International Conference on Recent Trends in Data Science and its Applications
DOI: rp-9788770040723.114

Forest Fire Prediction using Supervised Machine Learning Algorithms

Uddeshya Sharma
Data Science and Business Systems
SRM Institute of Science and Technology
Kattankulathur, India
us6487@srmist.edu.in

Sudipta Shaw
Data Science and Business Systems
SRM Institute of Science and Technology
Kattankulathur, India
ss5996@srmist.edu.in

K. Shantha Kumari
Data Science and Business Systems
SRM Institute of Science and Technology
Kattankulathur, India
shanthak@srmist.edu.in

Adarsh Shailendra
Data Science and Business Systems
SRM Institute of Science and Technology
Kattankulathur, India
as8969@srmist.edu.in

Chirag Bengani
Data Science and Business Systems
SRM Institute of Science and Technology
Kattankulathur, India
cb4849@srmist.edu.in

Shruti Ramesh
Data Science and Business Systems
SRM Institute of Science and Technology
Kattankulathur, India
sr9478@srmist.edu.in

***Abstract*—Nowadays, when global warming is increasing day by day, one of the most significant issues harming flora and animals around the planet today is wildfires. Forest fires are an uncontrollable disaster which causes damage to society as well as endangering nature. This paper uses machine learning regression techniques and artificial neural network algorithm for predicting the possibility of a forest fire to occur. This paper uses and compares various algorithms to try and deduce the best one possible out of all them. The various processes of a machine learning project are seen here and furthermore, techniques like cross-validation and hyperparameter tuning are carried out here.**

***Keywords*—cross-validation, hyperparameter tuning, CatBoost regression.**

I. Introduction

Wildfires are unplanned, uncontrolled, and unpredictable. As the name suggests they are often pretty huge and cover large swathes of an area on which there is combustible vegetation. Wildfires can be caused both naturally and by human intervention. The common natural cause for the start of a wildfire is generally lightning. Lightning strikes and then the dry fuel, mostly grass or twigs or other organic materials catches on fire. As for human causes, they range from slash-and-burn agriculture, a common practice in nomadic societies, to more serious methods like arson. But one of the most common ways a human can start a wildfire or a forest fire is the discarded cigarette butt after smoking.Across the world, there were about 4.2 million wildfires detected in 2019. Butwe are discussing only about the detected ones, which generally come into purview. Not to mention, the economic damage that can happen because of the fact that we lose multiple economic resources that are used in trying to extinguish the fires and also there can be massive economic damage due to the climate change threat.

Multiple continents experienced forest fires during the 2021 wildfire season. Even halfway through the year, there were more wildfires than ever before, and climate change-related increases in extreme weather events (such as droughts and heat waves) helped to amplify the size and scope of the fires. 2021 - Simlipal National Park in Odisha, Shimla-Kullu Wildfire, Dzukou Valley on Nagaland-Manipur borders.There are cases reported in Rajasthan' Sariska Tiger Reserve, Odisha's Similipal Wildlife Sanctuary, Madhya Pradesh's Ladkui jungles in Sehore district and the forest cover of the Majhgawan region of Satna district and the Perimalmalai Peak near the Kodaikanal hills of Tamil Nadu's Dindigul district were also struck with the disaster before the end of March, 2022.

Nowadays, forest fires have emerged as one of the most significant issues harming many regions worldwide. Wildfires are uncontrollable disasters which cause damage to society as well as endanger nature.

Why is there a need for ML when there is an accomplished forest department that has been handling these issues for quite some time? The answer is simple: ML can cover many parameters that the accomplished forest department cannot, such as latitude, longitude, satellite, version, and other variables, whereas the experienced forest department can only check on 3 to 4 parameters using human capability. In our use case ML actually helps out by covering various parameters that would otherwise not be covered due to the sheer information overload that would happen and would make it nearly impossible to actually deduce and come to a decision.

