Software Requirements Specification
Version 1.0

FlameGuard

Theme: Green Environment

Project Name: FlameGuard

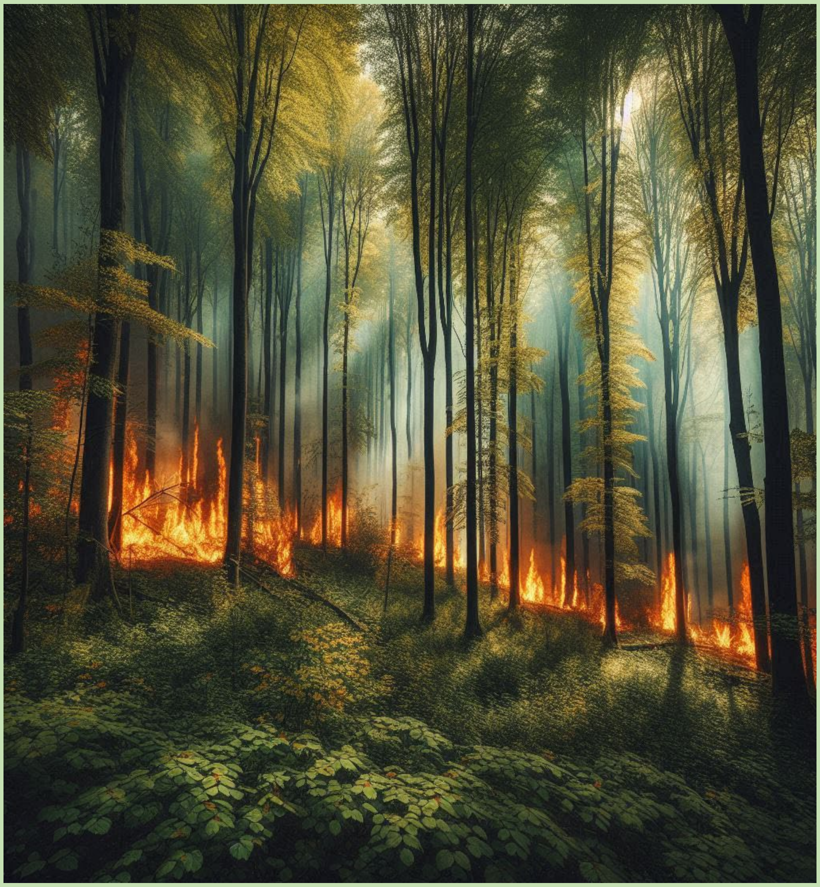
Category: Next-Gen Intelligent Solutions with AIML

Contents

1.1 Background and Necessity for the Application

Forest fires are a significant environmental concern, causing devastating damage to ecosystems, wildlife, and human communities. The frequency and intensity of these fires have been increasing due to climate change, prolonged droughts, and human activities. There is an urgent requirement for advanced technological solutions that can predict and monitor forest fires to enable faster and more efficient intervention.

FlameGuard, leveraging cutting-edge predictive analytics and Machine Learning algorithms, addresses this critical necessity. By analyzing vast amounts of data from various sources such as weather patterns, vegetation indices, and historical fire occurrences, **FlameGuard** can forecast potential fire outbreaks with high accuracy. This approach not only helps in safeguarding natural habitats and biodiversity, but also minimizes economic losses and protects human lives. The necessity for such an advanced application is underscored by the growing impact of climate change and the increasing occurrence of extreme weather events, making forest fire prediction and early warning systems more crucial than ever.



1.2 Proposed Solution

The proposed solution, 'FlameGuard', for the Forest Fire Prediction project involves creating a robust Machine Learning model to predict the likelihood of forest fires based on various environmental factors. The primary objective is to develop an accurate and efficient prediction application that can be easily accessed and used by users to mitigate the risks associated with forest fires.

Dataset and Features
The dataset for this project is provided to you. Alternatively, you can download the dataset from following link:

https://www.kaggle.com/datasets/elikplim/forest-fires-data-set.

