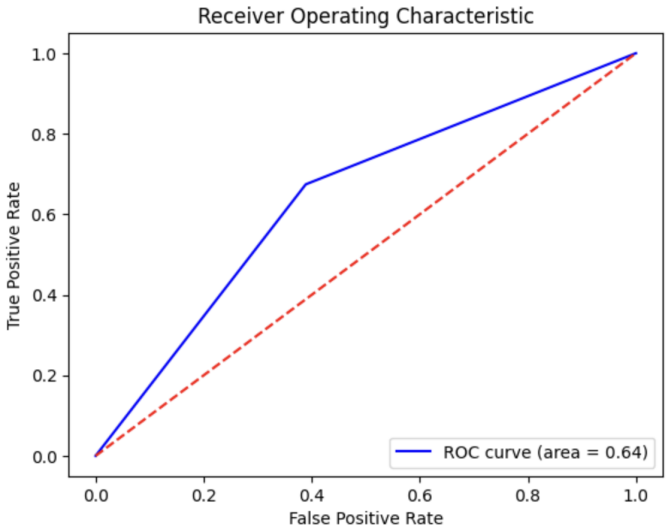


Fig. 35: ROC for Logistic Regression Model

**Lessons Learned:**

- Through LSTM, our project can be used as an early fire-warning system.
- Advanced models like LSTM and XGBoost perform significantly better in both accuracy and AUC-ROC compared to simpler models such as Logistic Regression and Decision Tree.
- There is a trade-off between metrics; while LSTM excelled in recall (0.90), it slightly lagged in precision compared to XGBoost, highlighting a potential trade-off.
- Models like Random Forest and XGBoost handle feature complexity better than simpler algorithms due to their ability to model interactions and non-linearities.
- Random Forest delivered consistent performance with a good trade-off between recall and precision.
- The SVM model performs reasonably well but lags behind advanced techniques like XGBoost and LSTM in overall accuracy and AUC scores.

**What Worked Well:**

- LSTM achieved the highest recall (0.90) and the best overall performance, making it suitable for tasks requiring high sensitivity.
- XGBoost and Random Forest delivered consistently strong performances across all metrics, showing their reliability for balanced datasets.
- AUC-ROC Performance: XGBoost and LSTM achieved high AUC-ROC scores (0.876 and 0.90, respectively), indicating strong model discriminative abilities.

**What Didn't Work:**

- Logistic Regression had the lowest scores across all metrics, struggling with the complexity of the data, indicating its limitations for non-linear relationships.
- SVM achieved only a 0.69 accuracy and a 0.75 AUC-ROC, making it less effective compared to tree-based models and neural networks.
- Simpler models like Decision Tree and KNN likely suffered from overfitting or an inability to generalize well to unseen data.

**Directions for Future Work:**

- The project further can be further expanded through convolutional LSTM using time-series dependent data like vegetation indices.
- Utilize explainability tools like SHAP or LIME to understand feature importance and improve model interpretability.

## VII. Conclusion

This study successfully developed an integrated wildfire risk prediction and monitoring system by leveraging multi-source data and advanced machine learning techniques. Through the use of models like LSTM, Random Forest, and XGBoost, the system achieved high performance, with LSTM attaining a recall of 0.90 and an AUC of 0.9, demonstrating its capability to accurately predict wildfire occurrences.

The key findings highlight the significance of temporal features, environmental conditions, and vegetation indices in wildfire prediction. This work lays a strong foundation for real-time monitoring, risk assessment, and early warning systems.

Future directions include incorporating convolutional LSTM architectures, integrating satellite imagery, and exploring interpretability techniques like SHAP to enhance the robustness and practical applicability of the system.

## REFERENCES


[1] R. T. Bhowmik, Youn Soo Jung, J. A. Aguilera, M. Prunicki, and K. Nadeau, "A multi-modal wildfire prediction and early-warning system based on a novel machine learning framework," Journal of Environmental Management, vol. 341, pp. 117908–117908, May 2023, doi: https://doi.org/10.1016/j.jenvman.2023.117908.

[2] Y. O. Sayad, H. Mousannif, and H. Al Moatassime, "Predictive modeling of wildfires: A new dataset and machine learning approach," Fire Safety Journal, vol. 104, pp. 130–146, Mar. 2019, doi: https://doi.org/10.1016/j.firesaf.2019.01.006.

[3] N. Agrawal, P. V. Nelson, and R. D. Low, "A Novel Approach for Predicting Large Wildfires Using Machine Learning towards Environmental Justice via Environmental Remote Sensing and Atmospheric Reanalysis Data across the United States," Remote Sensing, vol. 15, no. 23, pp. 5501–5501, Nov. 2023, doi: https://doi.org/10.3390/rs15235501.

[4] F. Huot et al., "Deep Learning Models for Predicting Wildfires from Historical Remote-Sensing Data," arXiv.org, 2020. https://arxiv.org/abs/2010.07445 (accessed Sep. 28, 2024).

[5] Shaddy, B., and Coauthors, 2024: Generative Algorithms for Fusion of Physics-Based Wildfire Spread Models with Satellite Data for Initializing Wildfire Forecasts. Artif. Intell. Earth Syst., 3, e230087, https://doi.org/10.1175/AIES-D-23-0087.1.

[6] Abdessemed, Ferdaous Bouam, Souheila Arar, Chafik. (2023). Forest Fire prediction using Machine Learning Methods: A Comparative Study. ResearchGate, May. 2023.

[7] Y. Ji, D. Wang, Q. Li, T. Liu, and Y. Bai, "Global Wildfire Danger Predictions Based on Deep Learning Taking into Account Static and Dynamic Variables," Forests, vol. 15, no. 1, pp. 216–216, Jan. 2024, doi: https://doi.org/10.3390/f15010216.

[8] D. Radke, A. Hessler, and D. Ellsworth, "FireCast: Leveraging Deep Learning to Predict Wildfire Spread," ResearchGate, Aug. 2019.

[9] NASA FIRMS. MODIS Collection 6 Hotspot / Active Fire Detections MCD14ML distributed from NASA FIRMS. Available online at https://earthdata.nasa.gov/firms. doi:10.5067/FIRMS/MODIS/MCD14ML

[10] NASA FIRMS. NRT VIIRS 375 m Active Fire product VJ214IMGGTDL_NRT distributed from NASA FIRMS. Available online at https://earthdata.nasa.gov/firms. doi:10.5067/VIIRS/VJ214IMG_NRT.002

[11] Zippenfenig, Patrick. Open-Meteo.com Weather API., Zenodo, 2023, doi:10.5281/ZENODO.7970649.

[12] Scikit-learn Developers, "Scikit-learn: Machine Learning in Python," Scikit-learn Documentation, 2023. Available online: https://scikit-learn.org/.

[13] Optuna Team, "Optuna: A Hyperparameter Optimization Framework," Optuna, 2023. Available online: https://optuna.org/.


## VIII. APPENDIX

By submitting this group report, we acknowledge adherence to the Honor Code Pledge:

On my honor as a University of Colorado Boulder student I have neither given nor received unauthorized assistance.

**Individual Contributions**

- Girish Jeswani: Data Collection, Random Forest Classifier and LSTM model
- Siddhant Kodolkar: Fire EDA and XG Boost Model
- Nishchal Shetty: Weather EDA and Support Vector Machine
- Darshan Vijayaraghavan: Decision Tree, K Nearest Neighbors and Logistic Regression models.