Predicting Wildfires Using Machine Learning Models

Kazi Md. Al-Wakil | Anika Tahsin | Ankita Roy | Sababa Rahman Meem
School of Data and Sciences
Brac University
Dhaka, Bangladesh
kazi.md.al.wakil@g.bracu.ac.bd, anika.tahsin5@g.bracu.ac.bd,
ankita.roy.ponty@g.bracu.ac.bd, sababa.rahman.meem@g.bracu.ac.bd

Index Terms—Machine Learning, Regression, Wildfire Prediction

I. Introduction

Scholars and forest managers have been intrigued for many years by the difficulty of predicting the incidence of wildfires. Attempts to understand the conditions that lead to the ignition and spread of wildfires can be dated at least 150 years back [1].
If a wildfire continues to spread uncontrollably due to strong winds, and starts affecting nearby fields or farms, we might face unavoidable famines due to the destruction of crops and other food resources. Loss of property, and food resources also direct to economical mayhem which may lead to instability in local labor markets. Wildfires have a very negative impact on our climate and weather as it releases large quantities of greenhouse gasses such as carbon dioxide, carbon monoxide, and fine particulate matter into the atmosphere. Due to climate change, forest fires are likely to become more frequent, and more severe, and cause billions of dollars in damage yearly. For the prevention of disasters and environmental protection, wildfire forecasting is vital.
In recent years, Artificial Intelligence (AI) models have proven to be very effective for predicting natural hazards [2]. The models applied to predict forest fires just forecast the parameter of the area impacted by fires because the study of forest fire prediction has mostly been concentrated on the prediction of forest fire frequency [3], [4].
In this wildfire prediction study, the data were quantitative, numerical and categorical data to predict forest wildres. We implemented a Regression model and Neural Network model to predict the scale of wildfires. The main objective of this study is to investigate the capability of regression (i.e., Linear, Ridge, Lasso, Decision Tree, Random Forest, XGBoost) and a neural network model to predict the scale of wildfires in forests using quantitative data.

The forest fire has become a threat not only to the forest wealth but also flora and fauna and ecology of the environment. All the authors more or less aimed to not just determine if a forest fire will take place or not, they predicted the confidence of the forest fire based on some attributes in different cases and areas of forest fire.

II. Software

Data collection and preprocessing, and the establishment of the regression and neural network models, were developed and implemented in the TensorFlow framework of the Anaconda 3 software (Anaconda, Inc.).

III. Abbreviation

Names	Abbreviation
Machine Learning	ML
Neural Network	NN
Artificial Neural Network	ANN
Mean Square Error	MSE
Root Mean Square Error	RMSE
Ordinary Least Squares	OLS
Decision Tree	DT
Support Vector Machine	SVM
Support Vector Regression	SVR
Variance Inflation Factor	VIF
Cross-validation	CV
K-Nearest Neighbours	KNN
Exploratory Data Analysis	EDA

IV. Research Methodology

A. Dataset Description

The data was collected from the United States Department of Agriculture Forest Service (USDA) [5], [6].

The data consists the Monitoring Trends in Burn Severity (MTBS) program which assesses the frequency, extent, and magnitude (size and severity) of all large wildland fires (including wildfires and prescribed fires) in the conterminous United States (CONUS), Alaska, Hawaii, and Puerto Rico from the beginning of the Landsat Thematic Mapper archive to the present. All fires reported as greater than 1,000 acres in the western U.S. and greater than 500 acres in the eastern U.S. are mapped across all ownerships.

