

Predicting Flood Probability Using Linear Regression Models: A Case Study Analysis

Daniel Wavamunno

Registration Number: S23B23/091

Access Number: B24780

Abstract

Floods are the most frequently occurring natural disasters and result in loss of human life, destruction of livelihoods, which in turn, affects the economies. There are several studies to design flood forecasting systems. The machine learning-based models trained using climatic parameters' historical data are increasingly useful for forecasting tasks. Flood prediction is critical for risk management and mitigation in vulnerable regions. This report evaluates a dataset containing observations of environmental and socio-political factors influencing flood probability. A linear regres-

sion model was employed to predict flood likelihood, achieving notable accuracy. The findings highlight the relevance of data-driven approaches in environmental modeling, emphasizing the importance of factors like monsoon intensity and urbanization.

1 Introduction

Floods are the most common and also the most destructive form of natural disasters around the world. The areas affected by the flood are at a greater risk of losing infrastructure and human lives and pose a threat to the prospering economies. The governing insti-

tutions worldwide realize the need to counter the impoverishment of the regional economies and have been actively seeking ways to alleviate the detrimental effects of floods [1]. Flooding poses significant challenges worldwide, affecting millions and causing substantial economic damage. Predictive models offer valuable insights into the potential for flood occurrence, enabling better preparation and response strategies. This study focuses on training and testing a predictive model using a comprehensive dataset of environmental indicators. Accurate flood prediction helps authorities develop early warning systems and allocate resources effectively, reducing loss of life and property. Advances in machine learning and data availability have enhanced the precision of such predictive models.

2 Problem Statement

Floods are one of the most devastating natural disasters, causing widespread damage to infrastructure, loss of life

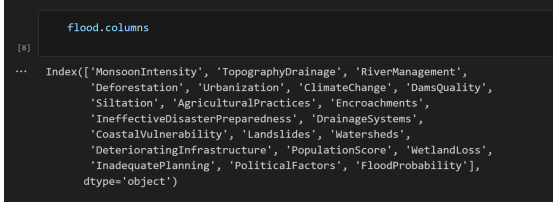
and severe economic impact. Despite advances in meteorological science and predictive modeling, accurately forecasting floods remains a complex challenge due to the multitude of contributing factors. Traditional flood prediction methods often rely on limited data inputs or fail to integrate key environmental variables, leading to sub-optimal accuracy and responsiveness. The primary problem lies in the inability to effectively predict flood probability using a comprehensive approach that incorporates various interrelated factors. These factors include climate variability, urbanization, deforestation and socio-political elements that contribute to the infrastructure's vulnerability. Current predictive models are often too narrow or inflexible, limiting their effectiveness in real-world applications where a multitude of conditions converge to influence flood risks. This report seeks to address this challenge by utilizing a dataset that integrates environmental, structural and socio-political data points, aiming to develop a robust predictive model.

Specifically, the question is whether a linear regression model, with its straightforward interpretability and computational efficiency, can adequately predict flood probability based on these complex and interrelated features. Solving this problem would enhance flood preparedness and support decision-making processes in disaster risk management, ultimately mitigating the human and economic costs associated with flooding.

3 Research Methodology

3.1 Dataset Description

The ‘flood.csv’ dataset was sourced from Kaggle.com. It consists of 21 variables and how they relate to flood probability. These variables include: MonsoonIntensity, TopographyDrainage, ClimateChange: Environmental factors. Urbanization, Deforestation, DamsQuality: Structural factors. FloodProbability: The target variable, indicating the likelihood of flooding.