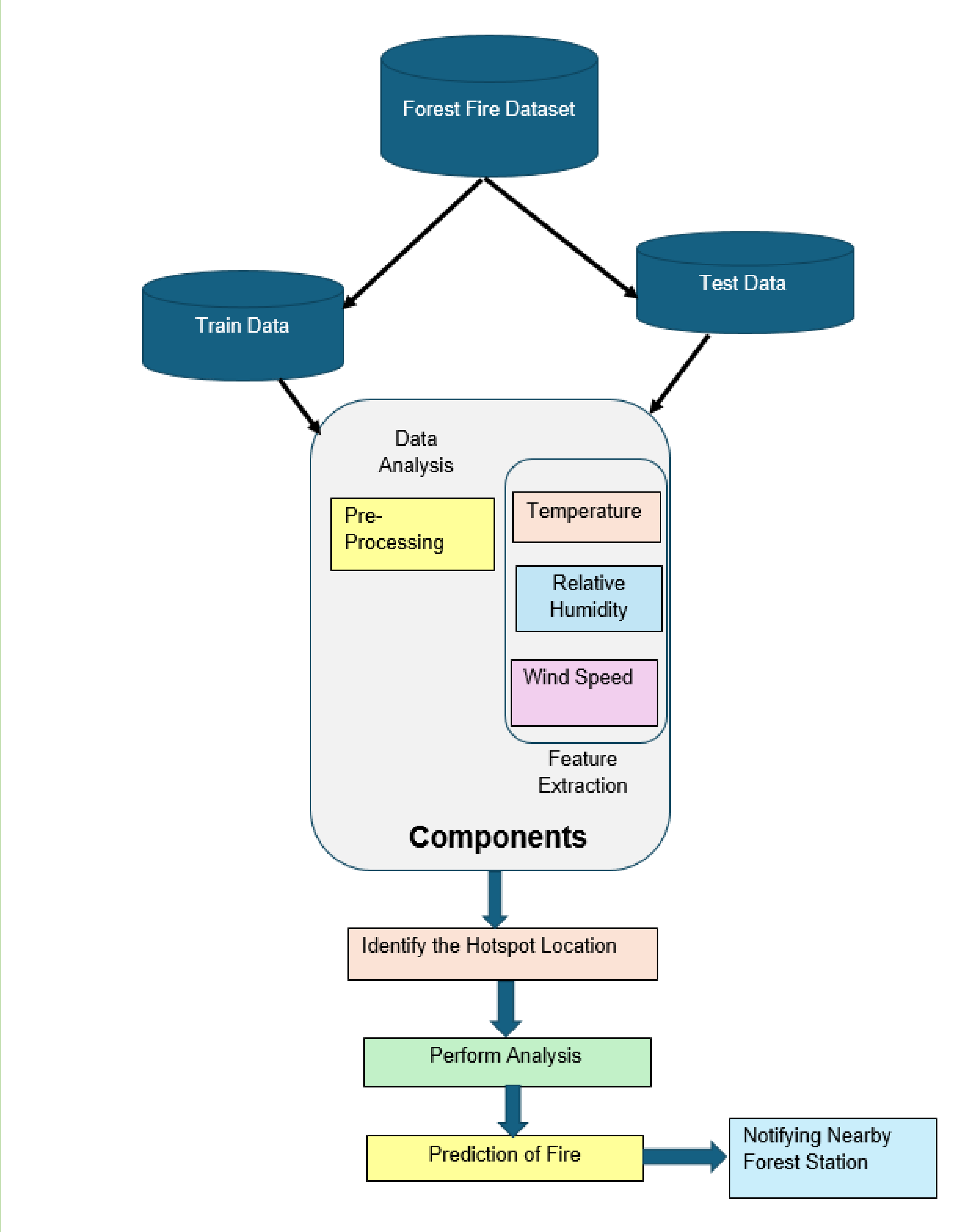
The dataset comprises a data matrix where columns represent variables and rows represent instances, specifically focusing on the northeast region of Portugal. The dataset includes 515 instances divided into training (333 instances), generalization (45 instances), and testing (45 instances) subsets. Key variables in the dataset are as follows:

- **Month:** The month of the year when the data was recorded.
- **Fine Fuel Moisture Code (FFMC):** Indicates the moisture content of surface litter and fine fuels, influencing how easily fires can start and spread.
- **Duff Moisture Code (DMC):** Represents the moisture content of loosely compacted organic layers, affecting how well a fire can sustain itself.
- **Drought Code (DC):** Measures the moisture content of deep, compact organic layers, indicating the likelihood of sustained burning.
- **Initial Spread Index (ISI):** Combines wind speed and the FFMC to estimate the initial spread rate of a fire.
- **Temperature:** The temperature in degrees Celsius.
- **Relative Humidity (RH):** The relative humidity in percentage.
- **Wind Speed:** The wind speed in km/h.
- **Class:** The target variable indicating the presence (1) or absence (0) of fire.

