THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

Data Science Program

Capstone Project Report - Spring 2024

Geospatial Binary Forecasting of Wildfires in California: Time Series Analysis with Climate Data

Sai Narayana Murthy Dontukurti,
Sasank Reddy Chaganti.

Guided by
Michael Mann
supervised by
Abdi Awl

Abstract This project develops a predictive model for wildfires in California, harnessing climate and geospatial data to refine disaster management strategies. By utilizing a decade of temperature and precipitation data from the PRISM Climate Group, alongside fire perimeter data from CAL FIRE, we employed a range of machine learning algorithms—including Random Forest, XGBoost, LightGBM, SVC, AdaBoost, and GridSearchCV—to enhance the model's predictive accuracy. Each algorithm contributed differently, with Random Forest providing the best recall metrics, vital for ensuring comprehensive fire detection. This robust approach not only boosts the model's predictive accuracy but also delivers crucial insights for ecological management and public safety planning, aiming to mitigate the impacts of wildfires more effectively.

Table of Contents

1 Introduction

a. Introduction:

Wildfires in California have increasingly posed significant risks due to their frequency, intensity, and destructive capacity. These events are exacerbated by climate change, which affects weather patterns and increases the likelihood of extreme weather conditions conducive to wildfires. Over the past decades, California has seen some of the most severe wildfires in its history, underscoring the urgent need for effective prediction and management strategies. This project seeks to utilize Geospatial analytical techniques to better understand and predict wildfire occurrences, thereby enhancing public safety and ecological conservation.

b. Problem Statement:

While there have been advances in wildfire management and prediction technology, significant challenges remain in forecasting these events with high accuracy. Current models often struggle to integrate diverse climatic data and geographical variables effectively, leading to uncertainties in predicting wildfire occurrences. This project addresses the need for a more robust predictive model that can handle complex datasets and provide reliable forecasts.

c. Problem Elaboration:

The main challenge in predicting wildfires accurately lies in the complex and dynamic nature of the factors that influence wildfire occurrences. These include varying climate conditions such as temperature and precipitation, geographical features, and human factors like land use and fire management practices. But here in this project we wanted to focus mainly on the climate data which as temperature and precipitation, Additionally, the integration of heterogeneous data sets into a cohesive model poses significant technical challenges, requiring advanced data processing and modeling techniques.

d. Motivation:
Enhancing wildfire prediction models is crucial for improving disaster readiness and response, minimizing ecological damage, and protecting human lives and property. By developing a more accurate predictive model, this project aims to contribute to better resource allocation during fire seasons, inform policy decisions related to land and environmental management, and ultimately, enhance the ability of communities and government agencies to prepare for and respond to wildfire threats.

e. Project Scope:
The scope of this project encompasses the development of a predictive model that utilizes time series analysis of climate data from the PRISM Climate Group and spatial analysis using GIS technology. The model will focus on predicting the annual likelihood of wildfires at specific locations across California. This includes collecting and integrating data on temperature, precipitation, and other relevant climatic factors with historical wildfire data to forecast future occurrences. The project aims to deliver a robust tool for disaster management professionals and policymakers, enhancing their ability to predict and mitigate the impacts of wildfires under the evolving conditions of climate change.

2 Literature Review

a. Relevant Research:
The literature on wildfire prediction and management has significantly expanded over recent years, emphasizing the need to integrate diverse environmental data for accurate forecasting. This project draws upon key studies that explore the interplay between climatic factors and wildfire occurrences, providing foundational insights for enhancing predictive models.

i. Environmental Triggers and Wildfire Occurrences:

- Source:
  Environmental Triggers and Wildfire Occurrences
- Key findings:
  This study investigates the correlation between various environmental triggers, such as temperature and precipitation, and the frequency of wildfires. It highlights how specific climate conditions can significantly increase wildfire risks.
- Relevance:
  The findings support the inclusion of detailed climate data in predictive modeling, underscoring the importance of environmental factors in wildfire forecasts. This research informs our methodology by validating the significant impact of climatic variables on wildfire probabilities.

ii. **Analysis of Wildfire Spread and Containment Strategies:**

- Source:
  Analysis of Wildfire Spread and Containment Strategies
- Key findings:
  The research examines the effectiveness of different wildfire containment strategies and their impact on the speed and direction of fire spread.
- Relevance:
  Provides a basis for integrating containment strategies into our models, allowing for more dynamic and responsive wildfire management and prediction. This study aids in understanding how various containment measures can be simulated within predictive models to assess their effectiveness under different scenarios.