Our forest fire prediction technique uses and takes in several factors that can affect a fire and then output the confidence of the fire occurring. This dataset contains various features like the latitude, longitude, brightness, the terrain that is visible from the satellite, and the time of the day the data was collected, i.e., whether it was daytime or night, and other factors. Training, evaluating, and comparing the performances ofseveral ML algorithmsare carried out in this paper.In the training of the machine learning algorithms, we use two different datasets, that are curated with a few different features that are selected using the feature selection process and further processed using feature engineering, after which the data is finally split into training and testing data.We examine the effectiveness of the various methods using the R2 score, Mean Absolute Error, and Mean Squared Error as the various metrics and we also use a technique called Cross Validation that checks how the model performs on completely unseen data. Finally, hyperparameter tuning is conducted on the best-performing algorithm because hyperparameter tuning helps us in choosing the ideal set of hyperparameters. The hyperparameter is a parameter that controls the learning process, but by contrast, other parameters are learned. The tuning of the hyperparameters are absolutely critical because they affect and control the behaviour of the model. This is done in order to achieve higher accuracies and better performance from the model, as the default parameters may not suit the existing data due to variations in the problem statement.

II. Literature Review

A. *Forest Fires Detection Using Machine Learning Techniques*

In this paper[1], they have considered various climate and physical factors to be mapped. The models experimented with are Linear, Ridge, and Lasso Regression. The way of working was first to take all the features into account and only 70% the next time around. Out of the three models, Linear displayed the best results.

B. *A Perceptron Algorithm for Forest Fire Prediction Based on Wireless Sensor Networks*

They illustrate the use of perceptron algorithms [2] with wireless sensor networks to offer a quick and trustworthy technique to detect a potential forest fire early. Weather data like temperature, humidity, and many more are collected with the help of sensors. This data is then passed to a sensor which calculates a value called Fire Hazard Index.

C. *Evolution of Burned Area in Forest Fires under Climate Change Conditions in Southern Spain Using ANN*

This paper [3] is a study of the effectiveness of an artificial neural network in predicting the burned area and then using that information to evaluate how future wildfires will develop and the area they would affect in Southern Spain.

D. *Machine learning to predict final fire size at the time of ignition*

In this paper [4], an investigation is conducted into the size of a fire at the end and how accurately we can predict and control it at the time of ignition. With the help of decision trees, the fires are classified into small, medium, and large with a 50.4 ± 5.2% accuracy. This model predicted that 40% of the ignitions would develop into big fires, which would then account for 75% of the overall burned area. None of the other classification methods, including Random Forest and Multi-layer Perceptrons, performed as well as the decision tree approach in terms of output.

E. *Learning to predict forest fires with different data mining techniques*

Using a variety of data mining techniques, such as predictive modelling based on GIS of a forest structure, weather prediction model Aladin, and MODIS satellite data, a study [5] on how to anticipate forest fires in Slovenia was conducted. Decision trees, logistic regression, and Random Forest with bagging and boosting of decision trees were few of the Machine Learning techniques used. Bagging Decision Trees was the model with the best performance.

F. *Detection of forest fires using machine learning technique: A perspective*

In the quest to predict forest fires, a variety of machine learning methods, including SVM, regression, decision trees, and neural networks, have been experimented with [6]. The results of this paper explain why regression is the better approach to be taken in the detection of forest fires by dividing the dataset for higher accuracy. The paper mainly focuses on the quick detection of forest fires by completing the analysis before the other machine learning techniques.

G. *Burned area prediction with semiparametric models*

To describe and forecast the weekly burnt area, two semiparametric time-series models [7] are implemented and tested. The two models are Autoregressive moving average after smoothing and smoothing after Autoregressive moving average. They are examined and contrasted with a purely parametric model, and it is found that the first method yields results that are less error-prone than the second.

H. *Forest fire prediction using machine learning and deep learning techniques*

The goal of the paper [8] is to use various machine learning and deep learning approaches to predict the occurrence of forest fires. To determine the optimal model, a comparison analysis has been done. The Decision Tree has the highest accuracy of any model, at 79.6%. Additionally, the implementation included a User Interface for simple and clear access.

I. *Forest fire prediction using ML and AI*

In this paper [9], algorithms such as SVMs, KNN, Logistic Regression, Random Forest, as well as decision trees are implemented to describe a risk prediction method in the context of forest fires. Parameters such as humidity, temperature, and oxygen were considered as factors in forest fire prediction. The results of the paper indicate that the danger of the occurrence of forest fires can be predicted with an accuracy of up to 89.47% with the help ML machine learning models.

