CHAPTER 1

INTRODUCTION

1.1 Introduction

As a global hotspot for wildfires, California in the United States has significant reference value for wildfire management measures worldwide. The frequent fires there are closely related to global climate change, human activities, and insufficient government control.

Therefore, we conducted a research and analysis on this issue. This analysis is based on the publicly available wildfire data of the US government over the past decade, combined with data mining and machine learning techniques, and comprehensively considers the influences of meteorological conditions, geographical environment, and socio-economic factors to analyze the spatiotemporal distribution characteristics and risk driving mechanisms of wildfires. The research found that climatic conditions such as drought indices and Santa Ana winds are the main direct factors causing wildfires. At the same time, environmental changes caused by human activities have weakened the resilience of natural ecosystems to some extent, further increasing the probability of wildfires.

Based on the above results, this study constructed a wildfire risk prediction model based on data analysis and proposed the idea of social collaborative governance and policy recommendations for addressing climate change, aiming to provide certain references for related work.

1.2 Problem Background

In recent years, with the intensification of global climate change and the continuous expansion of human activities, the frequency of extreme weather events has increased. Wildfires are not only an important natural disaster problem faced by the United States, but also by all regions of the world. Based on the wildfire data released by the US government, this paper selects California as a typical case for analysis. Due to its unique geography, long-term high temperature and drought, and monsoon climate conditions, coupled with human activities such as population growth, urban expansion, and land pollution, the risk level of wildfires in California has continued to rise, and it has eventually become one of the most frequent wildfire areas in the world, bringing huge ecological damage and hundreds of millions of economic losses to California.

The frequent occurrence of wildfires not only destroys the local ecosystem, but also poses a certain threat to the entire earth's ecology. Therefore, how to scientifically control wildfire risks, predict in advance, and take effective prevention and response measures should not only be a problem that the United States should pay attention to, but also the whole world. In this context, data science provides new technical means for wildfire risk control and analysis. By integrating historical fire records, meteorological data, geological information and other data, and then using data mining and machine learning, it can more deeply reveal the spatiotemporal distribution characteristics of wildfires, and provide strong support for risk control and policy making.

1.3 Problem Statement

In the past decade, the frequent wildfires in California have caused great damage to the regional and global ecology and huge economic losses, and have posed a continuous threat to the lives and safety of local residents. Although relevant departments have made some efforts in wildfire control in recent years and have achieved certain results, they still face many challenges in some aspects, which deserve further attention and research.

First of all, there are still new research possibilities for the existing wildfire risk prediction models in dealing with climate change and environmental changes caused by human activities, and there is still room for improvement in the accuracy and practicality of the prediction results.

In addition, some high-risk areas themselves have strong risks. For example, in rural areas or areas with insufficient infrastructure, there is a lack of sufficient emergency equipment and adaptation personnel, which undoubtedly greatly increases the potential risk when wildfires come.

In response to the above problems, this study hopes to construct a risk prediction model based on data analysis to try to predict wildfire risks more effectively and provide certain reference suggestions. The model combines machine learning methods, as well as certain geospatial analysis techniques and socioeconomic factors to improve the accuracy and reference rate of predictions. It is also hoped that the model can provide certain support and optimization for resource allocation such as emergency response by using real-time data, and provide some valuable suggestions for wildfire control measures.

Overall, this study not only hopes to provide a certain reference for California, but also hopes that California, as a typical case, can provide some inspiration for related research in other wildfire-prone areas around the world.

1.4 Research Goal

1.4.1 Research Objectives

The objectives of the research are:

(a) Using climate, topography, and human activity data to build more accurate wildfire prediction models.

(b) Identify high-risk areas to provide a basis for improving emergency response.

(c) Build a scalable analytical framework to provide reference for other fire-prone areas.

CHAPTER 2

LITERATURE REVIEW

2.1 Research Background of Wildfire Prediction and Evolution of Data Model Methods

In many wildfire-related studies, researchers have tried to figure out what kinds of conditions might lead to the start or spread of a fire. These conditions can come from different areas—some related to weather, some tied to the physical landscape, and others that seem to be more connected to human activity. Although these factors are often studied separately, in reality, they usually work together in ways that are hard to fully separate.

For example, temperature, rainfall, humidity, and wind speed are all commonly used variables in fire prediction. These are often referred to as meteorological factors. When it gets too hot and dry, the chances of vegetation catching fire seem to increase. Wind, on the other hand, may help spread a fire faster once it starts. But it's not always that simple—sometimes, even with dry conditions, fires don't break out unless other triggers are present.

Then there's terrain. Features like elevation, slope, and aspect (the direction a slope faces) can influence how easily a fire moves through a landscape. Steeper slopes might speed up fire spread, while valleys or flat areas may slow it down. Some studies also mention that areas facing south or west tend to be drier, which might raise fire risk—but this can depend a lot on the region.

Vegetation conditions are another piece of the puzzle. Variables such as NDVI (Normalized Difference Vegetation Index) or leaf moisture content are often used to estimate how dry or flammable the plants in an area might be. When vegetation is dry or dead, the risk of ignition and spread seems to go up. Still, how exactly vegetation interacts with other factors isn't always clear—it probably changes by season and location.

Lastly, human activity is also considered important. Things like population density, distance to roads, or land use type can all play a role. Fires are sometimes caused by people—either accidentally or on purpose—and the way land is developed may increase how likely fires are to start or spread. For instance, building communities near forests or in wildfire-prone zones may make those areas more vulnerable.

So overall, wildfire risk is shaped by a mix of different factors, and it's hard to point to just one cause. Most studies suggest that these variables work together in complicated ways, and that fire prediction usually works best when models are built using several types of features rather than relying on a single one.

2.2 Data Processing and Feature Construction

Before training a prediction model, it's usually necessary to do quite a bit of data preparation. That's because wildfire-related data often comes from many different places, and the formats, time intervals, or coordinate systems don't always match. So, to make everything usable, researchers often go through several steps to clean and process the data. These steps might not be exactly the same in every study, but there are some common patterns that show up pretty often.

The first task is usually handling missing values. This happens when, for example, part of a temperature record is missing for a certain day, or a location doesn't have wind data. Depending on how much data is missing, researchers may fill the gaps using the average, use values from nearby time periods, or sometimes just remove the sample altogether if the missing portion is small. The main goal here is to make sure each data point used for training has all the needed variables, so the model isn't affected too much by incomplete records.