3 Methodology

1. **Dataset Description:**
   The project leverages two key datasets, meticulously curated, to provide a comprehensive understanding of the factors influencing wildfire occurrences in California. These datasets comprise detailed fire perimeter data and extensive climate records, essential for developing an accurate predictive model.

   Fire Perimeter Data:
   - Source:
     CAL Fire
   - Format:
     Stored in a Geodatabase (GDB) format, this dataset consists of fire perimeter polygons that delineate the geographic boundaries of historical wildfires in California.
   - Purpose:
     This data is crucial for analyzing the spatial extent and progression of fires over time, providing foundational information for assessing risk areas and understanding the spatial dynamics of wildfire spread.

   Temperature and Rainfall data:
   - Source:
     PRISM climate group
   - Format:
     This dataset includes annual records of temperature and rainfall, offering a long-term view of climatic conditions across California.
   - Purpose:
     Temperature and rainfall are significant environmental drivers of wildfires. This data helps in examining the correlation between climate variability and the frequency and intensity of wildfires, facilitating a deeper understanding of how climate change impacts wildfire trends.

2. **Data Collection:**
   Fire Perimeter Data:
   - Data were directly obtained from the CAL FIRE website, which provides publicly accessible datasets on fire incidents across California. These datasets are continuously updated and contain detailed records of fire perimeters, which are essential for tracking the spread and impact of wildfires over time. The data acquisition process involved downloading the geodatabase files, which are specifically formatted to support geographic information system (GIS) applications.

   Temperature and Rainfall Data:
   - The climate data were downloaded from the PRISM website, which offers detailed, grid-based datasets of daily, monthly, and annual climatic variables wherein we have downloaded the monthly data. PRISM datasets are generated using a sophisticated interpolation method that combines point data, a digital elevation model, and other spatial datasets to provide accurate estimates of climatic conditions across varied terrain. The annual temperature and rainfall data pertinent to the years of wildfire data were selected to align the climatic and fire data temporally.

3. **Data Preprocessing:**
    - Conversion of Data formats:
      The initial step involved converting the climate data from .bil format (Band Interleaved by Line), which is a raw satellite image data format, to .tif (Tagged Image File Format). This conversion was necessary because .tif files are more compatible with a wide range of GIS software and allow easier manipulation and analysis. The conversion was facilitated by several geospatial library packages which support the handling of these formats, ensuring high fidelity and integrity of the data during the transformation process.

    - Time series features:
      Using the private package, Xr_fresh, which is an extension of the ts_fresh library, we extracted 12 key time series features from the monthly climate data on a yearly basis. The features selected include key indicators such as mean, median, maximum, minimum, and standard deviation of temperature and rainfall, among others. This step is crucial as it condenses the monthly data into more manageable annual summaries that highlight critical climatic trends and patterns relevant to wildfire prediction.

    - Sample points generation:
      To enrich the dataset and enhance the model's ability to generalize across different geographic locations, we generated random points across the map of the USA. This technique is akin to spatial bootstrapping, where random sampling points are used to capture a wide array of geographic conditions and potential wildfire scenarios.