J. *Riau forest fire prediction using supervised machine learning*

In this study [10], data of the weather is used as a primary source to analyze forest fires and build early warning systems for the Indonesian island of Riau. Supervised ML techniques such as Bayesian networks and decision tree were utilized to build prediction models that provided an accuracy of 99%. Although decision trees were found to be slightly less accurate than Bayesian networks, the study found that both models were efficient in extracting relevant features in an effective manner.

K. *Artificial Intelligence for Forest Fire Prediction*

This paper [11] describes and analyzes methods to predict forest fires by leveraging artificial intelligence. The ideal model recommended utilizes support vector machine algorithm as the base algorithm for the prediction model, wherein past data is used to predict hazard levels for the day. The paper concludes that algorithms such as support vector machines can predict forest fire hazards with 96% accuracy even with a limited amount of data.

L. *Predicting Burned Areas of Forest Fires: an Artificial Intelligence Approach*

This research [12] was a study on how to create an intelligent system based on genetic programming to identify and forecast the area that burns using data specific to the forest under analysis and readily available meteorological data. Due to its smaller MAE in comparison to standard genetic programming and state-of-the-art machine learning methods, geometric semantic genetic programming was used, and the experiment results obtained were significantly better. This justifies further research into the geometric semantic method as it may be helpful for further learning deployment.

M. Forest Fire Prediction Using Machine Learning Techniques

This paper [13] is a comparative study of various models which are used to predict forest fires like Decision Trees, Random Forest, Support Vector Machines, etc., and to also compare with the RandomizedSearchCV algorithm used in the current study. The improvement proposed in this study is the usage of meteorological parameters like temperature, rain, wind and humidity. RandomizedSearchCV fits various decision trees together and uses averaging in order to improve the accuracy of prediction and also to control overfitting, a common problem noticed in various machine learning models. The results show that factors like extreme temperatures, moderate humidity and higher wind speeds exponentially increase the chance of burning in a forest fire. It is also noticed that compared to other areas, the forests are inclined to catch fire first.

N. Parallel SVM model for forest fire prediction

This paper [14] talks about a new method of trying and detecting forest fires with the help of Parallel SVM. It is a newer method which has been developed in order to tackle the problems of low efficiency and high overfitting which affect the actual real-world results. Parallel SVM intends to provide better performance also than conventional SVM, with the help of PySpark. The use of meteorological data is also done here and a better RMSE value is observed.

III. METHODOLOGY

O. Overview

The objective of this paper is to evaluate several ML algorithms such as linear regression, support vector machine, gradient boosting, and so on, and compare their performances. The best-performing algorithm can then be used to forecast the occurrence of a wild fire with confidence and reasonable accuracy. The overall methodology for this paper follows a streamlined path that helps in clarity of thought as you follow through. We start with standard procedures such as performing data pre-processing on imported data. We will then perform feature selection and feature engineering before splitting the data into training and test datasets. Next, we train the model, following which we validate the model with the various machine learning algorithms that we have proposed. Finally, we perform hyperparameter optimization. The problem is a regression task that takes the described features into account and provides an output that represents the chances of the occurrence of a forest fire.