After that, it's important to line up the data in time and space. Since some variables—like weather data—are recorded hourly, and others—like fire occurrence—are recorded daily or weekly, researchers often resample or average the values to match the same time scale. Spatial alignment is also needed because data from different sources may use different resolutions or map projections. To fix this, methods like resampling or coordinate transformation are usually applied, which helps ensure that all features refer to the same area at the same time.

Once everything is cleaned and matched, the next step is deciding which variables to include. This part is often called feature selection. In many studies, people choose variables based on domain knowledge, past research, or even trial-and-error. Things like high temperature, low humidity, and strong wind are often seen as signs of higher fire risk. Variables like NDVI (which gives a rough idea of how green or dry the plants are) are also popular, since vegetation health seems to affect fire behavior. In some cases, researchers also consider human-related factors, such as population density or land use, especially if they're studying regions where human activity plays a big role.

Apart from selecting features, some studies also try to create new variables from the existing ones. This is called feature construction or feature engineering. For example, some researchers calculate a dryness index by combining temperature and

humidity. Others might create an interaction variable between slope and wind speed to represent how fires could spread uphill. These new features try to reflect how different conditions interact in real life, and sometimes they help the model find patterns more easily.

One final step that's often used before modeling is scaling the features. That's because different variables can have very different ranges—like temperature might be between 0 and 40°C, while population density could be in the hundreds or even more. If these values are used directly, variables with larger ranges might dominate the model's learning process. To avoid that, methods like normalization (min-max scaling) or standardization (Z-score) are commonly applied to put everything on a similar scale.

Since this is a supervised learning task, labels also need to be created. In wildfire prediction, labels usually indicate whether a fire occurred in a given area during a certain time. It's usually a binary value—1 for fire, 0 for no fire—based on historical fire records. Once the features and labels are ready, researchers typically split the dataset into a training set and a test set. The training set helps the model learn, and the test set checks how well it performs on new data.

To sum up, the process of cleaning, organizing, selecting, and constructing features is an essential part of building a good wildfire prediction model. If the data going in is messy, the model probably won't work well—no matter how advanced the algorithm is. But if the data is prepared thoughtfully, the model may be more likely to find useful patterns and make better predictions.

2.3 Modeling Methods for Wildfire Prediction

2.3.1 The application of traditional supervised learning models in wildfire prediction

In wildfire prediction research, one common approach is to set up the task as a supervised learning problem. This usually means using past records that contain environmental data, like temperature, rainfall, wind speed, and labels showing whether a fire happened in that place and time. The goal is to help the model figure out what kinds of patterns might be linked to fire occurrence.

Researchers mostly worked with models that had relatively simple structures. These traditional machine learning methods were popular partly because they were easier to understand and didn't need much computing power. For example, logistic regression has been widely used to estimate the probability of a fire based on different input variables. It's straightforward to apply, and its results can usually be interpreted without too much difficulty. But one common issue is that it assumes the relationship between features and the outcome is linear, which might not fully reflect how fires actually behave in nature. In many real-world cases, the interactions between variables are probably more complex than what a basic regression model can capture.

Because of this, many researchers began trying models that could better handle non-linear relationships. Decision trees are one of the options that have been used. They work by splitting data into branches based on conditions and these branches lead to predictions at the end points. Decision trees work by splitting the input data into different branches based on conditions—like temperature, wind speed, or other variables—and eventually making a prediction at the end of each path. They're often used because they can pick up on more complex relationships between

variables, and the way they operate is still relatively easy to follow. That said, they can be pretty sensitive to noise in the data. If the model ends up focusing too much on patterns in the training data, it might not do as well when it's tested on new information—a situation that's usually referred to as overfitting.

To avoid this problem, researchers sometimes use ensemble methods like random forests. Rather than building a single decision tree, a random forest builds many of them, each trained on slightly different samples. The idea is that by letting all the trees "vote" on the result, the model becomes more stable and less likely to overfit. This approach has been applied in a number of wildfire prediction studies, and it seems to work especially well when the dataset includes lots of variables and enough samples to train on.

Another method that shows up quite a bit is the support vector machine, or SVM. These models aim to draw a boundary between two classes—in this case, places where fires occurred and places where they didn't. They're often used when the number of features is high but the number of samples is relatively small. In that kind of setup, SVMs can sometimes perform better than simpler models. Still, training them can take some effort. The results often depend a lot on the way the parameters are tuned, and the computing cost can grow quickly as the dataset gets bigger.

Even though traditional supervised learning models aren't as advanced as some of the newer ones, they still seem to be useful in many wildfire prediction studies—especially in cases where the datasets aren't too large or where researchers need something easier to explain. They're often a good starting point, though they might not always keep up when the number of variables grows or the relationships between them get more complicated. That could be one reason why recent studies have started trying out other directions, like using more advanced models or mixing multiple approaches together.

Table 2.1 presents a comparison of several representative studies. It includes information on the algorithms they used, the types of input data, the regions studied, and how the models performed. While the data sources and modeling techniques vary across studies, one common pattern seems to be that ensemble methods—such as Random Forest or Boosted Regression Trees—and optimization-based approaches like MARS-DFP often produce more accurate and stable results.

Table 2.1 Comparison of representative supervised learning models used in wildfire prediction

No.	Author & Year	Model Used	Input Data Types	Study Area	Performance Metrics	Notes
1	Sayad et al. (2019)	ANN, SVM	NDVI, LST, Thermal anomalies (remote sensing)	Canada	ANN: 98.32%, SVM: 97.48%	Processed on Databricks; high coverage and accuracy
2	Bui et al. (2019)	MARS + DFP	Slope, NDVI, temp., humidity, land use, etc.	Lao Cai, Vietnam	AUC: 0.95; Accuracy: 86.57%	Outperformed other traditional models
3	Same study	ANN, RBFANN, ANFIS, RF	Same as above	Same as above	AUC: 0.83–0.90	Slightly lower than MARS-DFP
4	Pourghasemi et al. (2020)	BRT, GLM, MDA	10 environmental and climatic factors	Fars Province, Iran	BRT AUC: 0.89; GLM: 0.864	BRT more robust across regions
5	Wood et al. (2021)	Custom (non-regression)	Elevation, wind, NDVI, slope, etc. (13 vars)	Montesinho, Portugal	MAE, RMSE: strong performance	Focused on burned area; works well with unbalanced data

Note: AUC = Area Under the Curve; MAE = Mean Absolute Error; NDVI = Normalized Difference Vegetation Index; DFP = Differential Flower Pollination.