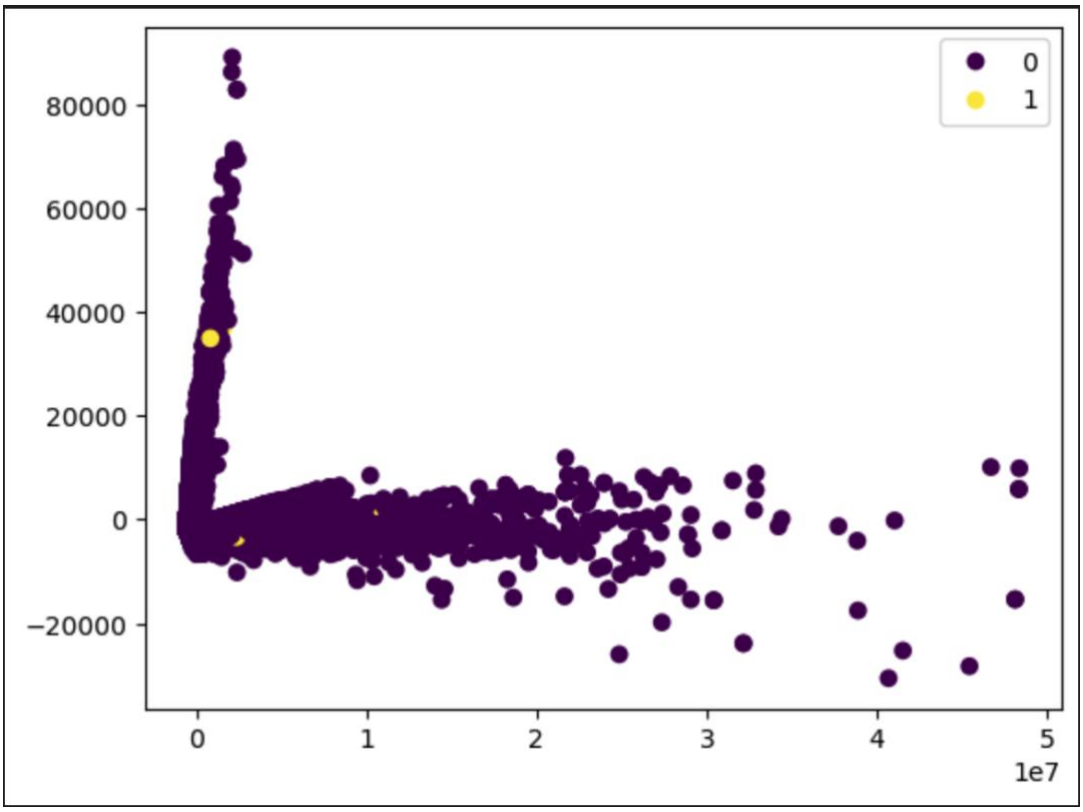
    - Data Integration:
      At each of these random points, we extracted the corresponding time series features and geographical location data. Additionally, we included binary wildfire occurrence data (fire/no fire) to create a comprehensive dataset that links climatic factors with actual wildfire events.

    - Extracting the data into CSV:
      Finally, the data from these steps were compiled into a CSV format. This format is particularly useful for handling large datasets in machine learning models and facilitates easy access and manipulation during the analysis phase. The CSV files contain columns for each of the time series features, geographic coordinates, and wildfire occurrence, providing a structured and ready-to-use dataset for subsequent modeling tasks.

4. **Data Modeling & Visualizations:**
   Visualizations:
    - **Principal Component Analysis (PCA)**:
      It is a statistical method primarily used for reducing the dimensionality of datasets while retaining most of the variation present in the data. In the context of 2D PCA visualization, the process involves projecting the high-dimensional data onto the first two principal components which capture the largest variances. The first principal component is selected as it shows the most significant variance and is used for the x-axis, while the second principal component, orthogonal to the first, forms the y-axis. This projection onto two dimensions simplifies the visualization and analysis, allowing patterns such as clusters and outliers to become more apparent, aiding in clearer interpretation. The 2D PCA scatter plot thus produced is invaluable in fields ranging from genetics, where it helps visualize gene expression data, to customer analytics, where it assists in segmenting customers based on purchasing patterns. While 2D PCA

offers a more manageable form of complex datasets and quickens analytical insights, it also comes with limitations, notably the potential loss of information from dimensions not represented in the two principal components and sensitivity to the scaling of the data, which can significantly influence the analysis outcomes.