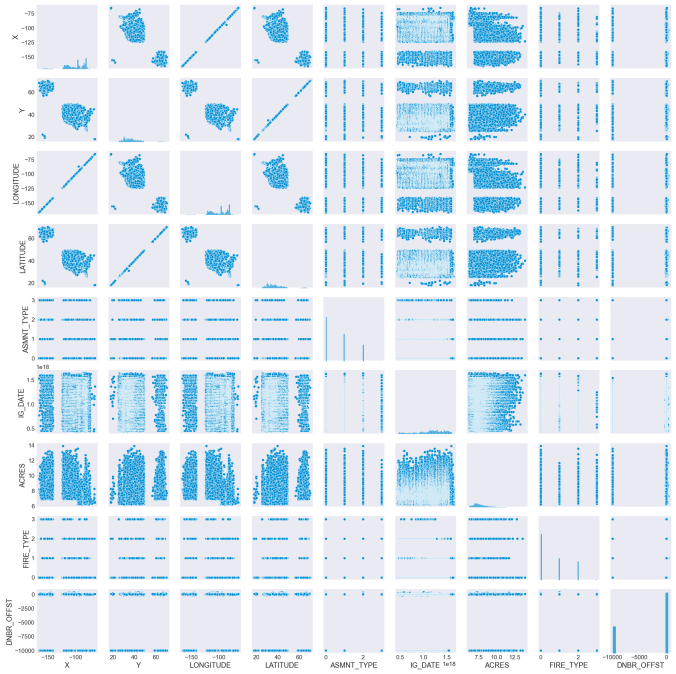

Fig. 1. Pairplot For Dataset

B. Data Filtering and Handling

A total of 29,533 fire data with 25 feature records were obtained from the year 1984 to 2021. The data is arranged by Acres, Threshold, Latitude, Longitude, Fire Type, Coordinates. It also consists of comments and ID series of Object, Fire, Area, Map. Since the ID features will not affect the target feature, these were dropped using numpy.

The data also had several missing values along with some redundant features for Wildfire prediction. To handle missing data and Na/NaN values of the features, Pandas was used to deal with numeric missing values. For the categorical features, there were 4 unique values which were encoded into indicator variables using Pandas. Exploratory Data Analysis (EDA) was used to analyze data, correlation of the target feature was shown with correlation matrix using heatmap.

The ID features were dropped and stored for future prediction.

C. Finding the Target Variable

To start with, we selected the ACRES feature for prediction at random to run the models, but this gave a huge root mean squared error. Since RMSE is better near zero, We tested out Pearson and Spearman correlations to determine what feature to choose as the target variable.

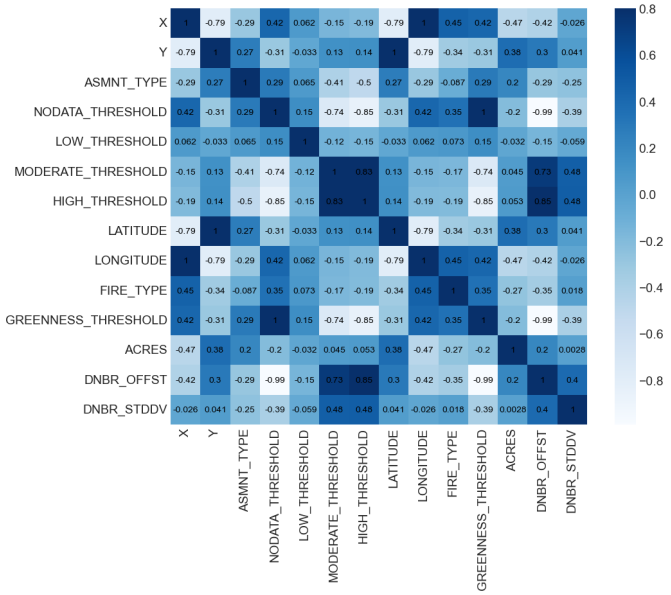

Fig. 2. Correlation Matrix

From the correlation matrix, there were 6 pairs of features that were either 1 or were very close to 1. The pair of features and their respective values were,

- X, LONGITUDE = 1
- Y, LATITUDE = 1
- GREENNESS_THRESHOLD, NODATA_THRESHOLD = 1
- HIGH_THRESHHOLD, DNBR_OFFST = 0.85
- HIGH_THRESHHOLD, MODERATE_THRESHOLD = 0.83
- MODERATE_THRESHOLD, DNBR_OFFST = 0.73

After testing the Pearson and Spearman correlation, the following results were found,

Test	X, Y	Nodata with High	Moderate with DNBR	Moderate with High	High with DNBR
Pearson	-0.792	-0.85	0.73	0.83	0.84
Spearman	-0.707	-0.83	0.642	0.88	0.66

Since, in each pair the Pearson test had better outcomes than Spearman so We leaned towards the Pearson results. As for DNBR_OFFST the Pearson test result were better than the rest, so We chose the feature DNBR_OFFST to be target. This feature did give a better RMSE than

ACRES but it still needed some modification.
So We went back to the feature 'ACRES' again. But this time, before multicollinearity test, We standardized the data and plotted it to find the value density to be between 0 to 1. Finally, standardization helped in finding the target feature, 'ACRES'.