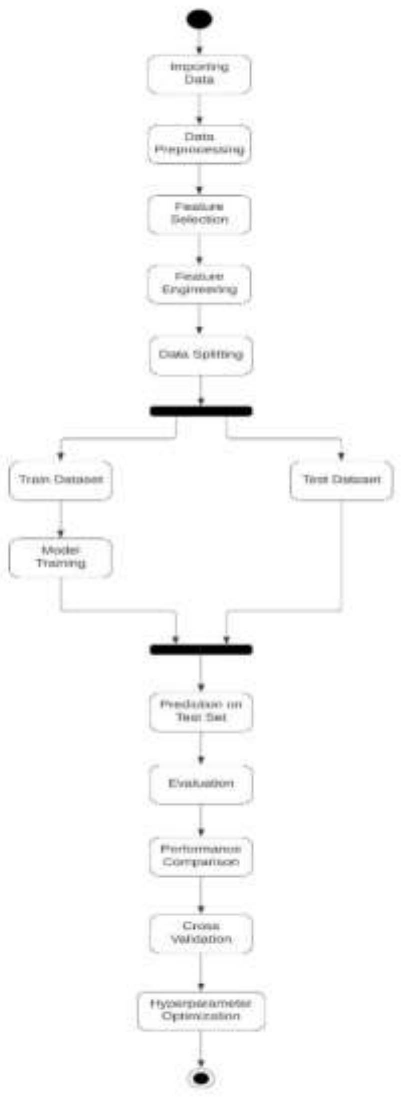

Fig. 1. MethodologyDiagram

P. Environment

The entire project is implemented on Google Colaboratory, commonly known as Google Colab. It is a free Jupyter notebook environment that runs on the cloud that allows users to write Python code through the browser. Hosted by Google, Colab provides access to powerful computing resources such as Graphical Processing Unit (GPU) and Tensor Processing Unit (TPU), which allows for seamless work on the platform. Google Colab is thus particularly useful for researchers focused on data science and machine learning, who wish to collaborate with others on coding projects in a shared environment without hindrances. Unfortunately, Google Colab might be unable to process larger datasets as it is a limited platform, and does not support all Python libraries. However, for the purposes of this paper, Google Colab is a suitable work environment which is more than capable of handling the necessary datasets and libraries. In order to increase the scalability of this project, environments such as Kaggle can also be utilized.

Q. Importing Data

The data is imported via a private link. Google Colab allows users to browse directories in the local system. The dataset contains 36011 rows and 15 columns. The features of the dataset that are taken into consideration include latitude and longitude, brightness, terrain from satellite, time of the day during data collection, fire radiative power, and fire type. This data is uploaded into Google Colab's workspace directory, which provides fast file access wherein uploaded files can be accessed without their absolute path.

R. Data Preprocessing

An important phase of any machine learning research is data preprocessing. It is a technique that involves transforming raw data into well-formed datasets and cleaning the raw data in order to check whether or not it is fit for analysis. More often than not, raw data is incomplete and inconsistent. Therefore, preparing the data has a direct impact on the final results post-analysis. The primary steps that are usually involved in data preprocessing are data cleaning, data integration, data transformation, data reduction, and data discretization. We first check the information generated by the dataset. This information primarily lets the user gain an understanding of the data type of each column as well as the total number of columns within the dataset, along with the count of non-null elements that are present in each column. This information helps make data cleaning easier. We use the describe() function, which is a function that comes with the pandas library, to check the statistical summary of the data. Doing so also helps us find any discrepancies in the data. Using the describe() function helps us in getting a quick understanding of the distribution and range of values in the dataset. We can additionally identify any potential outliers, missing values, or other problems pertaining to the quality of the data as a whole. In this case, no discrepancies are discovered. In order to check the count of null elements or corrupt elements in the columns of the dataset, we use the Python library missingno [reference]. The ability to understand the distribution of missing data is provided by this library. The visualisations can be done in form of bar charts or heat maps. It is possible to see where the missing values are and examine their relationships to the columns thanks to Missingno. It displays null values as a white line between a black column block. In this instance, we find that there are no null values or corrupt values in the dataset.