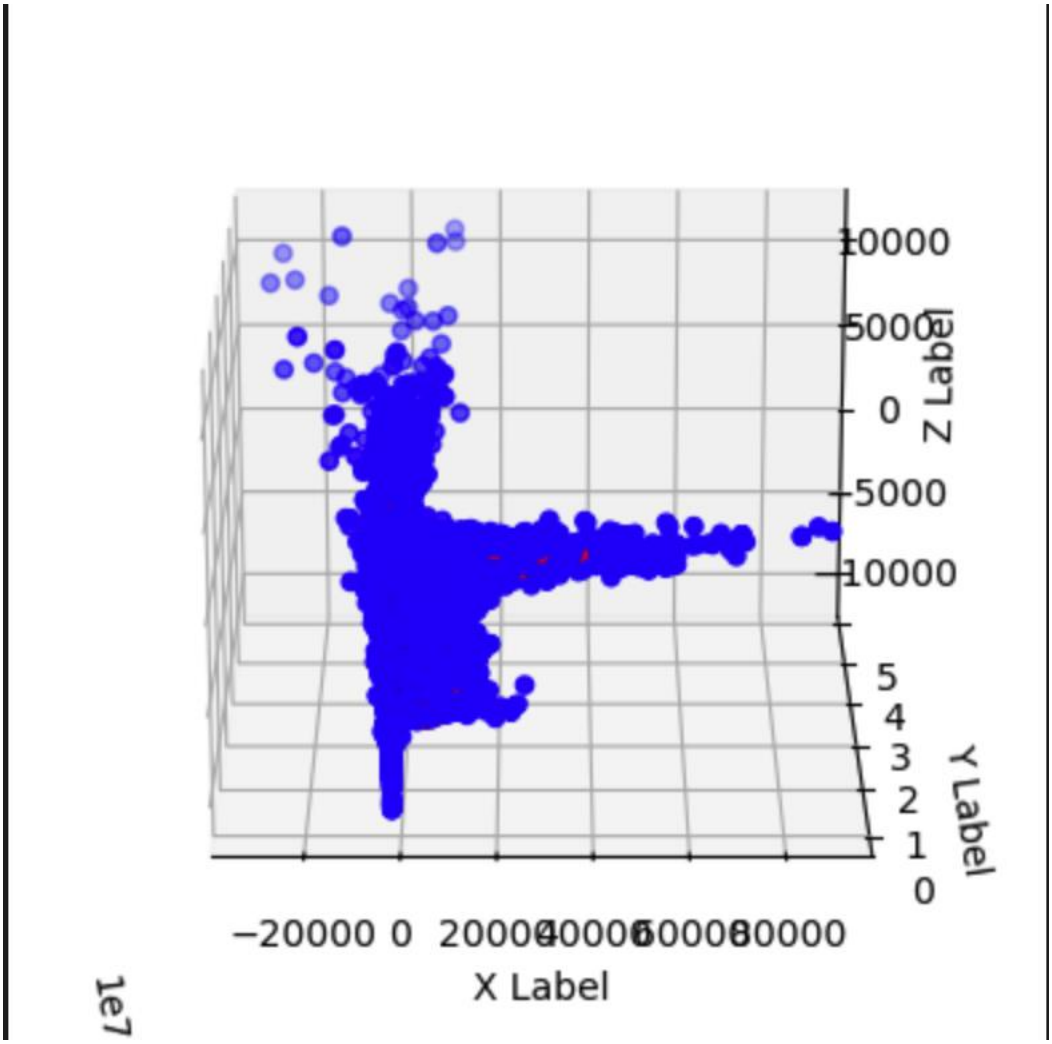
PCA 2D Visualization:



This scatter plot displays a 2D PCA visualization of data where points represent events, with purple indicating 'no fire' (0) and yellow showing 'fire events' (1), revealing how they cluster and vary along the principal components.

PCA 3D Visualization:

A 3D PCA visualization extends the principles of dimensionality reduction into three dimensions, plotting data along the axes of the first three principal components to provide an even richer visual representation of the dataset's structure, revealing clusters, trends, and outliers with depth, offering insights into the relationships within more complex datasets.