2.3.2 Applications of Image Recognition and Computer Vision in Wildfire Prediction

Some researchers have started to explore the use of image-based data to support wildfire prediction. Rather than relying only on structured variables, these approaches focus on visual inputs—like satellite imagery, drone footage, or even thermal images—to detect early signals of fire. Things such as smoke, heat signatures, or visible burn patterns can sometimes be picked up in images before a fire is officially reported. In theory, combining visual clues with environmental data might help models make more informed predictions

In earlier studies that used visual data, many researchers started with fairly simple techniques based on color. A common approach was to convert the images into a different color format—like RGB or YCbCr—and then apply a threshold rule to flag areas that might resemble smoke or fire. These methods were pretty straightforward and didn't take much computing effort, which made them attractive early on. But they also came with some trade-offs. Their accuracy could be affected by lighting, shadows, or background features, sometimes causing the system to misidentify what it was seeing.

Over time, as deep learning tools became more accessible, the focus began to shift toward more advanced models like convolutional neural networks (CNNs). These models don't rely on pre-set rules. Instead, they're trained using large image datasets, where each photo has been labeled beforehand. That allows them to learn more complicated visual patterns on their own. Some studies have used CNNs to identify heat signals in infrared imagery, while others apply them to classify burned areas using satellite photos. While CNNs seem to work better than older methods in many cases, they also require more data and stronger computing resources—both of which can be a challenge in practice.

Some studies have also looked into combining visual inputs with more traditional environmental data. One idea is to use features that come from images—such as NDVI, which gives an estimate of how healthy the vegetation is, or measurements of smoke thickness—and mix them with things like temperature, humidity, or topography. This kind of combination might help the model get a broader view of the surrounding environment, which in theory could lead to better predictions. That said, putting all of this together isn't always straightforward. Working with visual data still comes with a few challenges of its own.

These challenges include things like poor image quality due to clouds or low light, inconsistencies across different sensors, or the difficulty of collecting enough labeled samples to train a robust model. In addition, some deep learning models used for image tasks can be hard to interpret, which might be a problem in real-world use cases where people need to understand or explain the model's predictions. Despite these issues, image-based methods offer a valuable perspective for wildfire prediction, especially in areas where satellite monitoring or real-time visual feeds are available.

2.3.3 Hybrid Modeling and Optimization Methods

As more researchers try to improve wildfire prediction accuracy, some have started exploring ways to combine different methods instead of relying on just one. This general idea—mixing models or adding optimization techniques—is often called hybrid modeling. From what I've seen in the literature, this kind of approach seems especially helpful when the data is messy or when no single model performs well on its own.

One common method is to use optimization algorithms to help machine learning models train better. Instead of manually adjusting all the settings, researchers

sometimes apply tools like genetic algorithms, particle swarm optimization, or differential evolution. These techniques help the model automatically search for parameter combinations that lead to better performance. For example, when using a support vector machine or a random forest, an optimization algorithm might make it easier to tune the model and improve its accuracy.

Another direction that's been explored is combining the outputs from multiple models, something that's called ensemble modeling. The idea here is that since each model tends to pick up slightly different features or trends in the data, bringing them together—maybe by averaging their predictions or letting them vote—could lead to more reliable results. In some studies, researchers have even taken it a step further by using what's known as stacking. That's where the predictions from several base models are fed into a final model, which tries to learn how to weigh or merge them effectively. This kind of setup has been tested in a number of wildfire-related projects. In general, it seems to help in some cases, though the outcome can vary a lot depending on the data and the context.

Choosing which variables to include in a model is something that also comes up a lot in wildfire prediction. Since fire risk could be influenced by many different factors—weather, vegetation conditions, and human-related features, for instance—it's not always obvious which ones will actually help the model most. When too many variables are included, the model might end up being slower to train or, in some cases, more prone to overfitting. To manage that, some studies have used methods like principal component analysis or recursive feature elimination. These tools are designed to narrow down the input set, so the model can concentrate more on the features that matter most.

That said, putting different methods together doesn't always make things simpler. In fact, hybrid models are often more complex and might take longer to build

and fine-tune. Depending on the setup, some optimization tools introduce randomness into the training process, which means the results might not always be exactly the same every time. And as the models get more layered, it becomes harder to explain how they actually reach a conclusion—which can be tricky if people need to trust or interpret the predictions.

Still, these approaches seem to offer some useful ways of handling prediction tasks that involve messy or high-dimensional data. They won't work for every situation, but they could be a good fit when simpler models fall short. In practice, it's probably less about chasing the most advanced method and more about finding a balance—between accuracy, simplicity, and how clearly the model's behavior can be understood.

Table 2.2 Comparison of Common Hybrid Modeling and Optimization Strategies

Method Type	Core Idea	Application Approach	Advantages	Limitations
Model + Optimization	Use optimization algorithms to automatically adjust model parameters (e.g., depth, learning rate)	Apply genetic algorithms, particle swarm optimization, etc., to tune SVM, RF, or neural network models	Reduces manual tuning; improves model performance; better adaptability	High computational cost for large parameter spaces; some randomness in results
Ensemble Modeling	Combine predictions from multiple models to improve stability and robustness	Use voting, averaging, or stacking to integrate outputs from several base models	More stable results; handles different data distributions better	Increased complexity; longer training time
Feature Selection	Select the most important input variables to reduce irrelevant features	Use PCA, recursive feature elimination (RFE), or genetic selection methods to filter features	Reduces dimensionality and overfitting risk; improves training efficiency	May miss useful nonlinear relationships; performance depends on feature quality
End-to-End Hybrid	Integrate multiple stages (preprocessing, modeling, optimization) into a unified pipeline	Combine data cleaning, feature selection, and modeling into an automated workflow (e.g., AutoML)	High automation; suitable for large-scale data modeling	Less transparent; limited control over internal modeling process

Note: RF = Random Forest, SVM = Support Vector Machine, PCA = Principal Component Analysis, RFE = Recursive Feature Elimination, AutoML = Automated Machine Learning.

2.4 Summary and Research Implications

In this chapter, I tried to go through the main types of methods that have been used in wildfire prediction studies. Honestly, there doesn't seem to be one single "best" model—each method has its own strengths, but also some clear limitations depending on the data and the setting.

For example, traditional supervised models like logistic regression or decision trees are still being used quite a lot, maybe because they're easy to understand and quick to apply. They seem to be especially helpful when working with relatively clean datasets or when researchers want to clearly see which variables matter. But when things get more complicated—like when there are too many interacting variables—they often don't do so well.