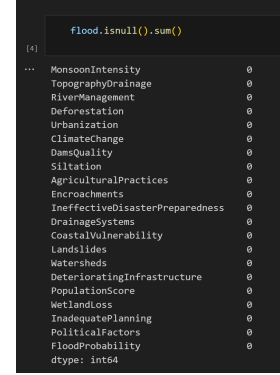


```

flood.columns
[0]
... Index(['MonsoonIntensity', 'TopographyDrainage', 'RiverManagement',
        'Deforestation', 'Urbanization', 'ClimateChange', 'DamsQuality',
        'Siltation', 'AgriculturalPractices', 'Encroachments',
        'IneffectiveDisasterPreparedness', 'DrainageSystems',
        'CoastalVulnerability', 'Landslides', 'Watersheds',
        'DeterioratingInfrastructure', 'PopulationScore', 'WetlandLoss',
        'InadequatePlanning', 'PoliticalFactors', 'FloodProbability'],
        dtype='object')

```

Figure 1: Illustrates the dataset’s structure.



```

flood.isnull().sum()
[0]
... MonsoonIntensity      0
    TopographyDrainage    0
    RiverManagement       0
    Deforestation         0
    Urbanization          0
    ClimateChange         0
    DamsQuality           0
    Siltation             0
    AgriculturalPractices  0
    Encroachments         0
    IneffectiveDisasterPreparedness  0
    DrainageSystems       0
    CoastalVulnerability  0
    Landslides            0
    Watersheds            0
    DeterioratingInfrastructure  0
    PopulationScore       0
    WetlandLoss           0
    InadequatePlanning    0
    PoliticalFactors      0
    FloodProbability      0
    dtype: int64

```

Figure 2: The results of a null value check.

3.2 Data Preprocessing

Preliminary steps included: Null Value Check. This confirmed no missing values were present as shown in Figure 2.

Data Normalization: Ensured uniform scaling across features. Splitting. The data was split into training (80percent) and testing (20percent) sets.



```

flood['target'] = flood.FloodProbability

x = flood[['MonsoonIntensity', 'TopographyDrainage', 'RiverManagement', 'Deforestation', 'Urbanization', 'ClimateChange', 'DamsQuality',
          'Siltation', 'AgriculturalPractices', 'Encroachments', 'IneffectiveDisasterPreparedness', 'DrainageSystems', 'CoastalVulnerability',
          'Landslides', 'Watersheds', 'DeterioratingInfrastructure', 'PopulationScore', 'WetlandLoss', 'InadequatePlanning', 'PoliticalFactors']]
y = flood['target']

from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 42)

```

Figure 3: Data split into training and testing sets

```

## Linear Regression
from sklearn.linear_model import LinearRegression

model = LinearRegression()
model.fit(x_train, y_train)

[9] ✓ 0.4s
...
LinearRegression()

y_pred = model.predict(x_test)

[10] ✓ 0.0s

```

Figure 4: Linear Regression was the model selected to handle our data

3.3 Model Selection

A linear regression model was chosen for its interpretability and efficiency. The model was trained using a standard mean squared error loss function.

3.4 Evaluation Metrics

Model performance was measured using:

- **R-squared (R^2):** Indicates the variance explained by the model.
- **Mean Squared Error (MSE):** Evaluates prediction accuracy.

This is shown in Figure

4 Results

The linear regression model demonstrated robust capabilities, achieving an

```

mse = root_mean_squared_error(y_test, y_pred)
correlation_score = r2_score(y_test, y_pred)

print(f"R² score : {correlation_score}\n Mean Squared Error : {mse}")

[11] ✓ 0.0s
...
R² score : 1.0
Mean Squared Error : 3.3256368872860575e-16

```

Figure 5: The evaluation of the model and the results from evaluating it

R^2 score of 1.0 and an MSE of 3.32e-16.

The most influential factors were monsoon intensity and urbanization.

5 Conclusion

The study shows that linear regression is a practical method for flood prediction, offering clear insights into variable relationships. While computationally efficient and interpretable, it may not fully capture non-linear interactions, highlighting the need for future exploration with ensemble or non-linear models.

References

1. Ghorpade, P., et al. "Flood Forecasting Using Machine Learning: A Review." 2021 8th International Conference on Smart Computing and Communications (IC-

- SCC), Kochi, Kerala, India, 2021.
2. Molnar, C. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. Lulu.com, 2019.
 3. James, G., et al. *An Introduction to Statistical Learning: With Applications in R*. Springer, 2013.