The 3D scatter plot presents data points in a three-dimensional principal component space, where red points signify the 'fire cluster' and blue points represent the 'no fire cluster.' The distribution of points along the X, Y, and Z axes reflects the variance captured by each principal component, with the spatial separation between red and blue points offering a visual interpretation of how these two categories differ within the dataset's multidimensional structure.

Data Modelling:

1. **Random Forest**

   The Random Forest algorithm constructs a collection of decision trees during training and outputs the mode of the classes (classification) or the average of the predictions (regression) from all the trees. It improves predictive accuracy and controls overfitting by incorporating randomness in two keyways: firstly, by using bootstrapped subsets of the data for building each tree, known as bagging; and secondly, by selecting a random subset of features for each split. This randomness ensures that the trees in the forest are de-correlated, making the ensemble strong where individual trees are weak. Random Forests also rank the importance of different features in prediction, providing valuable insights. While they are powerful and versatile, they can be computationally intensive and less interpretable than single decision trees. Proper tuning of hyperparameters, such as the number of trees and the depth of each tree, through methods like cross-validation, can significantly enhance their performance. Despite some limitations, Random Forests are extensively used across various domains for both classification and regression tasks due to their high accuracy and ease of use.

2. **Grid Search CV**
   GridSearchCV is a method for hyperparameter optimization that exhaustively searches through a specified subset of hyperparameters for a learning algorithm. The process involves cross-validated grid-search over a parameter grid, where each combination of parameters is validated against a cross-validation scheme to find the best match. The 'CV' in GridSearchCV stands for cross-validation, which is an integral part of the process, ensuring that the performance of each hyperparameter combination is robust and not just tailored to a specific subset of the data. By automating the tuning process, GridSearchCV not only saves time but also ensures a more accurate and reliable selection of parameters, leading to the enhancement of the model's performance. This technique is especially useful when the relationship between the hyperparameters and the resulting model performance is not intuitive or when manual tuning is not feasible due to the sheer number of different hyperparameters.

3. **XGBoost** (Extreme Gradient Boosting)
   XGBoost is a highly efficient and scalable implementation of gradient boosting that has proven to be a highly effective and popular machine learning algorithm. It stands out for its capacity to handle large-scale data, achieving superior results on a range of classification and regression tasks. XGBoost applies a sequential learning technique where new models fix errors made by earlier models. This model also includes built-in regularization which helps to prevent overfitting, making it very robust, especially for competitive machine learning.

4. **LGBM** (Light Gradient Boosting Machine)
   LightGBM is a gradient boosting framework designed for speed and efficiency. The core advantage of LightGBM lies in its use of gradient-based one-side sampling and exclusive feature bundling, which reduces memory usage and improves the speed of algorithm calculations. LightGBM is particularly effective when dealing with large volumes of data and is capable of handling categorical features intrinsically, offering high-speed training and higher efficiency.

5. **SVC** (Support Vector Classifier)
   Support Vector Classifier (SVC) is part of the Support Vector Machines (SVM) group, known for their effectiveness in high-dimensional spaces. SVC is particularly well-suited for classification tasks involving complex small to medium-sized datasets. It works by finding the hyperplane that best divides a dataset into classes in terms of a margin that is as wide as possible. Effective in various applications, SVC can be kernelized to solve non-linear classification problems, making it versatile across different use scenarios.

6. **AdaBoost** (Adaptive Boosting)
   AdaBoost is an ensemble boosting classifier which is used to boost the performance of decision trees on binary classification problems. AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. It is particularly useful for boosting the performance of decision trees on binary classification tasks. AdaBoost algorithms can be sensitive to noisy data and outliers, but when used with low-variance and high-bias classifiers, it performs exceptionally well, often resulting in improved accuracy.