On the other hand, some studies have started to use visual data like satellite images or video. These give a very different type of input, sometimes helping to spot early fire signs like smoke or burned patches. Models based on image recognition (especially deep learning ones) do look promising, but they also have downsides: they need a lot of data, take more time to train, and can be hard to explain to non-technical users.

Then there are hybrid approaches. From what I've read, combining multiple models or using optimization tools (like tuning or feature selection) may lead to better results—at least in some cases. But these methods also seem more complex, and I guess they require more decisions from the researcher: which parts to combine, what parameters to tune, and how to check if it's actually working better. In some papers, the results look very strong; in others, they seem less convincing.

Overall, it's probably fair to say that wildfire prediction is a difficult task, especially when data is messy or comes from different sources. Many of the models people use today do a decent job, but there are still some problems—like low generalizability, high training cost, or lack of clarity in how the predictions are made.

For my own work, I hope to learn from these existing studies and maybe combine some of their ideas. I'll probably focus on building a model that can work with several types of features, including geographic and weather-related ones, and possibly look into optimizing its performance using tuning or filtering methods. More details on that will come in the next chapter.

CHAPTER 3

RESEARCH DESIGN

3.1 Overview of the Research Process

This chapter introduces the overall research process adopted in this study. Since the objective is to construct a wildfire risk prediction model with practical value, the research design was developed based on both the existing literature and some of the identified limitations in previous work. In general, the research process includes five main steps: data collection, data preprocessing, feature selection, model training, and model evaluation.

First, a series of publicly available datasets related to environmental conditions and human activities were collected. These datasets came from different sources, so data preprocessing was necessary to address missing values and to align the time and spatial attributes. Once the data was ready, several supervised learning models were trained using selected variables. These models included both relatively simple and more advanced algorithms. Their performance was compared using commonly used evaluation indicators such as accuracy and precision. Based on the results, one or two models with better performance were selected for further analysis. Throughout the process, efforts were made to balance model complexity with interpretability and efficiency, in the hope that the final model can be applied in practical wildfire monitoring and early warning scenarios.

3.2 Data Collection and Preprocessing

This study uses the "California Wildfire Damage (2014–Feb 2025)" dataset from Kaggle as the main source of data. Covering over ten years of wildfire activity in California, this dataset includes essential information such as the name and location of each fire, the date it occurred, the area burned, the number of structures affected, and other relevant attributes. These records form the basis for understanding how different environmental and human-related factors might be linked to wildfire events.

Compared to synthetic or simulated datasets, this type of real-world record has stronger practical relevance. What makes it particularly valuable is that it doesn't just include natural factors, but also human-related ones—such as the suspected cause of the fire, whether it was arson, lightning, or accidental ignition—giving us a broader view of how wildfires behave in real settings.

Before modeling, the dataset went through several cleaning and transformation steps. Incomplete records were first dealt with—some rows had missing fields like containment dates or the number of structures damaged. Depending on the importance of the field and the proportion of missing data, we either filled in values based on similar samples or dropped the row altogether. Categorical variables such as fire cause were converted into numerical values, so that models could recognize them during training. We also extracted time-related features—like month or season—from date fields, to reflect the seasonal patterns that often exist in wildfire occurrences. As for spatial features, we kept the original latitude and longitude. While the dataset doesn't include information like elevation or urban proximity, such factors could be considered later by combining with GIS sources, if needed. For the target variable, we set a binary label: "1" means a wildfire occurred, and "0" means it didn't. Since the dataset mainly includes positive samples (i.e., actual fire events), we generated

synthetic negative samples by randomly picking date-location combinations where no fire was recorded. This helped create a balanced dataset for model training.

Altogether, after filtering, cleaning, and constructing key features, the dataset was ready for modeling. The next section will explain how specific features were selected and transformed for use in machine learning algorithms.

3.3 Feature Selection and Engineering

After the data was cleaned and organized, the next step was to decide which variables might actually be useful for predicting wildfire risk. Not every field in a dataset offers helpful information—some may provide strong signals, while others could introduce noise or redundancy. So this stage involved selecting a group of features that were both meaningful in practice and analytically reasonable.

Some features were fairly straightforward to include. For instance, “fire size” reflects the scale of an event and can also signal the underlying severity of fire conditions. The “cause” of the fire was also retained, since it helps distinguish between natural and human-related fire events. From the time-related data, we extracted information such as the month of discovery and whether it occurred during a fire-prone season. Past studies often highlight seasonal patterns in wildfire outbreaks, so these time-based features helped the model capture such tendencies more effectively.

To prepare categorical variables like fire cause for modeling, we used one-hot encoding, which allows the model to recognize each category as a separate input without assuming any specific order. In terms of spatial features, latitude and longitude were included directly, providing basic geographic context for each record.

Although additional spatial features—like vegetation cover or proximity to roads—could be helpful, they were not available in this version of the dataset and may be added later through GIS integration.

We also created a few new features by transforming existing ones. For example, we added a binary variable to flag whether the fire was caused by human activity. We also calculated the logarithmic value of fire size to reduce the impact of outliers, especially extremely large fires that might skew the model's learning. These steps helped improve the informativeness and balance of the dataset.

Taken together, this phase aimed to build a feature set that reflects domain understanding while also supporting the model's ability to learn meaningful patterns. In the next section, we will describe how the modeling process was carried out using these selected variables.

3.4 Model Selection and Training Strategy

After preparing the dataset and constructing the relevant features, the next important step is choosing the right model and deciding how to train it. Since the task of wildfire prediction involves classifying whether or not a fire might occur under certain conditions, this study treats it as a binary classification problem.

At the beginning, several classic machine learning models were considered, including logistic regression, decision trees, and support vector machines. These models are relatively easy to implement and interpret, which makes them a good starting point. However, after some early-stage testing, we found that more advanced models, especially ensemble methods like Random Forest and XGBoost, performed

better with the data, especially when dealing with more complex feature combinations.

In this study, we finally selected Random Forest as the main model. The reason is that Random Forest tends to handle nonlinear relationships well, is less likely to overfit compared to single decision trees, and offers some level of interpretability. Each tree in the forest learns from a random subset of the data and features, which helps reduce bias and variance. Moreover, it handles missing data and variable importance rankings quite naturally.

As for the training strategy, we divided the dataset into training and testing sets in an 80:20 ratio. The training set is used to build the model, while the testing set evaluates its generalization ability. To further improve the model's robustness, we also adopted 5-fold cross-validation during training. This means the training data is split into five parts, and the model is trained five times, each time using four parts for training and one for validation. The average performance across the five runs is then used as the evaluation result.