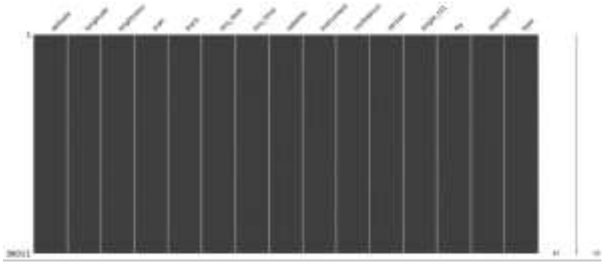

Fig. 2. MissingNo Evaluated Result

S. Feature Selection

Feature Selection is the process of selecting the most important features from a dataset to use in model construction. Feature selection helps to reduce the dimensionality of the dataset and refines the data to present only the most relevant variables. The goal of feature selection is to decrease the number of input variables to obtain only those variables which are the most useful in predicting the target variable. This can result in improved accuracy and performance of the model. By removing redundant or irrelevant features we can also reduce computational complexity and improve overall interpretability. At first, we remove the features acq_date, instrument, acq_time, and version. We do so as these features do not carry any relevance for the algorithm. Therefore, the model cannot be trained with these features. Once the aforementioned features have been removed, a correlation matrix is drawn out for the remaining features. A correlation matrix is essentially a table which describes the correlation coefficients between the various variables, or features in this case. A correlation matrix is often visualized in the form of a heatmap. A correlation matrix can help discern observable patterns within the data or can be used as inputs for further analyses. In this case, the correlation threshold is fixed at 0.6. This means that columns with a correlation coefficient of greater than 0.6 depict a high or strong correlation. These columns shall be dealt with. Based on the correlation observed, two datasets are created.

T. Feature Engineering

We perform feature engineering on both datasets that were generated after the process of feature selection. For dataset 1, One Hot Encoding is performed on the categorical columns of the dataset. One Hot Encoding allows the representation of categorical variables in the form of binary vectors. This can then be provided as input to machine learning models for further processes. In this case, One Hot Encoding is performed using the pandas function pd.get_dummies on Python. This function converts categorical data into “dummy” variables. As mentioned above, this dummy variable is binary and is either of the value 0 or 1. We follow a similar procedure for dataset 2, by performing One Hot Encoding on the categorical columns of the dataset with the help of pd.get_dummies. In addition to this, the range of the column scan is divided into separate categories using the method of binning. Binning is a method that reduces continuous and discrete data cardinality by grouping related values in “bins” so as to lower the number of distinct values in the data.

U. Data Splitting

As the name suggests, data splitting involves the splitting or dividing of data into subsets. Each subset serves a different purpose. For example, one subset can be used to train the machine learning model while the other subset can be used to test the model. The training subset can help develop the model while the test subset can assess the performance of the trained model. The most common method to split data into subsets is random sampling. In this case, we use the train_test_split function provided by sklearn library in Python. Sklearn, or Scikit-learn, is one of the most robust Python libraries that can be used abundantly in machine learning applications. We split the datasets such that the size of the test subset is 20% of themain dataset and the training dataset is 80% of the main dataset. We find that the training data has 28808 rows while the test data contains 720 rows in total. Since our aim is to predict the chances of confidence of the occurrence of a forest fire, our target column is “confidence”. This column will give us the probability of forest fire occurrence. Dataset 2 is scaled with the help of StandardScaler() function from the sklearn library. It is important to scale data before modelling in order to avoid issues such as misclassification or bias. By standardizing the data, we can scale the data with zero mean and unit variance. StandardScaler helps us achieve this.

V. Model Training and Prediction

We use the sklearn library for initializing various regression machine-learning algorithms. We also use this library to calculate the performance metrics such as r2 score, mean absolute error, and mean squared error. These performance metrics provide us with an insight into the performance of the model. First, the model is fit onto the training data, after which it is tested on the testing data. In this case, the machine learning algorithms used for training are linear regression, support vector regression, decision tree, random forest regression, gradient boosting, extreme gradient boosting, and catboostregressor.

Linear Regression - Linear regression is one of the most rudimentary machine learning algorithms. It is used to forecast the value of a continuous dependent variable based on the value of a continuous independent variable. It is a method for supervised learning that can carry out regression tasks.

- Support Vector Regression- SVR, also known as support vector regression, is a supervised machine learning technique that may be applied to regression tasks. It works on a similar principle to that of a traditional support vector machine and defines an acceptable error through a hyperplane.

- Decision Tree Regression - Decision trees work on the principle of supervised learning and can be used for both classification and regression tasks. Decision trees contain nodes such as the root node, interior nodes, and leaf nodes, as well as branches connecting these nodes in a tree-like structure. Decision trees are helpful as they are easy to understand and possess a rather simple structure.

- Random Forest Regression - For regression on a supervised method, Random Forest Regression uses ensemble learning. The mean of the classes is produced as a prediction of the trees after numerous decision trees have been built. This algorithm is extremely powerful and provides good performance on non-linear models as well. However, it is prone to overfitting.