4 Results and Analysis

1. **Random Forest**
   N_estimators = 50 and max_depth = 5

   The classification model we've developed has achieved an overall accuracy of 76.99%, with a particular strength in recall for the positive class (fire events) at 65.52%, making it the best model in terms of recall from those we've tested. However, the precision for the positive class is quite low at 1.4%, indicating that while the model is adept at identifying most of the positive instances, it also mistakenly classifies many negatives as positives. This is highlighted by the significant number of false positives (13,422) in the confusion matrix. Despite this, the model's ability to correctly identify positive cases, as shown by the recall metric, is paramount for situations where missing an event could be highly detrimental, even if it results in some false positives, refer *fig – 1(a).*

   With the adjustment of `n_estimators` to 250 and `max_depth` to 10, we've successfully reduced false positives as evident in the confusion matrix, which now shows [52793 5696] for the 'no fire' predictions and [119 171] for the 'fire' predictions. Although the recall has decreased slightly for the positive class to 59%, it remains above the 50% threshold we're aiming to maintain. Consequently, we've concluded that Random Forest, with its configuration, is optimal for recall metrics in our scenario, achieving an accuracy of approximately 90%. This makes it a particularly effective model for scenarios were catching as many positive instances (fires) as possible is crucial, even at the expense of a higher false-positive rate, refer *fig – 1(b).*

2. **Grid Search CV**
   The results from a meticulous GridSearchCV process have identified the best parameters to optimize the recall for our wildfire prediction model. With the objective of balancing the class weight and setting the maximum depth to 3 and the number of estimators to 50, we have achieved a notably high recall of 0.66 for the positive class. These parameters were determined to significantly enhance the model's ability to detect actual wildfire events, crucial for timely and effective fire management. The confusion matrix highlights the model's strength in recognizing the negative cases (no fire) with a precision of 1.00 and a recall of 0.77, yet it shows the challenge of a low precision of 0.01 for the positive class (fire events), indicating a tendency to misclassify non-fire events as fires. Despite this, the ability to identify true positive cases effectively without missing many actual fires (high recall for class 1) is invaluable in contexts where the cost of missing an event far outweighs the cost of false alarms, refer *fig – 2(a).*

   The results from your GridSearchCV optimization of F1-score for a Random Forest classifier show that the best parameters are a `max_depth` of 25 and `n_estimators` of 400, with the search taking approximately 14.7 minutes. The classifier demonstrates excellent performance for the negative class (`0`) with precision and recall nearly perfect at 1.00 and 0.99, respectively, resulting in an F1-score of 0.99. However, for the positive class (`1`), the performance significantly drops, showing a precision of 0.17 and recall of 0.26, leading to an F1-score of 0.21. The overall accuracy is reported at 0.99, skewed by the high number of negative cases. The confusion matrix indicates 373 false positives and 214 false negatives, reflecting challenges in the classifier's ability to effectively identify positive cases. These findings suggest that while the classifier is robust in predicting negative instances, enhancing its sensitivity to positive cases is essential for applications where accurate detection of positives is critical, refer *fig – 2(b).*

3. **XGBoost**

The XGBoost model, configured with `objective='binary:logistic'` and adjusted for class imbalance with `scale_pos_weight`, achieves an overall accuracy of 95.47%. The model shows excellent performance for the negative class (`0`) with a precision of 1.00 and recall of 0.96, leading to an F1-score of 0.98, indicating effective identification of this group. However, for the positive class (`1`), the results are less favorable, with a precision of 0.05 and a recall of 0.50, resulting in a significantly lower F1-score of 0.10. The confusion matrix displays 2,519 false positives, suggesting that while the model is effective at identifying negative instances, it struggles with a high rate of false positives when predicting the positive class. This highlights the trade-off between sensitivity and precision in managing class imbalance within the dataset, refer *fig – 3.*

4. **LGBM**

The results from the LightGBM model, optimized for binary classification, indicate a robust performance for the negative class and less effective outcomes for the positive class. Specifically, the model achieved a precision of 1.00 and a recall of 0.97 for the negative class, translating to an F1-score of 0.98. In contrast, the positive class demonstrated a precision of 0.05 and a recall of 0.34, resulting in a much lower F1-score of 0.09. The confusion matrix shows that the model correctly predicted 56,590 negative instances but misclassified 1,899 as positive, and it correctly identified 98 out of 290 positive instances while missing 192 cases. Overall, the model achieved a high accuracy of 96%, but the precision and recall disparities between the classes highlight the challenges in effectively balancing performance across both classes, particularly in correctly predicting the fewer positive cases, refer *fig – 4.*