Hint: A sample of the dataset for Forest Fire downloaded from Kaggle for implementation purpose is as follows:

X	Y	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
7	5	mar	fri	86.2	26.2	94.3	5.1	8.2	51	6.7	0	0
7	4	oct	tue	90.6	35.4	669.1	6.7	18	33	0.9	0	0
7	4	oct	sat	90.6	43.7	686.9	6.7	14.6	33	1.3	0	0
8	6	mar	fri	91.7	33.3	77.5	9	8.3	97	4	0.2	0
8	6	mar	sun	89.3	51.3	102.2	9.6	11.4	99	1.8	0	0
8	6	aug	sun	92.3	85.3	488	14.7	22.2	29	5.4	0	0
8	6	aug	mon	92.3	88.9	495.6	8.5	24.1	27	3.1	0	0
8	6	aug	mon	91.5	145.4	608.2	10.7	8	86	2.2	0	0
8	6	sep	tue	91	129.5	692.6	7	13.1	63	5.4	0	0
7	5	sep	sat	92.5	88	698.6	7.1	22.8	40	4	0	0
7	5	sep	sat	92.5	88	698.6	7.1	17.8	51	7.2	0	0
7	5	sep	sat	92.8	73.2	713	22.6	19.3	38	4	0	0
6	5	aug	fri	63.5	70.8	665.3	0.8	17	72	6.7	0	0
6	5	sep	mon	90.9	126.5	686.5	7	21.3	42	2.2	0	0
6	5	sep	wed	92.9	133.3	699.6	9.2	26.4	21	4.5	0	0
6	5	sep	fri	93.3	141.2	713.9	13.9	22.9	44	5.4	0	0
5	5	mar	sat	91.7	35.8	80.8	7.8	15.1	27	5.4	0	0
8	5	oct	mon	84.9	32.8	664.2	3	16.7	47	4.9	0	0
6	4	mar	wed	89.2	27.9	70.8	6.3	15.9	35	4	0	0
6	4	apr	sat	86.3	27.4	97.1	5.1	9.3	44	4.5	0	0
6	4	sep	tue	91	129.5	692.6	7	18.3	40	2.7	0	0
5	4	sep	mon	91.8	78.5	724.3	9.2	19.1	38	2.7	0	0
7	4	jun	sun	94.3	96.3	200	56.1	21	44	4.5	0	0
7	4	aug	sat	90.2	110.9	537.4	6.2	19.5	43	5.8	0	0
7	4	aug	sat	93.5	139.4	594.2	20.3	23.7	32	5.8	0	0
7	4	aug	sun	91.4	142.4	601.4	10.6	16.3	60	5.4	0	0

The sample architecture of application can be as follows:



Sample Architecture of the Application

FlameGuard project involves developing a Machine Learning model to predict forest fires based on various environmental factors. The project will utilize the provided dataset containing variables such as temperature, humidity, wind speed, and moisture indices. The solution includes several key steps and serializes the trained model using Pickle to save it for future use. This allows the model to be loaded and used for predictions without retraining. Users can input temperature, humidity, and other relevant values to predict the likelihood of a forest fire, providing a user-friendly interface to assess fire risk. It can also notify the nearby Forest Station to assess fire risk and facilitate a timely response if the forest is in danger.

Steps to build the model for forest fire prediction are as follows:

1. Data Collection
2. Data Pre-Processing
3. Exploratory Data Analysis (EDA)
4. Feature Engineering
5. Feature Selection
6. Model Building
7. Model Selection
8. Hyperparameter Tuning
9. Model Serialization (Pickle the model)

1.3 Purpose of the Document

The purpose of this document is to outline the complete development plan for the Forest Fire Prediction system using Artificial Intelligence (AI) and Machine Learning (ML) techniques. This document ensures all stakeholders have a common understanding, promotes effective communication, and equips the team to deliver a high-quality Forest Fire Prediction application. This document is intended for both stakeholders and developers of the application.

