MACHINE LEARNING

FOREST FIRE RISK PREDICTION

Submitted by

UDHAYAPRABHA S (95072215054)

in partial fulfillment for the award of the

degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

FRANCIS XAVIER ENGINEERING COLLEGE

(Autonomous)

TIRUNELVELI – 627 003

NOV 2025

FRANCIS XAVIER ENGINEERING COLLEGE
MACHINE LEARNING

BONAFIDE CERTIFICATE

Certified that this project report " **Forest Fire Risk Prediction** " is the bonafide work of " **Udhayaprabha S (95072215054)** " who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

HEAD OF THE DEPARTMENT **SUPERVISOR**

TABLE OF CONTENTS

INTRODUCTION

The Forest Fire Risk Prediction System is a data-driven, intelligent solution designed to predict and assess the likelihood of forest fire incidents based on a combination of environmental and meteorological parameters. Forest fires are among the most devastating natural disasters, resulting in the destruction of ecosystems, loss of biodiversity, and severe impacts on air quality and climate. Traditional fire monitoring systems are often reactive, detecting fires only after they have started. In contrast, this project aims to build a proactive system that can identify potential fire risks in advance, enabling timely preventive measures.

The model utilizes a Random Forest Classifier, a robust machine learning algorithm known for its accuracy and resistance to overfitting, to analyze parameters such as temperature, humidity, rainfall, wind speed, and fire weather indices (FFMC, DMC, DC, ISI). These variables are key indicators of forest dryness and flammability, making them critical in predicting fire risk levels. The dataset used for training and testing is preprocessed to remove noise and ensure reliable prediction performance.

The project integrates Flask, a lightweight Python web framework, to serve as the backend for data processing and model inference. A simple yet user-friendly HTML/CSS-based web interface allows users to enter environmental data and obtain instant predictions. The system outputs the predicted fire risk category (High or Low) along with a probability percentage and generates a bar graph for intuitive visualization of fire risk levels. Additionally, the system features an automated PDF report generator that compiles the user's input data, model prediction, and probability visualization into a downloadable document. The integration of technology into environmental monitoring contributes significantly to achieving sustainable forest management and preventing large-scale ecological damage.

ABSTRACT

The Forest Fire Risk Prediction System is a machine learning–based web application designed to estimate the likelihood of forest fire occurrences based on environmental and meteorological inputs. Forest fires pose severe threats to natural resources, biodiversity, and human safety, making early detection and risk assessment essential components of environmental management. This project applies the Random Forest Classifier algorithm to classify the risk level of forest fires as High or Low by analyzing key features such as temperature, humidity, rainfall, wind speed, and fire weather indices (FFMC, DMC, DC, ISI).

The system is implemented using Python and Flask as the backend framework, offering a lightweight and scalable architecture. The user interface is developed with HTML, CSS, and JavaScript, allowing users to input data through a clean and responsive web form. Once the data is submitted, the backend model processes the inputs, generates predictions, and returns results that include a textual fire risk level, probability percentage, and a corresponding graphical representation using Matplotlib.

In addition to visualization, the system provides a PDF report generation feature that automatically compiles the user's inputs, the model's prediction, and a probability graph into a downloadable document for record-keeping and analysis. This functionality makes the tool practical for use by forest officers, researchers, and government agencies engaged in environmental monitoring.