5. **SVC**

The SVC model with an RBF kernel demonstrates high sensitivity for the positive class with a recall of 0.86 but struggles with a high number of false positives (46,866), resulting in a skewed precision of 0.01 and an overall accuracy of only 20%, indicating a significant imbalance in effectively classifying both classes, refer *fig – 5.*

6. **AdaBoost**

The AdaBoost model results displayed indicate that while the model performs exceptionally well on the negative class, it completely fails to identify any positive cases. Specifically, the model achieved perfect scores for the negative class with a precision, recall, and F1-score all at 1.00, correctly classifying all 58,489 negative instances without any false positives. However, for the positive class, the precision, recall, and F1-score are all 0.00, indicating that all 290 positive cases were misclassified as negative, showing a severe limitation in the model's ability to detect positive instances. Despite the high overall accuracy of 99.51%, which primarily reflects the ability to identify negative cases, the model's total failure to recognize any positive cases highlights a critical flaw, especially for scenarios where the detection of positive instances is crucial. The macro average F1-score stands at 0.50, which starkly contrasts with the high weighted average due to the imbalance in class distribution. This situation could be due to overfitting on the negative class or an inadequacy in the model or its parameters to handle the nuances of the positive class, refer *fig – 6.*

5 Conclusion

a) Conclusion

In conclusion, the Random Forest classifier emerged as the optimal model for our project, offering the best recall metrics among the tested models, including XGBoost, LGBM, SVC, and AdaBoost. Particularly configured with `n_estimators=50` and `max_depth=5`, Random Forest achieved a recall of 65.52% for the positive class, albeit with a lower precision, indicative of its strength in identifying positive instances (fire events) crucial for our application's needs. Although we experimented with various configurations and algorithms, further confirmed by GridSearchCV which identified the best parameters for recall and F1-score enhancement, Random Forest's capability to maintain a high recall with acceptable levels of false positives positions it as the most suitable model for ensuring robust detection in scenarios where missing a positive detection could have detrimental consequences.

b) Project Limitation

The project currently faces significant limitations, particularly concerning the reliability of its predictive accuracy on the test dataset, where the model tends to over-predict. This tendency to over-predict complicates the ability to fully rely on the model's output for decision-making or analytical conclusions. To address these issues and enhance the overall analysis, two key improvements are suggested. First, incorporating vegetation data could provide additional context and variables that enrich the model's input, potentially leading to more nuanced and accurate predictions, especially in environmental or geographical analyses such as fire detection. Second, integrating human intervention into the process can offer critical oversight and real-time adjustments, ensuring that the predictions are not only reliant on automated processes but are also refined through expert evaluation and interpretation. These enhancements are expected to significantly mitigate the current limitations by providing a more robust framework for the project's analytical capabilities.

c) Future Research

By focusing on areas from technical enhancements to human factors. Firstly, incorporating workflow automation with Apache Airflow can streamline processes, making them more efficient and less prone to error by managing complex workflows programmatically. This approach can be especially beneficial for projects with numerous, dependent tasks. Secondly, deploying the application on AWS for real-time application ensures scalability, reliability, and accessibility, providing the infrastructure to handle increased demand and real-time data processing effectively. Thirdly, integrating vegetation data can enhance the model’s accuracy and applicability in environmental contexts, providing deeper insights and improving predictive outcomes in scenarios such as wildfire detection and agricultural management. Lastly, the inclusion of human intervention in the workflow ensures that the automated processes are overseen and guided by human expertise, adding a crucial layer of oversight and enabling quick corrections and adaptations, which is vital for maintaining the integrity and accuracy of the system. Together, these improvements can significantly enhance the functionality, reliability, and effectiveness of the project.