1.4 Scope of Project

The scope of the project includes developing a complete Machine Learning pipeline for forest fire prediction. Preprocess the data, perform EDA, and perform feature engineering to build and select the most suitable model. The project will also involve hyperparameter tuning to optimize the model's performance. Once the model is built, a Web application using Flask/Django will be created to allow users to input environmental variables and receive predictions about forest fire risks. Furthermore, it should be noted that notifying nearby forest station to assess the fire risk falls outside the scope of this project.

1.5 Constraints

The project may face several constraints such as the computational limitations for training the model. Additionally, the variability in regional environmental conditions, such as different vegetation types and climatic factors, can challenge the application's accuracy and reliability impacting its predictive capabilities across diverse geographic areas.

1.6 Functional Requirements

This comprehensive project aims to provide an effective tool for predicting forest fires by leveraging advanced Machine Learning techniques. The Functional requirements are as follows:

i. **Data Collection** - The dataset should be collected from the specified source and prepared for analysis.

ii. **Data Preprocessing** - The data should undergo cleaning and preprocessing to handle missing values, outliers, and normalization to ensure it is ready for model building.

iii. **EDA** - Various data visualization techniques and statistical analyzes should be performed to understand the distribution, relationships, and patterns within the data.

iv. **Feature Engineering** - New features may be created from existing ones to improve model performance.

v. **Feature Selection** - Key features that significantly impact the target variable should be selected for model building.

vi. **Model Building** - Multiple machine learning algorithms should be applied to build models including logistic regression, decision trees, random forests, and support vector machines.

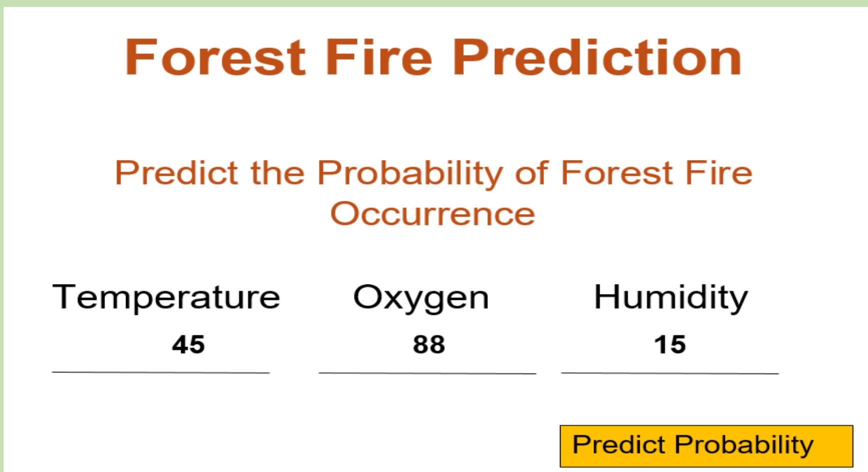
vii. **Model Evaluation and Selection** - Models should be evaluated using metrics such as accuracy, precision, recall, and F1-score. The best-performing model should be selected.

viii. **Hyperparameter Tuning** - The selected model should undergo hyperparameter tuning to optimize its performance.

ix. **Pickle the trained model** - Serialize the trained model to preserve it for making predictions.