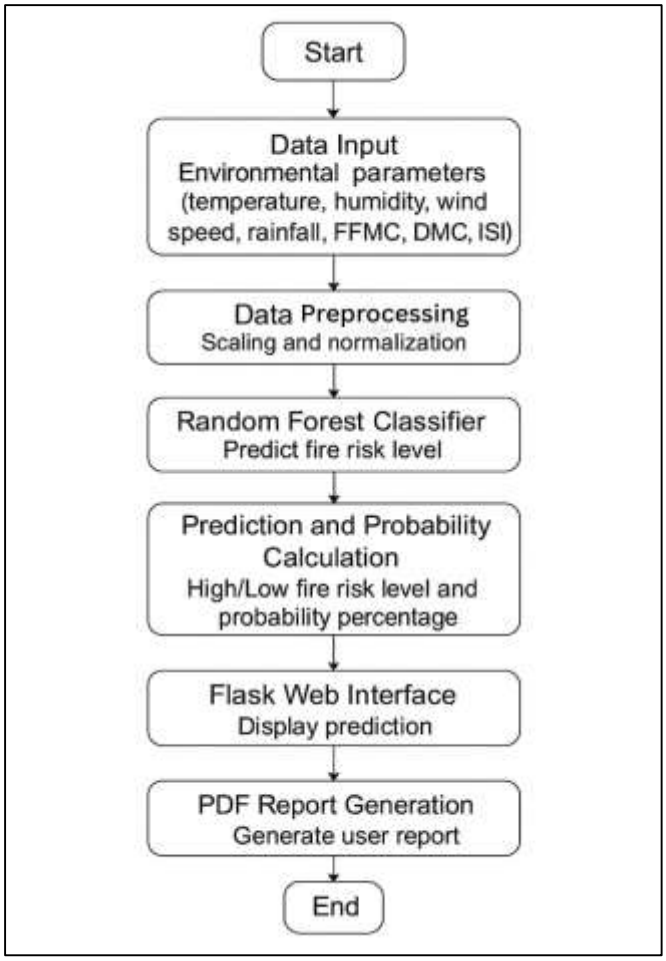
Overall, this project demonstrates the effective application of machine learning and web technologies in environmental risk prediction. By providing accurate, interpretable, and easily accessible predictions, the system serves as a valuable tool for early warning, decision-making, and sustainable forest management.

OBJECTIVE

The main objectives of the Forest Fire Risk Prediction System are as follows:

- To develop an intelligent prediction model using Machine Learning (Random Forest Classifier) to accurately assess forest fire risk levels based on environmental and meteorological parameters.
- To design a user-friendly web application using Flask that allows users to easily input data and instantly receive fire risk predictions.
- To enhance early warning and preventive measures by providing real-time fire risk classification (High or Low) along with probability scores.
- To visualize prediction outcomes through graphs and charts for better understanding of the fire risk probability.
- To generate automated PDF reports containing user inputs, model predictions, and probability charts for easy documentation and record-keeping.
- To integrate data preprocessing and model inference seamlessly in the backend to ensure accurate and efficient performance.
- To contribute to environmental protection and sustainable forest management by supporting timely decision-making and risk mitigation efforts.

FLOWCHART



SAMPLE CODE

```
from flask import Flask, render_template, request, send_file
import pickle, numpy as np, pandas as pd, matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
from reportlab.lib.pagesizes import letter
from reportlab.pdfgen import canvas
from reportlab.lib import colors
from reportlab.lib.utils import ImageReader
import io, os
from textwrap import wrap
app = Flask(__name__)
```