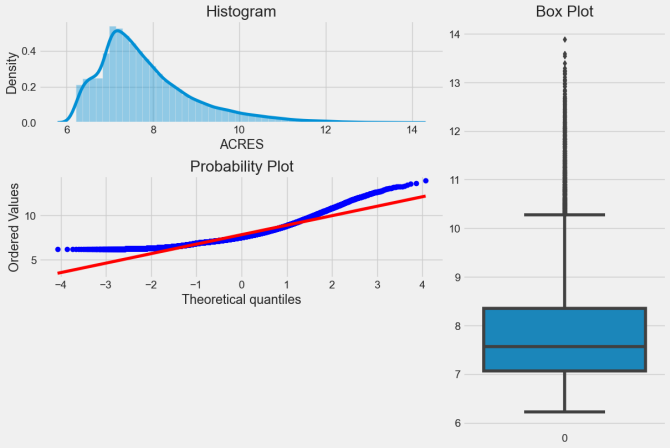

Fig. 3. Standardized Target feature 'ACRES' data

D. Multicollinearity Test

Strong correlations between the explanatory variables in a regression model are referred to as multicollinearity, and they can lead to departures from the actual data and distort the model's estimates [7].
The accuracy of the wildfire scale prediction model should not be adversely affected by factors with significant collinearity. So, We applied a correlation matrix and selected those independent variables with high correlation with dependent variables. We set the high correlation to be >0.1.

E. Models

In this study, two types of models were employed to evaluate the likelihood of wildfire occurrence. The first group of ML models were the regression models which predict the ACRES for wildfire. The objective function for wildfire scale prediction modeling was the root mean square error (RMSE) measuring the magnitude of the error between the observations and predictions.
The wildfire dataset was randomly split into two datasets for training (80% of data) and test (20% of data) before fitting the models and predicting on the target value.

Application of Regression Models

1) Linear Regression: To start with, We used Linear Regression on our preprocessed data, which makes the assumption that the predictors and target variable have a linear relationship, here 'ACRES'. The linear regression can be expressed in the following form:

$$Y \approx \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n$$

Fig. 4. Linear Regression Equation

Where a = coefficients, x = features, b is the parameter of the model. The model's parameters a and b are chosen using the Ordinary Least Squares (OLS) method. It operates by reducing the sum of residuals' squares (actual value - predicted value).

2) Ridge Regression: An approach to regularization is called Ridge Regression. Ridge regression is a linear regression extension in which the loss function is changed to reduce the model's complexity. The modification is accomplished by including a penalty parameter that is equal to the square of the coefficients' magnitude. A high alpha can result in underfitting, so We set the parameter alpha with a value of 0.001 for the loss function computation.

$$\sum_{i=1}^{M} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{M} \left(y_i - \sum_{j=0}^{p} w_j \times x_{ij} \right)^2$$

Fig. 5. Ridge Regression Equation

3) Lasso Regression: Lasso Regression is a regularization technique. Another adaptation of the linear regression is lasso regression, commonly known as the Least Absolute Shrinkage and Selection Operator. By limiting the sum of the absolute values of the model coefficients, Lasso modifies the loss function to reduce the model's complexity. In this case, the parameter alpha is the same as used in the Ridge regression model.
Hyper Parameter Tuning: The ridge, lasso models are tuned by various hyperparameters. We performed hyperparameter optimization for both models to achieve optimal predictions. The Grid Search Cross-validation method present in the sklearn library was used to find the best parameters of ridge, lasso. The following hyperparameters were examined: tuned alpha, cv = 10, n_jobs = -1, verbose = 1.
The accuracies presented in the Results section are of the optimal hyperparameters.