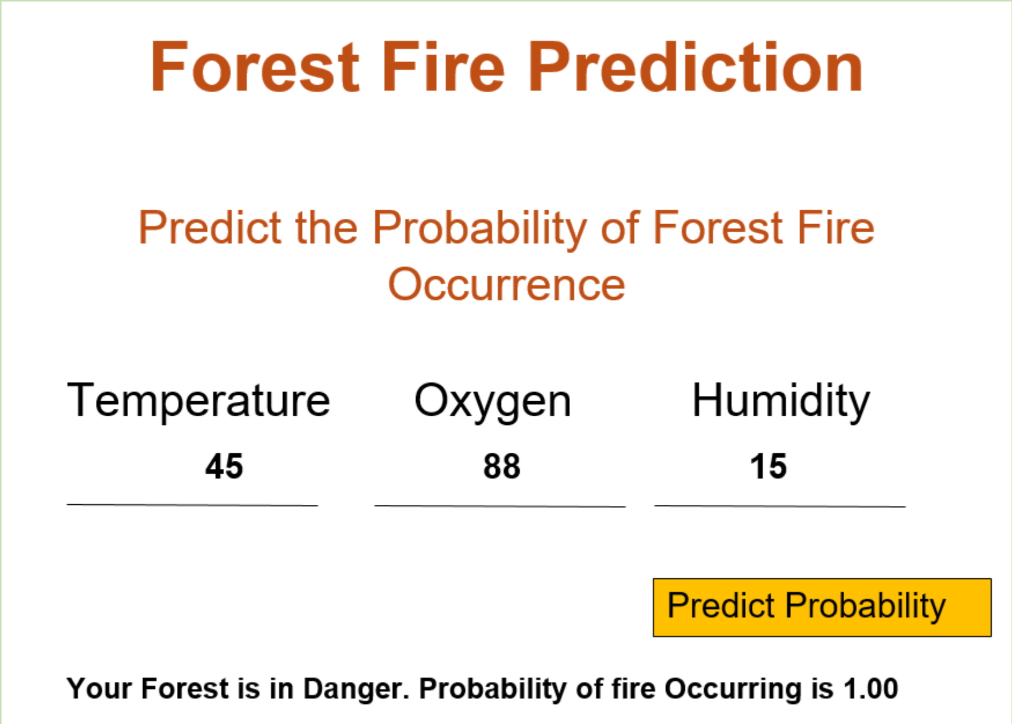
x. **Machine learning predicts fire risk** - Use the serialized model to predict the likelihood of a fire occurring based on new input data.

Here is a sample Home page of an application for your reference:



Users should input Temperature, Oxygen, and Humidity values to predict the probability of a forest fire based on calculations. The application will indicate whether the forest is at risk or safe.

Here is the output sample of this application:



1.7 Non-Functional Requirements

There are several non-functional requirements that should be fulfilled by the application.

The application should be:

1. **Accurate**: The application should achieve a high level of accuracy in predicting forest fires, ideally exceeding 90%.

2. **Secure**: The application must ensure data security and protect against unauthorized access.

3. **Robust**: The application must be robust, handling a wide range of input values and providing reliable predictions under various conditions.

4. **<u>Scalable</u>**: The application should be capable of handling increasing amounts of data and user requests without performance degradation.

5. **<u>User-Friendly</u>**: The Web application must be intuitive and easy to use, with a clear interface for inputting data and receiving predictions.

These are the bare minimum expectations from the project. It is a must to implement the FUNCTIONAL and NON-FUNCTIONAL requirements given in this SRS.

Once they are complete, you can use your own creativity and imagination to add more features if required.

1.8 Interface Requirements

1.8.1 Hardware

Intel Core i5/i7 Processor or higher
8 GB RAM or higher
Color SVGA
500 GB Hard Disk space
Mouse
Keyboard

1.8.2 Software

Technologies to be used:

1. **Frontend**: HTML5 or any other scripting languages
2. **Backend**: Flask/Django
3. **Data Store**: CSV
4. **Programming/IDE**: Python, Jupyter Notebook, Anaconda, or Google Colab
5. **Libraries**: Sklearn, NumPy, Pandas, Matplotlib, Keras, and Tensorflow

1.9 Project Deliverables

You will design and build the project and submit it along with a complete project report that includes:

- Problem Definition
- Design Specifications
- Diagrams such as Dialog Flow
- Test Data Used in the Project
- Project Installation Instructions
- Proper Steps to Execute the Project
- Link of GitHub for accessing the uploaded project code (Link should have public access)
- Link of Published Blog

The source code, including .ipynb files for Jupyter Notebook and Google Colab, should be shared via GitHub. Appropriate access permissions should be granted to users to allow testing for Jupyter Notebook and Google Colab. The consolidated project must be submitted on GitHub with a ReadMe.doc file listing assumptions (if any) made at your end.

Provide the GitHub URL where the project has been uploaded for sharing. The repository on GitHub should have public access. Documentation is a very important part of the project; hence, all crucial aspects of the project must be documented properly. Ensure that documentation is complete and comprehensive.

You should publish a blog of minimum 2000 words on any free blogging Website such as Blogger, Tumblr, Ghost or any other blogging Website. The link of the published blog should be submitted along with the project documentation.

Submit a video (.mp4 file) demonstrating the working of the application, including all the functionalities of the project. This is MANDATORY.

~~~ End of Document ~~~