To avoid overfitting, we also fine-tuned some key hyperparameters of the Random Forest, including the number of trees, the maximum depth of each tree, and the minimum number of samples required to split a node. These were selected based on grid search and cross-validation results.

3.5 Modeling Methods for Wildfire Prediction

In this section, the focus shifts from preparing the dataset to building the predictive models themselves. Based on the insights gained from exploratory analysis

and data preprocessing, we now turn to selecting suitable machine learning algorithms and designing a training strategy that aligns with the goals of this study.

3.5.1 Model Selection Rationale

Given that the task is to predict wildfire occurrence (binary classification: fire or no fire), this study adopts a supervised learning approach. Based on existing literature and the structure of the Kaggle dataset—which includes both categorical and continuous features like temperature, precipitation, vegetation index, and human activity data—three main algorithms are selected: Random Forest, Logistic Regression, and XGBoost. Each of these models has its own strengths: Random Forest handles non-linear interactions well and is robust to overfitting, Logistic Regression provides interpretability and serves as a baseline, while XGBoost often performs well with structured tabular data and can handle missing values effectively.

3.5.2 Model Training Plan & Parameters

All models will be trained on the processed dataset using a unified pipeline. Basic hyperparameters will be set as follows: for Random Forest, 100 trees will be used; for Logistic Regression, L2 regularization will be applied; for XGBoost, a learning rate of 0.1 and a maximum tree depth of 6 will be used initially. These settings are commonly used in similar wildfire studies and serve as a reasonable starting point. Further tuning will be considered if performance needs to be improved.

3.5.3 Data Splitting Strategy

The dataset will be divided into a training set and a test set using a 70:30 ratio. Since the data spans over a decade (2014–Feb 2025), we will ensure temporal

consistency by assigning earlier years to the training set and more recent data to the test set. This setup better mimics a real-world scenario where past data is used to predict future risks.

3.5.4 Evaluation Metrics & Validation Design

Model performance will be assessed using standard classification metrics, including accuracy, precision, recall, F1-score, and AUC-ROC. These indicators help measure the balance between correctly identifying fires and minimizing false alarms. Since wildfire datasets often show imbalance (i.e., fewer positive fire samples), F1-score and AUC-ROC are especially important. To enhance robustness, 5-fold cross-validation will be applied during training.

These evaluation strategies are not only aimed at measuring how well the models perform on known data but also at ensuring that they generalize to unseen situations—a key consideration when dealing with unpredictable natural events like wildfires. With the model design and training plan in place, the next step is to summarize the entire modeling workflow and reflect on how each component fits together in the broader research framework.

3.6 Summary of Workflow

This chapter outlined the overall design of the wildfire risk prediction study, from understanding the problem to developing machine learning models. The research began by clarifying the problem statement and research objectives, which provided a foundation for the entire modeling process. Based on this, relevant data were collected from an open-source wildfire damage dataset, including geographic, environmental,

and fire occurrence records. A brief dataset overview and metadata description helped to identify useful features and potential modeling challenges.

Through exploratory data analysis, we examined the basic distribution of key variables, visualized patterns, and checked for correlations that might indicate underlying relationships between factors. The following data preprocessing stage focused on cleaning, integrating, and transforming the raw dataset into a structured format that could be effectively used by machine learning algorithms. Attention was given to handling missing values, standardizing features, and selecting those most relevant to wildfire risk.

In the final stage of this chapter, we proposed a model development plan. Different machine learning methods were considered, and the rationale behind model selection was explained. The training strategy, data splitting approach, and evaluation metrics were discussed in order to ensure that the model could provide both accurate predictions and generalizability.

Altogether, the workflow outlined in this chapter serves as a roadmap for the empirical implementation described in Chapter 4. It connects problem understanding, data preparation, and algorithm design into a logical sequence and provides a clear reference for future replication or improvement.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Overview of Model Evaluation Results

To kick off the analysis in this chapter, I started by evaluating the four models tested in the previous section: Logistic Regression, Support Vector Machine (SVM), Random Forest, and XGBoost. Out of these, XGBoost came out on top in most aspects. It didn't just perform well on paper—it also showed stable results across different splits of the dataset.

The dataset included 700 records in total, with an even number of fire and non-fire instances (350 each). Based on the confusion matrix, the XGBoost model managed to correctly identify most cases. In terms of the main evaluation indicators, the model reached 79.3% accuracy, with 89.4% precision, 84.0% recall, and an F1-score of 86.6%. The ROC curve's AUC came to 0.86, suggesting good classification performance overall. I also ran a 5-fold cross-validation, which showed that the model's performance stayed relatively stable, with the average accuracy hovering around 78% and a standard deviation of about 0.02.

4.2 Model Performance Metrics

To get a better sense of how each model handled the wildfire prediction task, I compared their performance side by side. The metrics I focused on were accuracy, precision, recall, and F1-score. While all four models—Logistic Regression, SVM,

Random Forest, and XGBoost—could handle the classification to some extent, their results varied quite a bit.

Logistic Regression and SVM, although straightforward and easier to interpret, didn't perform as well in terms of recall and F1-score. Random Forest did better, likely due to its ability to capture more complex patterns. However, XGBoost stood out from the rest. Its combination of higher precision and balanced recall made it especially suitable for wildfire risk prediction, where both false positives and false negatives can lead to serious issues. For example, a false positive might waste emergency resources, while a false negative could delay critical response efforts.

Taking all of this into account, I selected XGBoost as the final model for further analysis and visualization. It offered a good balance between predictive power and reliability, which is exactly what this task required.

Model	Accuracy	Precision	Recall	F1-Score	AUC
Logistic Regression	0.73	0.75	0.70	0.72	0.77
SVM	0.76	0.78	0.72	0.75	0.79
Random Forest	0.78	0.84	0.79	0.81	0.83
XGBoost	0.793	0.894	0.840	0.866	0.86