4) Decision Tree Regression: Decision Tree [8] is a Supervised learning technique that can be used for both

classification and Regression problems. The decision node is where the tree divides into different branches, with each branch denoting the specific decision the algorithm is making and leaf nodes denoting the results of the model. The objective of DTs is to accurately capture the relationships between input and outputs using the smallest possible tree that avoids overfitting. In a decision tree, the algorithm starts with X_train and y_train as parameters with a cross validation of 5. This is done to predict the class of the given dataset.

5) Random Forest Regression: Random Forest [9], a well-known machine learning algorithm that falls under the category of supervised learning, is an ensemble model made up of several individually trained Decision Trees (DTs). This approach achieves excellent performance by minimizing correlation across trees and reducing model variance, resulting in a huge number of diverse trees that are more accurate than individual trees.

6) K-Nearest Neighbor: K-nearest neighbor algorithm is utilized for grouping and used in pattern recognition. It is widely used in predictive analysis. On the arrival of new data, the K-NN algorithm identifies existing data points that are nearest to it. Any attributes that can differ on a large scale may have sufficient influence on the interval between data points. Given a positive integer k, k -nearest neighbors looks at the k observations closest to a test observation x_0 and estimates the conditional probability that it belongs to class j using the formula

$$Pr(Y = j|X = x_0) = \frac{1}{k} \sum_{i \in \mathcal{N}_0} I(y_i = j)$$

Where $mathcalN_0$ is the set of k-nearest observations and I($y_i = j$) is an indicator variable that evaluates to 1 if a given observation in $mathcalN_0$ is a member of class j, and 0 if otherwise. The following hyperparameters were examined: 'n_neighbors': 11, 'p': 1, 'weights': 'uniform' The accuracies presented in the Results section are of the optimal hyperparameters.

7) Support Vector Machine: Support Vector Machine is a supervised learning model used for analyzing both classification and regression data. The idea is to find a line or hyperplane with the help of a kernel trick to separate two classes in a dimension. Support Vector Regression (SVR) is a part of the Support Vector Machine (SVM) specially used in regression analysis. SVR tries to find a suitable function which will approximate the relationship between the features and the continuous label values as well as minimize the error while predicting.

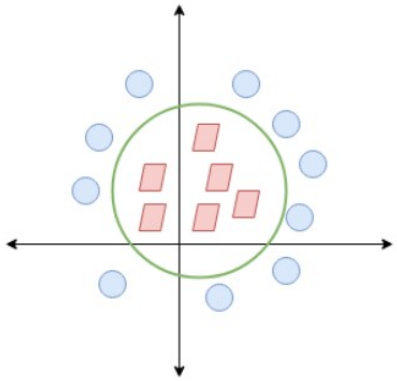

Fig. 6. Support Vector Regression

8) AdaBoost: AdaBoost is a popular boosting technique which helps to combine multiple "weak classifiers" into a single "strong classifier". A weak classifier is simply a classifier that performs poorly, but performs better than random guessing. AdaBoost can be applied to any classification and regression algorithm, so it's really a technique that builds on top of other classifiers as opposed to being a classifier itself.
The final classifier consists of 'T' weak classifiers. h_t(x) is the output of weak classifier 't' (in this paper, the outputs are limited to -1 or +1). Alpha_t is the weight applied to classifier 't' as determined by AdaBoost. So the final output is just a linear combination of all of the weak classifiers, and then we make our final decision simply by looking at the sign of this sum.

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

Fig. 7.

9) XGBoost Regression: XGBoost [10] is a powerful ensemble learning approach that can be used directly for building supervised regression models. By knowing about its objective function that contains loss function and a regularization term, and also base learners, the validity of this statement can be inferred. It tells about the model results from the real values, which is mainly the difference between actual and predicted values.

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^{n} [g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i)] + \Omega(f_t)$$

Fig. 8. XGBoost Simplified Objective

The second model We employed was the Artificial Neural Network (ANN).

Application of Neural Network Model

Artificial Neural Networks are functions that attempt to mimic the way a human brain makes decisions. They are particularly useful when dealing with complex non-linear classification problems [11].
In this study, the ANN architecture used was that of the Multilayer Perceptron. In the algorithm, a hidden layer node j gets the values of a group of independent variables (x_i to x_p) from an input layer. Then each value is multiplied by a weight w_h and then added together to produce a value u_j. This value is transferred through a non-linear sigmoid transfer function, f(x), to create the value h_j, which is then subjected to additional weighting before being sent to the output layer. Finally, the weighted v_j values are added together and passed into another sigmoid transfer function f(x), which outputs the final y_k values of the model [12].

