Project #12: Inferno Alert: Wildfire Prediction System

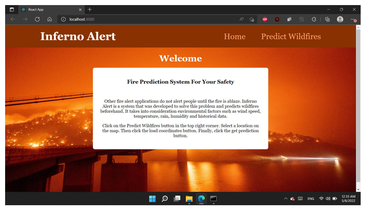
Project Advisor: Prof. Ahmed Banafa

Software Engineering Department

Patel, Deep (BS Software Engineering)
Patel, Nimay (BS Software Engineering)
Solanki, Jay (BS Software Engineering)
Trivedi, Soumya (BS Software Engineering)

Introduction

The United States has witnessed more than 1.8 billion acres of land getting burnt this year alone and the number is suspected to increase due to global warming. Predicting such fires beforehand will not only reduce the cost to combat these fires by effective allocation of resources but will also save thousands of lives. Existing fire alert applications don't alert people until the fire is ablaze. There are many options available for civilians to get live updates regarding active wildfires but unfortunately, they don't predict future instances.

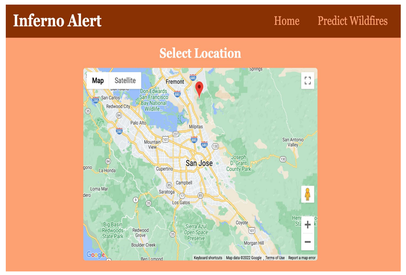


The project aims to build upon the customized dataset tailored specifically from existing fire detection models and datasets to include weather information that has the most impact on the wildfire. The project takes into consideration the environmental factors such as: wind, temperature, humidity and rain to calculate Fire Behavior Indices which are used to make as accurate a prediction as possible displaying its results on a web application.

Methodology

UI and API

React.js which is a JavaScript library was used to build the front-end of the system. Code was written using JavaScript, JSX, and CSS. Node.js which is a JavaScript runtime was used to execute the code. The npx create-react-app command was used to get the build setup. A map was integrated into the web application with the help of Google Maps Platform. The ability to drop a marker on the map was provided to the user.

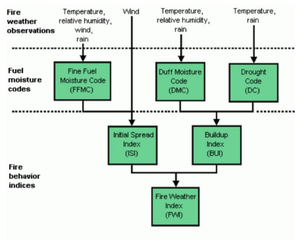


The coordinates received from the marker were then stored into a localStorage object. Functionality was provided to load these coordinates into a Weather API requested with the help of Axios from OpenWeatherMap. Axios is a JS dependency that helps make HTTP requests from the browser. Another important dependency this project utilised was react-router-dom which gives the ability to access different pages of the UI by providing routing.

Flask is used as a service layer to interact with the front-end React and backend Machine Learning. The application contains a python file where post and get methods are defined to fetch the data from the user which are weather and location.The python files contain certain formulas that are defined to calculate the moisture code, duff code and fire weather index which takes in input from the machine learning model to display the user with a Fire or No Fire message through the front-end.

Dataset and Machine Learning Algorithms

The dataset was custom created by merging elements from research datasets on Canadian Forest fires, with data provided by national wildlife agency of Algeria along with information recorded by Portuguese National Park Service. The dataset has 793 individual entries and the corelation between all of them can be identified using the following diagram that shows how to calculate behavior index.



Gathering the data, importing the Python libraries, dealing with missing values, dividing the dataset into independent and dependent variables, dealing with categorical values, removing unnecessary data, and partitioning the dataset into training and test sets were the procedures involved in preparing the dataset.

The four main modeling methods that we used for analysis are K-Nearest Neighbors , Decision Trees, Support Vector Machines, and Random Forest.We got very accurate and precise findings by applying these Machine Learning Models.

Analysis and Results

Data Analysis:

	class	temp	rain	wind	RH	FFMC	DMC	DC	ISI	BUI	FWI
1.	not fire	39	1.62	1.57	90	65.32	6.45	21.79	1.28	7.22	0.96
2.	fire	42	0.04	3.49	100	89.99	91.94	447.33	8.69	27.23	9.78

Upon data analysis, the correlation between the Fire/No Fire class and weather parameter is crystal clear. It is safe to establish that high temperature, wind, humidity and low rain causes higher FFMC, DMC, DC which results in high ISI, BUI, FWI and in turn, a higher risk of fire. This can be used to train ML model to accurately predict wildfire.

We compute machine learning predictions utilizing independent variables such as ISI, BUI, and FWI. To further reduce outliers, our algorithm used the boxplot chart to show them and remove them.

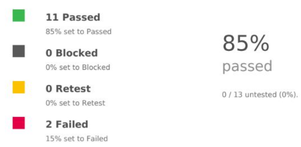
Decision Tree	95.1724	0.9517
Random Forest	96.5517	0.9655
Logistic Regression	95.8621	0.9586
Support Vector Machines	96.5517	0.9655
Naive Bayes	95.1724	0.9517
k Nearest Neighbors	95.1724	0.9517

We trained our models by 80% and then tested them with 20% of the dataset. In our training data, we trained Decision Tree Classifier, Random Forest Classifier, Logistic Regression, Support Vector Machine, Naive Bayes, and K Nearest Neighbors.

For our project we were able to achieve the prediction upto 96.55% by deploying Random Forest and Support Vector Machine.

Testing

For testing this web application,we first created 3 sections depending on our criteria in TestRail and wrote test cases (i.e. specifying preconditions, steps to follow and expected results). Based on these, we used Selenium IDE to write test scripts for all test cases. But execution of these scripts was a little complicated as our application is running on localhost, so we integrated our Selenium IDE project with BrowserStack Automate for this compatibility purpose.



Function	Cost
Discussion	250 minutes
Test methods	200 minutes
Test case	150 minutes
Test Data	40 minutes

Above given is the summary of the cost of doing testing. Each of 13 cases are made based on 13 requirements set at the beginning. We are happy to report that all requirements marked critical and high priority have been fulfilled and 2 failed test cases were for low priority requirements. No critical component failed testing.

Summary/Conclusions

Our application lets all customers select any location and make a prediction on whether the risk of fire is there or not using the current weather conditions in the location. Our main focus, building this application is to help civilians by informing them to stay away from vulnerable areas. This application is built to provide services, not just in the USA but for the world. In a distant future, we believe that there is room for analyzing historic weather data precisely for these prediction clusters over a period of years and determine which locations are not for human inhabitation.

Key References

[1] *Van Wagner, C.E.; Pickett, T.L. 1985. Equations and FORTRAN program for the Canadian Forest Fire Weather Index System. Canadian Forest Service, Ottawa, ON. Forestry Technical Report 33.*

[2] Sayad, Y. O., Mousannif, H., & Al Moatassime, H. (2019). Predictive modeling of wildfires: A new dataset and machine learning approach. Fire Safety Journal, 104, 130–146.

[3] J. Brownlee, "How to Connect Model Input Data With Predictions for Machine Learning," *Machine Learning Mastery*, Nov. 14, 2019.

[4] A. Srinivasan, "Leveraging Machine Learning to predict wildfires: Contributing to the United Nations Sustainable…," *Medium*, Dec. 15, 2020.

Acknowledgements

We would like to express our gratitude to our project advisor, Professor Ahmed Banafa, who guided us throughout this project and to our Project Class Instructor Professor Wencen Wu for providing us the necessary material for the completion of this project. We would also like to thank San Jose State University for providing us with facilities to complete our higher education.