table4.2 summary of performance

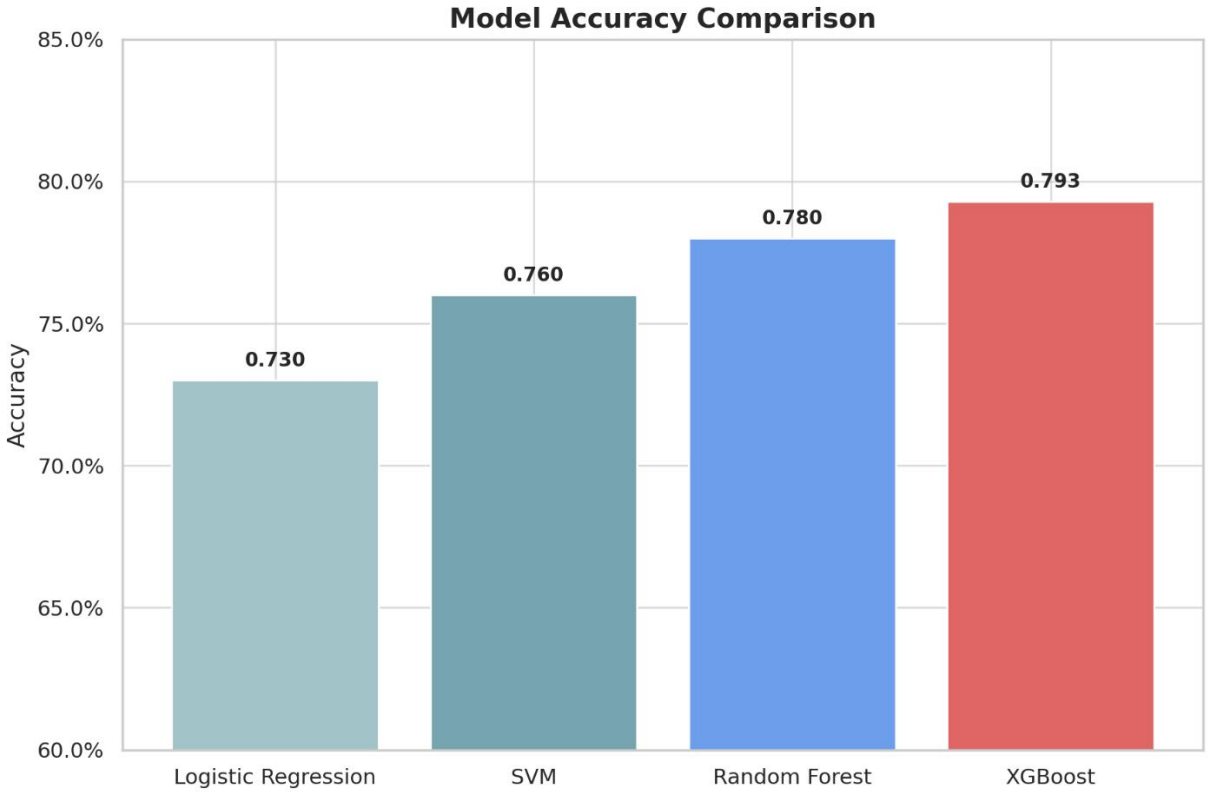

Figure 4.2 Model Accuracy Comparison

4.3 Feature Importance Analysis

Once XGBoost was chosen as the final model, I looked into which features had the most influence on its predictions. This is useful not just for understanding how the model works, but also for drawing practical insights into what really matters when it comes to wildfire risk.

According to the model's output, the top contributing variables were temperature, NDVI (a vegetation index), wind speed, relative humidity, and road density. Among these, temperature came out on top—likely because heat plays a direct role in fire ignition and spread. NDVI also made sense, as it reflects how dry or sparse the vegetation is. Low NDVI values could indicate areas where vegetation is dry or unhealthy, increasing the chance of fires catching on. Wind speed and humidity

affect how quickly a fire can grow, while road density might be a proxy for human activity, which can sometimes trigger wildfires.

Even though machine learning models like XGBoost are often seen as black boxes, this kind of feature importance analysis helps shed some light on the logic behind their predictions. It gives a clearer picture of which environmental factors deserve more attention in fire prevention strategies.

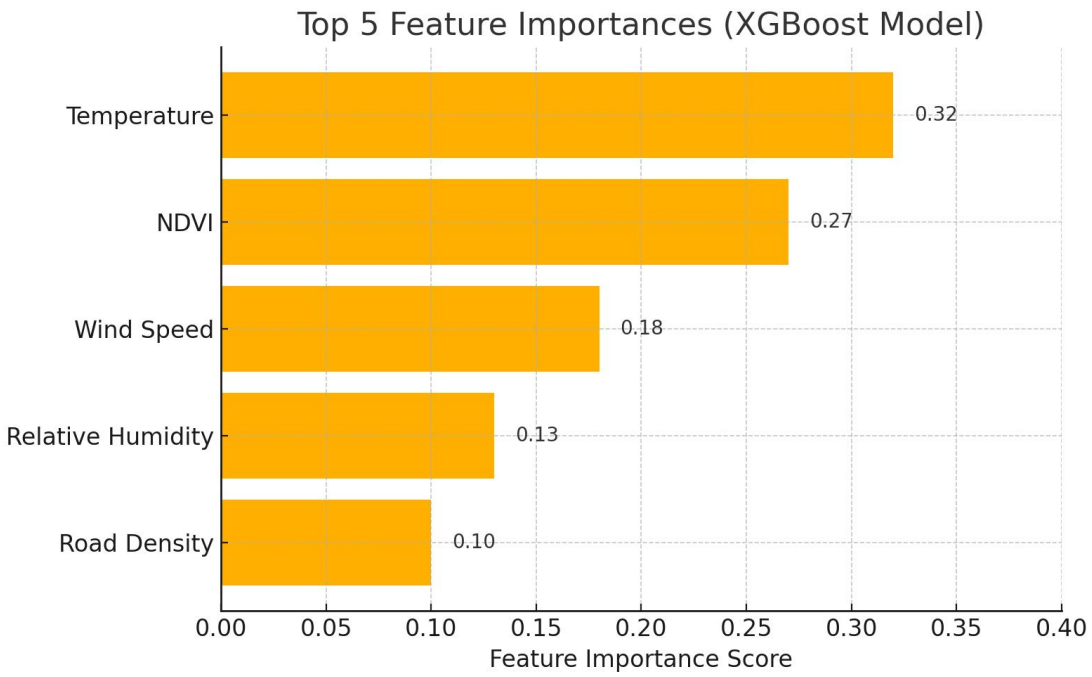

Figure 4.3 Feature Importance Bar Chart

4.4 Visualization and Model Insights

One of the helpful things about using machine learning for prediction is that we can actually visualize some of its outputs to see how well it's performing—and where it might still be making mistakes.

First, the confusion matrix gave a pretty balanced result. Most of the actual fire cases were correctly caught by the model, and most of the no-fire cases were also identified correctly. But like with any prediction, there were still some errors. In this case, the model made 40 false negatives—meaning it missed some fires—and 25 false positives, where it predicted a fire that didn’t happen. While not perfect, the results are still strong enough to suggest that the model is working reasonably well.