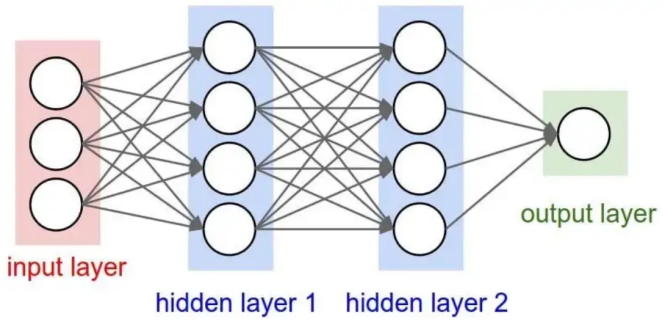

Fig. 9. Artificial Neural Network Layers

The ANN model was also tuned by various hyperparameters to optimize the model to achieve optimal predictions. Grid Search Cross-validation method present in the sklearn library to find the best parameters of ANN. The following hyperparameters were examined: a regression ANN model as estimator, batch size = [10,20,30], epochs between 10 and 20, Optimizer_trial as adam and rmsprop, cross validation = 5, and a customized scoring with greater accuracy score.

F. Model Accuracy

We evaluated the accuracy of the aforementioned models by measuring the mean-square error (MSE) and the root-mean-square error (RMSE) for each model. For Ridge and Lasso We did hyper parameter tuning and measured the Root Mean Squared Error (RMSE) with the new parameters. The R-squared value was also calculated in the study which is a measure that provides information about the goodness of fit of a model.

R-squared was calculated using [13],

$$R^2 = 1 - \frac{\text{sum squared regression (SSR)}}{\text{total sum of squares (SST)}},$$
$$= 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}.$$

Fig. 10. R-squared Equation

RMSE was calculated using the equation [14],

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(f_i - \bar{f}_i)^2}{N}}$$

Fig. 11. RMSE Equation

V. Results and Comparison

In this section, We are presenting the results of various ML regression and NN models which estimate the ratio of burned areas for each observation. From the results, we found that Random Forest outperformed all the other models, and from all models, XGBoost and Random Forest had a very close RMSE. The proportion of the variance, R-squared value of Random Forest was slightly higher than XGBoost. The model XGBoost regression gave a RMSE of 0.95 (95%), and Random Forest regression gave a RMSE of 0.936 (93.6%).

Models	R-squared	MSE (km^4)	RMSE (km^2)	Parameter Tested	Best Parameter
Linear	0.098	1.178	1.096	-	-
Decision Tree	-0.258	1.67	1.293	-	-
Random Forest	0.343	0.867	0.936	-	-
SVR	0.00052	-	1.154	C, ϵ, γ	1, 1, 100
KNN	0.138	-	1.072	n_neighbors, p, weights	11, 1, 'uniform'
Adaboost	-	-	-	-	-
Ridge	0.248	1.049	1.001	α	α: 10.0
Lasso	0.247	0.997	1.002	α	α: 0.0
XGBoost	0.319	0.897	0.953	-	-
ANN	-0.0005	1.334	1.155	-	-

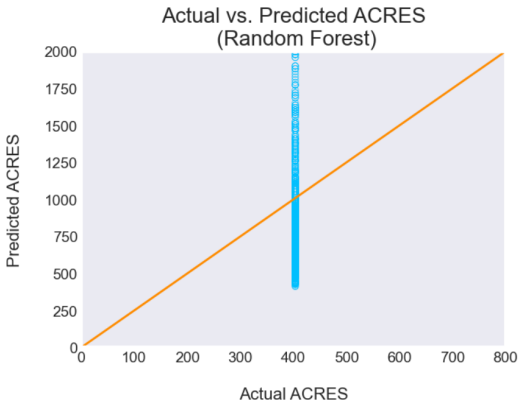

Fig. 12. Random Forest Regressor Prediction Plot

VI. Discussion

In this paper we applied multiple ML models and a NN model to predict the occurrence and size of Wildfires from the collected dataset.