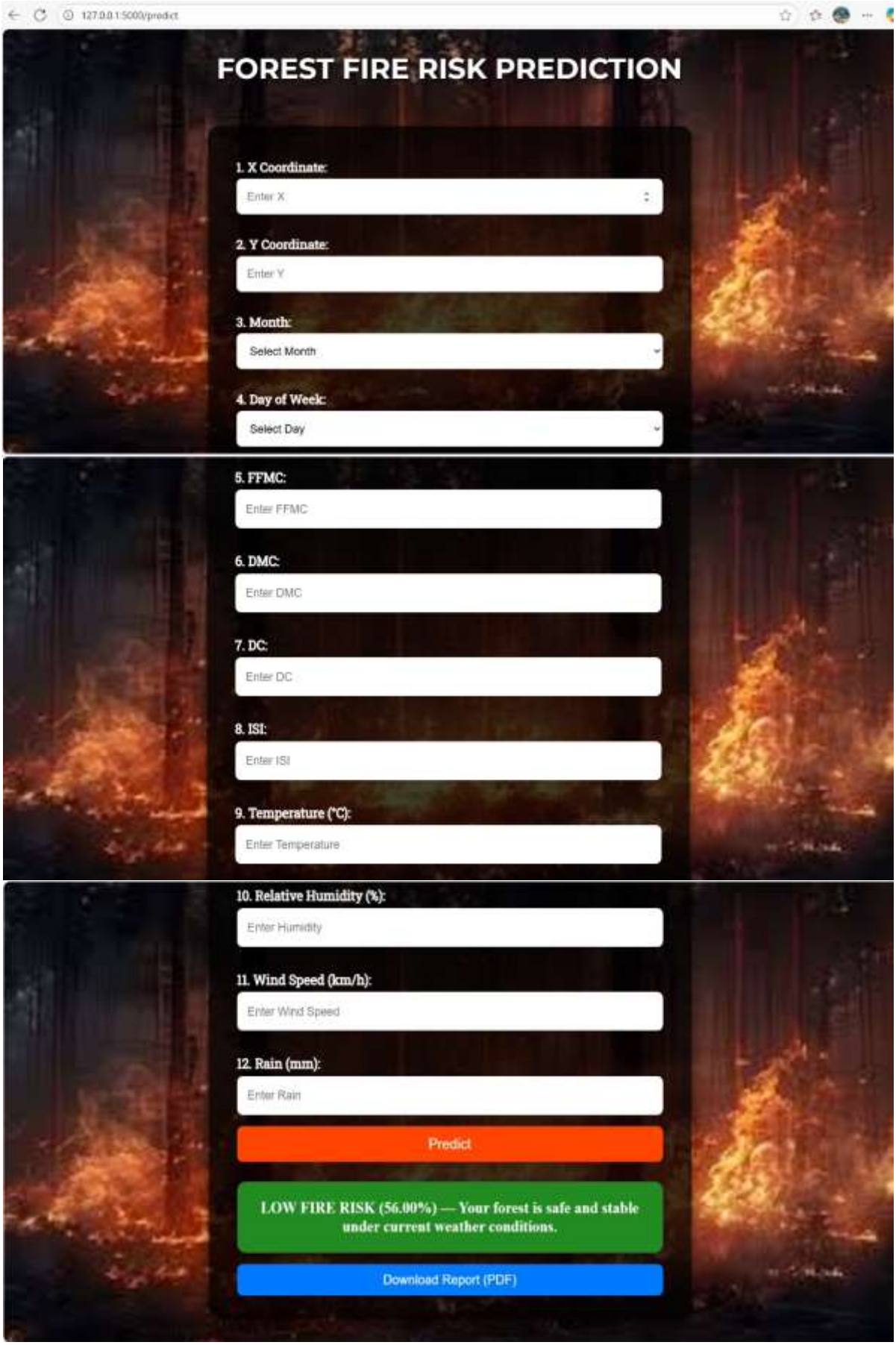
//Load Model & Scaler

```
model = pickle.load(open(r'C:\Users\user\Desktop\forestfire_project\forest_fire_model (2).pkl', 'rb'))
scaler = pickle.load(open(r'C:\Users\user\Desktop\forestfire_project\scaler (1).pkl', 'rb'))
last_result = last_input = last_probability = None
@app.route('/')
def home():
    return render_template('index.html')
@app.route('/predict', methods=['POST'])
def predict():
    global last_result, last_input, last_probability
    try:
        features = ['X','Y','month','day','FFMC','DMC','DC','ISI','temp','RH','wind','rain']
        input_values = [float(request.form[f]) for f in features]
        input_scaled = scaler.transform([input_values])
        pred = model.predict(input_scaled)[0]
        prob = model.predict_proba(input_scaled)[0][int(pred)] * 100
```

```
if pred == 1:
            result = f"HIGH FIRE RISK ({prob:.2f}%) — Take preventive measures!"
            color = "high-risk"
        else:
            result = f"LOW FIRE RISK ({prob:.2f}%) — Safe conditions."
            color = "low-risk"
last_result, last_input, last_probability = result, dict(zip(features, input_values)), prob
        return render_template('index.html', prediction_text=result, color_class=color)
    except Exception as e:
        return render_template('index.html', prediction_text=f"Error: {e}")
@app.route('/download_pdf')
def download_pdf():
    global last_result, last_input, last_probability
    if not last_result: return "No prediction yet."
    buf = io.BytesIO()
    pdf = canvas.Canvas(buf, pagesize=letter)
    w, h = letter
    pdf.setFont("Times-Bold", 18)
    pdf.drawCentredString(w/2, h-80, "Forest Fire Risk Prediction Report")
//Input Table
    y = h-130
    pdf.setFont("Times-Roman", 12)
    for k,v in last_input.items():
        pdf.drawString(60, y, f"{k}: {v}")
        y -= 18
```

```
//Result
    y -= 20
    pdf.setFont("Times-Bold", 14)
    pdf.drawString(60, y, "Prediction Result:")
    y -= 18
    pdf.setFont("Times-Roman", 12)
    for line in wrap(last_result, 70):
        pdf.drawString(80, y, line)
        y -= 14
//Graph
    fig, ax = plt.subplots(figsize=(3,2))
    ax.bar(['Fire Risk'], [last_probability], color='red' if "HIGH" in last_result else 'green')
    ax.set_ylim(0, 100)
    plt.tight_layout()
    img = io.BytesIO()
    plt.savefig(img, format='png')
    plt.close(fig)
    img.seek(0)
    pdf.drawImage(ImageReader(img), 80, y-150, width=250, height=140)
   pdf.save()
    buf.seek(0)
    return send_file(buf, as_attachment=True, download_name="Forest_Fire_Risk_Report.pdf")
if __name__ == '__main__':
    app.run(debug=True)
```