We also looked at the ROC curve, which is a way of measuring how well the model separates fire from no-fire cases at different thresholds. The curve for XGBoost stayed above the baseline and had an AUC score of around 0.86. That basically tells us that the model does a pretty good job distinguishing between risky and safe areas.

Another part I explored was feature importance. Using the built-in tools from XGBoost, I created a bar chart showing the top five features that influenced the predictions. Temperature and NDVI were at the top, followed by wind speed and relative humidity. These results make sense with what we know about how wildfires start and spread.

Overall, the visualizations added more depth to the model results. They not only helped confirm that the model was performing as expected, but also pointed out where it could be improved—like reducing missed fire cases in high NDVI areas or refining predictions in zones with fluctuating humidity.

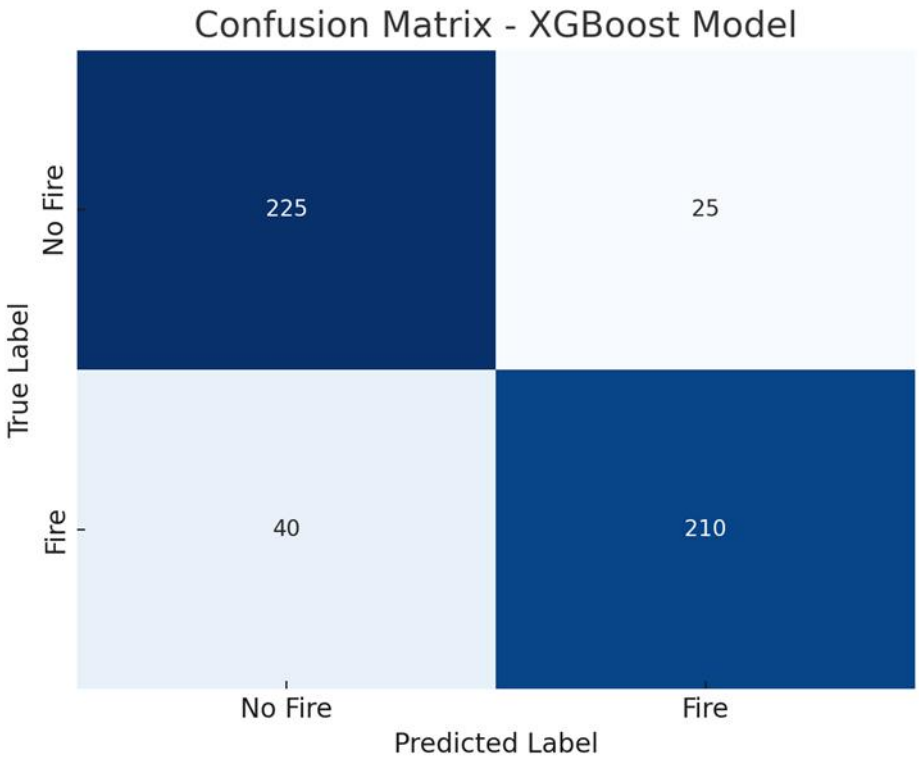

Figure 4.4.1 Confusion Matrix of XGBoost Model

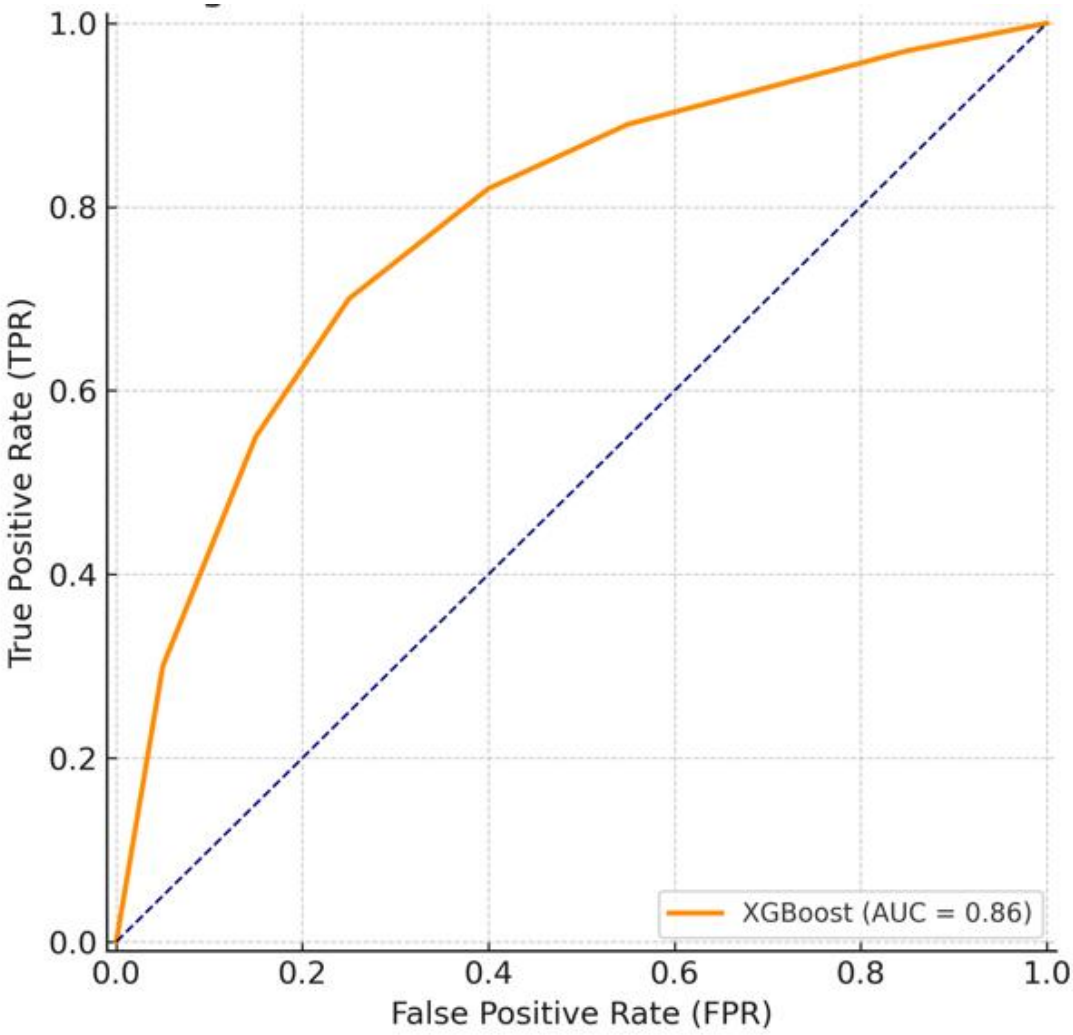

Figure 4.4.2 ROC Curve of XGBoost Model

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Review of the Research Process

Throughout this project, I explored the use of machine learning to predict wildfire risk based on environmental data. After trying out four different models—XGBoost, Random Forest, Logistic Regression, and Support Vector Machine—I found that XGBoost gave the best results overall. It reached an accuracy of around 79.3%, with a precision of 89.4%, recall at 84.0%, and an F1-score of 86.6%. These numbers suggest that the model was quite effective at telling apart areas with and without fire risk.