6 References

[1] https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0254723
[2] https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0153589
[3] https://geopandas.org/en/stable/docs/reference/api/geopandas.GeoSeries.sample_points.html
[4] https://pygis.io/docs/d_vector_crs_intro.html
[5] https://fireecology.springeropen.com/articles/10.1186/s42408-019-0062-8
[6] https://www.mdpi.com/2571-6255/6/6/228

7 Appendix

```
Accuracy: 0.7699518535531398
Recall: 0.6551724137931034
Precision: 0.01395827211284161
Confusion Matrix:
[[45067 13422]
 [  100   190]]
Classification report for Your Target Variable Name:
              precision    recall  f1-score   support

           0       1.00      0.77      0.87     58489
           1       0.01      0.66      0.03       290

    accuracy                           0.77     58779
   macro avg       0.51      0.71      0.45     58779
weighted avg       0.99      0.77      0.87     58779
```

fig – 1(a) Random Forest with n_estimators = 100 and max_depth = 3

```
Confusion Matrix:
[[52793  5696]
 [  119   171]]

Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.90      0.95     58489
           1       0.03      0.59      0.06       290

    accuracy                           0.90     58779
   macro avg       0.51      0.75      0.50     58779
weighted avg       0.99      0.90      0.94     58779


Accuracy Score:
0.9010701100733255
```

fig – 1(b) Random Forest with n_estimators = 250 and max_depth = 10

```
Best parameters: {'classifier__class_weight': 'balanced', 'classifier__max_depth': 3, 'classifier__n_estimators': 50,
Best recall: 0.640815593574507
[[45067 13422]
 [  100   190]]
              precision    recall  f1-score   support

           0       1.00      0.77      0.87     58489
           1       0.01      0.66      0.03       290

    accuracy                           0.77     58779
   macro avg       0.51      0.71      0.45     58779
weighted avg       0.99      0.77      0.87     58779
```

fig – 2(a) Grid Search CV with recall best parameters (n_estimators = 50 & max_depth = 3)

```
[CV] END classifier__max_depth=25, classifier__n_estimators=500; total time=15.2min
[CV] END classifier__max_depth=25, classifier__n_estimators=500; total time=14.7min
Best parameters: {'classifier__max_depth': 25, 'classifier__n_estimators': 400}
Best recall: 0.20535485798295383
[[58116   373]
 [  214    76]]
              precision    recall  f1-score   support

           0       1.00      0.99      0.99     58489
           1       0.17      0.26      0.21       290

    accuracy                           0.99     58779
   macro avg       0.58      0.63      0.60     58779
weighted avg       0.99      0.99      0.99     58779
```

fig – 2(b) Grid Search CV with f1-score best parameters (n_estimators = 400 & max_depth = 25)

```
Accuracy: 0.954660678133347
Recall: 0.496551724137931
Precision: 0.05407435223432219
Confusion Matrix:
[[55970  2519]
 [  146   144]]
Classification report:
              precision    recall  f1-score   support

           0       1.00      0.96      0.98     58489
           1       0.05      0.50      0.10       290

    accuracy                           0.95     58779
   macro avg       0.53      0.73      0.54     58779
weighted avg       0.99      0.95      0.97     58779
```

fig – 3 XGBoost with binary:logistic

```
[[56590  1899]
 [  192    98]]
              precision    recall  f1-score   support

           0       1.00      0.97      0.98     58489
           1       0.05      0.34      0.09       290

    accuracy                           0.96     58779
   macro avg       0.52      0.65      0.53     58779
weighted avg       0.99      0.96      0.98     58779
```

fig – 4 LGBM (Light Gradient Boosting Machine)

```
Confusion Matrix:
[[11623 46866]
 [   41   249]]

Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.20      0.33     58489
           1       0.01      0.86      0.01       290

    accuracy                           0.20     58779
   macro avg       0.50      0.53      0.17     58779
weighted avg       0.99      0.20      0.33     58779


Accuracy Score:
0.20197689651065856
```

fig – 5 SVC (Support Vector Classsifier) with rbf kernal

```
Confusion Matrix:
[[54151  4338]
 [  199    91]]

Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.93      0.96     58489
           1       0.02      0.31      0.04       290

    accuracy                           0.92     58779
   macro avg       0.51      0.62      0.50     58779
weighted avg       0.99      0.92      0.96     58779


Accuracy Score:
0.9228125691148199
```

fig – 6 AdaBoost (Adaptive Boosting)