The models included linear, ridge, lasso, decision tree, random forest, XGBoost, and ANN regressions. The models were trained using a large dataset which includes wildfire observations from the year 1984 to 2021. Since we had to select the target, it took a good amount of time just to find the target feature. we predicted using the ACRES at first, but that gave a large RMSE. After exploring the correlation matrix, we applied Pearson correlation test on a few features and found a better suited target variable, DNBR_OFFST. But then we standardized the ACRES feature and finally used it as the target variable for prediction. Upon training, the best models showed promising prediction accuracies and could predict the likelihood of wildfires with burned area ratios with RMSE scores of 0.95 km^2.
One of the most troubling issues we faced was hyper parameter tuning Artificial Neural Network. A promising prospect to enhance wildfire alerts and give forest managers tools to analyze regional wildfire risk is accurate wildfire hazard estimation by Machine Learning models.

VII. Conclusion

The ACRES provided a flexible, robust, analytically simple approach that could be applied anywhere within any continents.

The study's findings show that it is possible to forecast the size of wildfires using quantitative, numerical, and categorical data. This information will be useful for forest fire prevention and rescue, particularly for wildfires that start in forests. The results of this research demonstrate the superiority of machine learning (ML) models over conventional fire weather indices for estimating wildfire hazard.

References

[1] U. A. S. Service, "Report on the michigan forest fires of 1881." Office of the Chief Signal Officer:Washington, DC, USA, 1881, p. 37.

[2] M. G. Rollins, P. Morgan, and T. Swetnam, "Landscape-scale controls over 20th century fire occurrence in two large rocky mountain (usa) wilderness areas," Landscape Ecology, vol. 17, no. 6, pp. 539–557, 2002.

[3] D. T. Bui, B. Pradhan, H. Nampak, Q.-T. Bui, Q.-A. Tran, and Q.-P. Nguyen, "Hybrid artificial intelligence approach based on neural fuzzy inference model and metaheuristic optimization for flood susceptibilitgy modeling in a high-frequency tropical cyclone area using gis," Journal of Hydrology, vol. 540, pp. 317–330, 2016.

[4] H. Hong, M. Panahi, A. Shirzadi, T. Ma, J. Liu, A.-X. Zhu, W. Chen, I. Kougias, and N. Kazakis, "Flood susceptibility assessment in hengfeng area coupling adaptive neuro-fuzzy inference system with genetic algorithm and differential evolution," Science of the total Environment, vol. 621, pp. 1124–1141, 2018.

[5] fs.usda, "https://data.fs.usda.gov/geodata/edw/datasets.php."

[6] USDA, "https://catalog.data.gov/organization/usda-gov."

[7] H. Zainodin, A. Noraini, and S. Yap, "An alternative multicollinearity approach in solving multiple regression problem," Trends in Applied Sciences Research, vol. 6, no. 11, p. 1241, 2011.

[8] D. H. Moore, "Classification and regression trees, by leo breiman," 1984.

[9] B. Leo, "Statistical modeling: The two cultures," Statistical science, vol. 16, no. 3, pp. 199–231, 2001.

[10] J. Brownlee, XGBoost With python: Gradient boosted trees with XGBoost and scikit-learn. Machine Learning Mastery, 2016.

[11] A. Alonso-Betanzos, O. Fontenla-Romero, B. Guijarro-Berdiñas, E. Hernández-Pereira, J. Canda, E. Jimenez, J. L. Legido, S. Muñiz, C. Paz-Andrade, and M. I. Paz-Andrade, "A neural network approach for forestal fire risk estimation," in Proceedings of the 15th European Conference on Artificial Intelligence, 2002, pp. 643–647.

[12] S. Haykin, Neural networks and learning machines, 3/E. Pearson Education India, 2009.

[13] NCL, "https://www.ncl.ac.uk/webtemplate/ask-assets/external/maths-resources/statistics/regression-and-correlation/coefficient-of-determination-r-squared.html."

[14] statisticshowto, "https://www.statistic-showto.com/probability-and-statistics/regression-analysis/rmse-root-mean-square-error/."