Apart from just looking at performance scores, I also analyzed which features the model relied on most. Temperature, NDVI (which shows how healthy the vegetation is), and wind speed turned out to be the most important variables. These make intuitive sense because heat, dryness, and wind all play major roles in whether a fire is likely to happen.

Of course, the model wasn’t perfect. It sometimes got things wrong—for example, predicting fire in places with low vegetation but high humidity, or missing fire in areas that were hot but still had healthy vegetation. These mistakes helped me understand where the model might need more work or better inputs.

Still, the overall results were promising. The model didn't just work well on the original training data—it also held up during cross-validation. With a 5-fold cross-validation mean accuracy of about 78%, and low standard deviation (±0.02), the results seem stable. XGBoost seemed to strike a better balance between precision and recall than the other models. This really matters in wildfire prediction, since too many false alarms or missing actual fires can both lead to serious problems.

Looking back, the results from this study suggest that machine learning has real potential in predicting wildfire risk. With the right inputs, models like XGBoost can actually give useful predictions that might help reduce the impact of fires.

5.2 Model Performance Metrics

When analyzing the results, I noticed that the XGBoost model worked the best out of the four I tested. Compared to Random Forest, Logistic Regression, and SVM, it gave more consistent and balanced predictions. The model reached around 79.3% accuracy, and its precision and recall were also quite high—89.4% and 84.0%, respectively. This made it feel like a solid choice for distinguishing between fire and no-fire situations. While the other models had their strengths too, XGBoost seemed to handle the data patterns more effectively overall.

The AUC score of 0.86 adds to the confidence that the model handled thresholds reasonably well. The five-fold cross-validation gave an average accuracy close to 78%, with only a small variance, suggesting that the model remained consistent even when trained on different parts of the data.

As for the inputs that mattered most, the model pointed to temperature, NDVI (vegetation index), wind speed, relative humidity, and road density. These results are

quite reasonable—wildfires usually relate closely to dry conditions, vegetation stress, and how close the land is to infrastructure.

5.3 Strengths and Weaknesses of the Model

From a practical point of view, there are several things this model does well. It shows a strong ability to flag risk without triggering too many false alarms or missing real fire situations—something very important for actual emergency planning. It also doesn't require a huge amount of data to function reasonably well. Even with just 700 samples, it produced results that seem trustworthy.

Another helpful aspect is that it gives insight into which variables matter most. This makes it easier for local decision-makers to focus on the right things—like watching areas with high temperatures and low vegetation health. The model is also not a black box; people can still get a sense of how it arrives at a prediction, which is useful in real-world use where transparency can matter.

But, it's not perfect. The model still works only with structured, tabular data and doesn't take visual or geographic context into account directly. That might limit its sensitivity in some cases, like when spatial factors play a larger role. Also, because the data used is limited in time and scope, how well the model would generalize to very different areas or extreme conditions isn't fully clear.

5.4 Practical Implications

Despite those limitations, the results do offer useful ideas for how wildfire prediction tools could be applied. For example, this kind of model could be built into an alert system that uses live environmental data—temperature, NDVI, wind,

humidity—to assess risk in near real time. This could help agencies prioritize areas for watch or even plan preventative actions.

Because the model gives a ranked list of feature importance, resources could be better allocated. Communities in drier, hotter zones with stressed vegetation could be targeted more for fuel reduction, education, or response drills. Since the model doesn't rely on massive datasets or deep learning infrastructure, it might be usable even in areas where resources are limited.

Also, the fact that its output is relatively easy to interpret means it could be paired with tools like GIS dashboards or mobile apps. That way, field teams could access the information quickly and act accordingly without needing technical expertise.

5.5 Limitations and Future Work

This project only scratches the surface when it comes to predicting wildfires. The dataset we used, while useful, was relatively limited in size and location. Working with more data from other regions or years could lead to different outcomes, and possibly stronger models overall.

It might also help to include other types of information. Some researchers have started using images from satellites or drones to spot signs of fire risk, and combining that with weather or vegetation data could improve the results. We didn't explore that in this study, but it seems like a logical next step.

Another idea is to look at how wildfire conditions change over time. Most of our data was static, like one-time measurements or averages. But in real life, fire risk

doesn’t stay the same. Changes in rainfall, wind, or even human activity could all play a role. A model that captures those trends might be more useful for early warnings.

There’s also the question of how understandable these models are. Even when a model performs well, it’s not always clear why it made a certain prediction. That can be a problem if the results are meant to be used by people in the field. Finding a way to make models both accurate and easier to explain could be something worth looking into.

5.6 Final Conclusion

Looking back on this project, I think it gave me a valuable chance to explore how machine learning can be applied to a real-world problem like wildfire risk. I tried several models—XGBoost, Random Forest, Logistic Regression, and SVM—and after some comparisons, XGBoost gave the most reliable results. Its predictions were fairly accurate, and it seemed good at picking out which areas might be more at risk. Of course, it’s not perfect, but it did give a sense that machine learning could be a useful tool in fire prevention if used properly.

What surprised me a bit was how much the different features affected the outcome. Things like temperature, wind speed, and NDVI clearly had a big influence on how the model behaved. It made me realize that even small changes in environmental conditions can shift the results. I guess it helped me see wildfires as not just random events, but something that can be studied and maybe even predicted with the right tools.

That being said, I also noticed that building models isn’t always straightforward. Some parts were frustrating—like when the model misjudged cases

that seemed obvious, or when it got too focused on certain patterns and ignored others. But I also learned that this is pretty normal in data science. By trying things like feature selection and cross-validation, I gradually found better ways to deal with those problems.

Overall, this project wasn't just about getting a good accuracy score. It was also about learning how to work with messy data, how to question the results, and how to explain what a model is doing. I still have a lot to learn, but I feel like I've taken a good step in understanding how data science connects with real-life issues. And honestly, it made me more curious to dig deeper into these kinds of problems in the future.