RESULT


127.0.0.1:5000/predict
FOREST FIRE RISK PREDICTION
1. X Coordinate:
Enter X
2. Y Coordinate:
Enter Y
3. Month:
Select Month
4. Day of Week:
Select Day
5. FFMC:
Enter FFMC
6. DMC:
Enter DMC
7. DC:
Enter DC
8. ISI:
Enter ISI
9. Temperature (°C):
Enter Temperature
10. Relative Humidity (%):
Enter Humidity
11. Wind Speed (km/h):
Enter Wind Speed
12. Rain (mm):
Enter Rain
Predict
LOW FIRE RISK (56.00%) — Your forest is safe and stable under current weather conditions.
Download Report (PDF)

FOREST FIRE RISK PREDICTION REPORT

User Input Details:

Parameter	Value
X	5.0
Y	7.0
month	6.0
day	4.0
FFMC	45.0
DMC	34.0
DC	22.0
ISI	34.0
temp	45.0
RH	23.0
wind	27.0
rain	26.0

Prediction Result:

LOW FIRE RISK (56.00%) — Your forest is safe and stable under current weather conditions.

Fire Risk Probability: **56.00%**

Report generated by Forest Fire Risk Prediction System

CONCLUSION

The Forest Fire Risk Prediction System successfully demonstrates the application of machine learning in environmental safety and disaster management. By analyzing key meteorological and environmental parameters such as temperature, humidity, wind speed, and drought indices, the system effectively classifies areas into high or low fire risk zones. The integration of a user-friendly Flask web interface allows real-time prediction and automated PDF report generation, making the solution both practical and accessible. The model's predictions can assist forest authorities, researchers, and local administrations in implementing preventive measures, optimizing resource allocation, and minimizing fire-related losses. Overall, this project highlights the potential of data-driven decision-making in promoting sustainable forest management and early fire risk detection.

FUTURE SCOPE

- **Integration with IoT Sensors:** Incorporate real-time data from temperature, humidity, soil moisture, and wind sensors for continuous monitoring.
- **Use of Satellite & Drone Imagery:** Utilize remote sensing data to detect fires early and analyze vegetation dryness or fuel levels.
- **GIS-Based Visualization:** Display high-risk fire zones on dynamic maps for better decision-making and preventive actions.
- **Advanced AI & Deep Learning Models:** Employ CNNs, LSTMs, or hybrid ensemble models for more accurate and intelligent fire risk prediction.
- **Real-Time Alert System:** Develop mobile and web apps to send instant notifications to forest authorities and nearby residents.
- **Cloud Deployment:** Host the model on cloud platforms for large-scale data processing and 24/7 system availability.
- **Predictive & Preventive Insights:** Recommend fire control measures, resource allocation, and evacuation planning based on predicted risks.
- **Integration with Weather Forecasting APIs:** Combine predictive models with live weather data to improve forecast precision.
- **Community Awareness Programs:** Use prediction outputs to educate local communities on preventive actions and early responses.
- **Global Scalability:** Expand the model for use across different geographic regions with localized environmental data.

REFERENCE

- Breiman, L. (2001). *Random Forests.* Machine Learning, 45(1), 5–32.
- Jain, A., & Singh, P. (2022). *Forest Fire Prediction Using Machine Learning Algorithms.* International Journal of Computer Applications, 184(26), 1–6.
- Giglio, L., Schroeder, W., & Justice, C. O. (2016). *The Collection 6 MODIS Active Fire Detection Algorithm and Fire Products.* Remote Sensing of Environment, 178, 31–41.
- Elikplim, N. (2022). *Forest Fires Data Set.* Kaggle. Retrieved from https://www.kaggle.com/datasets/elikplim/forest-fires-data-set
- Flask Documentation. (2024). *Flask Web Framework.* Retrieved from https://flask.palletsprojects.com/
- Scikit-learn Developers. (2024). *Scikit-learn: Machine Learning in Python.* Retrieved from https://scikit-learn.org/