- Gradient Boosting - Gradient boosting is a kind of ensemble method that provides high accuracy and prediction speed. The principle behind gradient boosting involves the sequential building of models to minimize error. Gradient boosting regressor is used when the target column is continuous and not discrete.

- Extreme Gradient Boosting- Extreme Gradient Boosting or XGB wherein decision trees are created sequentially. In XGB, weights are assigned to independent variables to predict results. XGB is also a supervised learning algorithm and is very commonly used in machine learning implementations as it is an optimized implementation of gradient-boosted trees.

- CatBoostRegressor- CatBoostRegressor is a class in CatBoost which performs regression tasks. CatBoost uses gradient boosting algorithm to handle categorical features without encoding and is an open-source library. It acts as an alternative to XGB and has a simpler tuning process for hyperparameters. Additionally, it is also much faster than XGB algorithms.

W. Evaluation

Once each model is trained, we evaluate its performance. We do so by calculating the performance metrics for the respective model. The performance metrics that we take into consideration for evaluation for this regression task are the r2 score, mean absolute error (mae), and mean squared error (mse). We calculate each model’s metric between the predicted values on the test dataset and the actual values of the target column of the test dataset.

X. Cross-Validation

The effectiveness of machine learning models is frequently assessed using the cross-validation method. Through a number of subset iterations, the data is split into various training and testing subsets. The performance of each iteration is considered, and the results are averaged out to provide an estimate of the model’s overall performance. Cross-validation is essential as it can provide a more accurate estimate of the model’s performance than a single training-testing iteration. Moreover, it plays a major role in reducing overfitting and identifying hyperparameters for the model. After comparing the performance of each of the models in our research based on their performance metrics, we perform k-fold cross-validation on our dataset. The k-fold cross-validation method is useful because it helps us to improve the model prediction in cases of insufficient data. This method helps us to determine the skill of any model on unseen data. The scoring is based on the r2 score obtained. We also compare the cross-validation mean of each algorithm with the corresponding cross-validation standard deviation. Based on this the cross-validation score is obtained, and the algorithm with the best performance is selected for hyperparameter tuning.

Y. Hyperparameter Tuning

Hyperparameter tuning essentially involves selecting the best hyperparameters for a certain machine-learning model. We set the hyperparameters before training as they are not learned by the model during training. It is necessary that hyperparameters must be selected such that they can optimize the performance of the model so that when hyperparameter tuning is carried out, only the hyperparameters that provide the best performance on the validation set can be found. We perform hyperparameter tuning with the help of GridSearchCV. This allows for an automated and systematic search for the optimal hyperparameters, which can improve the performance of the model, as well as improve its generalization. Once we have the best parameters, we conduct a final prediction and calculate the resulting performance metrics. Then, the generalization for both the training and testing data is calculated using the score() function.

IV. Results

The algorithms involved in this study were compared twice for each dataset. At first, the performance of the testing data was checked against the models, and then secondly, the performance is compared after the k-fold cross-validation method is applied.

Z. Testing Results

The testing results were obtained by first training each and every algorithm on the training data and then the predictions are made on the test data. The effectiveness of the algorithms can be assessed using a variety of metrics. R2 Score, Mean Absolute Error, and Mean Squared Error are used in this instance.

The R2 score, sometimes referred to as the coefficient of determination, is a statistical indicator of how much variance in the dependent variable in a regression model can be anticipated from the independent variable. R2 score is calculated as 1 - (SSres/SStot), where SSres stands for squared residual sum and SStot for square total sum.

Mean absolute error is a measure of errors between paired observationsthat can express the same phenomenon. It is considered the average of all absolute errors, which is calculated like this, (1/n) * Σ|yi – xi|, here ‘n’ represents the number of observations, ‘yi’ represents the predicted value and ‘xi’ represents the actual value.

Mean squared erroris used in machine learning as a method of evaluation of the performance of a regression model. The value corresponds to the expected value of the squared error loss, which corresponds to it being a risk function.

An R2 score of above 0.6 is generally considered to be a model which is useful, else it may not be worth trying to work upon. For dataset 1, which contained the columns ‘track’ and ‘brightness’, the performance of each algorithm is as follows in the table below

TABLE I. DATASET I SCORES

Algorithms	**R squared Score**	**Mean Absolute Error**	**Mean Squared Error**
CatBoost Regression	0.634611	9.887964	196.594296
XGB Regression	0.627698	10.003912	200.313794
Random Forest Regression	0.610143	10.045055	209.758887
Gradient Boosting	0.595831	10.702130	217.459224
Linear Regression	0.423432	13.804181	310.217179
Support Vector Regression	0.362626	13.433373	342.933564
Decision Tree Regression	0.306102	12.242677	373.345828

For dataset 2, which had the correlated columns ‘scan’ and ‘frp’, the performance of the algorithms on testing data are as follows

TABLE II. DATASET II SCORES

Algorithms	**R squared Score**	**Mean Absolute Error**	**Mean Squared Error**
CatBoost Regression	0.661975	9.408696	183.255066
XGB Regression	0.659588	9.356515	184.548962
Random Forest Regression	0.639109	9.642471	195.651349
Gradient Boosting	0.623193	10.263339	204.279866
Linear Regression	0.445729	13.603326	300.489397
Support Vector Regression	0.420726	12.563390	314.044553
Decision Tree Regression	0.313652	12.131751	372.093156

In both the cases, for Dataset I and II, we see that the catboostregressor was the one with the best performance, even after k-fold cross-validation. The XGB and Random Forest Regression algorithms are a close second and third in terms of results in R2 score.

AA. Cross-Validation Results

The next set of results were derived from after the k-fold cross-validationmethod. Cross Validation mean and Cross Validation standard deviation are calculated. Both are defined as the parameters that can be used to estimate a model’s performance.

For Dataset I, the results are as follows in the table

TABLE III. DATASET I SCORES

Algorithms	**Cross Validation Mean**	**Cross Validation Std**
CatBoost Regression	0.641353	0.017031
XGB Regression	0.636471	0.017743
Random Forest Regression	0.616798	0.020645
Gradient Boosting	0.601998	0.016819
Linear Regression	0.420587	0.013146
Support Vector Regression	0.361914	0.008568
Decision Tree Regression	0.296257	0.029647

For Dataset II, the results are also placed below

TABLE IV. DATASET II SCORES

Algorithms	**Cross Validation Mean**	**Cross Validation Std**
CatBoost Regression	0.620616	0.032393
Gradient Boosting	0.607242	0.027426
XGB Regression	0.605931	0.033517
Random Forest Regression	0.592208	0.030492
Linear Regression	0.432178	0.027499
Support Vector Regression	0.414495	0.029684
Decision Tree Regression	0.251440	0.045719

As we see, CatBoost Regression is again far ahead of the other algorithms that are taken in consideration and this can be taken as proof that CatBoost works as the best-performing algorithm from all others.

We now find the best parameters for CatBoost to work on with the help of GridSearchCV, a hyperparameter tuning library that presents us with the best values possible. For

Dataset I and II, the best parameters were a depth of 15, iterations as 100 and a learning rate of 0.2. These values are auto-selected and tuned to provide the best-performing model possible.

We also see a bump in the scores and values for Dataset II after the cross-validation and a stable value for Dataset I.

V. Conclusion

Our paper is a deep dive into the various algorithms that can be used to detect wildfires and to estimate the accuracy of a forest fire to occur. We have used different methods, ranging from linear regression to CatBoost Regression to try and derive meaningful and sensible results. We start by performing data preprocessing, a crucial step to make the data that is available to us of use to us. We then continue on to making Features from it, deciding what is required and what is not. Then we move to the data splitting, from which we then go to the training and prediction scenarios, in which multiple models are selected and trained upon. The models are then evaluated and then finally, Cross Validation and Hyperparameter Tuning take place to finalize all the details for the model that is the best, which is the CatBoost model, which is because of the fact that CatBoost uses both the method of decision trees and gradient boosting. Boosting is a method which combines multiple weaker models and with the help of greedy search, create a strong and competitive model. Gradient Boosting also helps in rapidly reducing errors by sequential fitting.

Another improvement that can be made is the building of a web application that can be mass-distributed to get even laymen, who are not interested in the know-how of the model to use the model to get their predictions

References

[1] Elshewey, Ahmed and Elsonbaty, Amira, “Forest Fires Detection Using Machine Learning Techniques,” Xi'an Jianzhu Keji Daxue Xuebao/Journal of Xi'an University of Architecture & Technology. XII. 2020.

[2] Zhu, Haoran, Gao, Demin, Zhang, Shuo,”. A Perceptron Algorithm for Forest Fire Prediction Based on Wireless Sensor Networks,” Journal on Internet of Things, vol. 1, pp. 25-31, 2019, 10.32604/jiot.2019.05897

[3] Pérez-Sánchez, Julio, Jimeno-Sáez, Patricia, Senent-Aparicio, Javier, Díaz-Palmero, José, Cabezas-Cerezo and Juan,“Evolution of Burned Area in Forest Fires under Climate Change Conditions in Southern Spain Using ANN,” Applied Sciences, vol. 9, p. 4155, 2019, 10.3390/app9194155

[4] S.Coffield, C.Graff, Y.Chen, P.Smyth, E.Foufoula-Georgiou,and J. Randerson,“Machine learning to predict final fire size at the time of ignition,” International Journal of Wildland Fire, vol. 28, no. 11, pp.861-873, 2019, http://dx.doi.org/10.1071/wf19023 Retrieved from https://escholarship.org/uc/item/9v84h13m

[5] Stojanova, Daniela, Panov, Pance, Kobler, Andrej, Džeroski, Sašo, Taškova, and Katerina,“Learning to predict forest fires with different data mining techniques,” 2006.

[6] Gomathy, V., Janarthanan, K., Al-Turjman, F., Sitharthan, R., Rajesh, M., Vengatesan, K., &Reshma, T. P. (2021). Investigating the spread of coronavirus disease via edge-AI and air pollution correlation. ACM Transactions on Internet Technology, 21(4), 1-10.

[7] Boubeta, Miguel, Lombardía, María Jose, Gonzãlez-Manteiga, Wenceslao, Marey-Perez, and Manuel,“Burned area prediction with semiparametric models,” International Journal of Wildland Fire, vol. 25, 2016, 10.1071/WF15125.

[8] Rajesh, M., &Sitharthan, R. (2022). Introduction to the special section on cyber-physical system for autonomous process control in industry 5.0.Computers and Electrical Engineering, 104, 108481.

[9] Adithi M. Shrouthy, Syed Matheen Pasha, Yamini S. R. E., Navya Shree S., Lisha U.. Forest fire prediction using ML and AI, International Journal of Advance Research, Ideas and Innovations in Technology, www.IJARIIT.com.

[10] B. S. Negara, R. Kurniawan, M. Z. A. Nazri, S. N. H. S. Abdullah, R. W. Saputra and A. Ismanto, “Riau Forest Fire Prediction using Supervised Machine Learning”, November 2019.

[11] G. E. Sakr, I. H. Elhajj, G. Mitri and U. C. Wejinya, "Artificial intelligence for forest fire prediction," 2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Montreal, QC, Canada, pp. 1311-1316,2010, doi: 10.1109/AIM.2010.5695809.

[12] M. Castelli, L. Vanneschi, and A. Popovič, “Predicting Burned Areas of Forest Fires: an Artificial Intelligence Approach,” Fire Ecology, vol. 11, no. 1, pp. 106–118, Apr. 2015, doi: https://doi.org/10.4996/fireecology.1101106.

[13] T. Preeti, S. Kanakaraddi, A. Beelagi, S. Malagi and A. Sudi, "Forest Fire Prediction Using Machine Learning Techniques," 2021 International Conference on Intelligent Technologies (CONIT), Hubli, India, 2021, pp. 1-6, doi: 10.1109/CONIT51480.2021.9498448.

[14] Kajol R Singh, K.P. Neethu, K Madhurekaa, A Harita, Pushpa Mohan, Parallel SVM model for forest fire prediction,Soft Computing Letters, vol. 3, p. 100014,ISSN 2666-2221,2021,https://doi.org/10.1016/j.socl.2021.100014.