Analysis and Enhancement of Flood Probability Prediction Models

Exploring the Limitations of Traditional Methods and the Role of Statistical Features

Rui qi Xing, Zi xiao Wang

Abstract

This report outlines efforts to construct a flood probability prediction model using statistical and machine learning methods. Initially, we explored the dataset's characteristics through visualization and statistical tests to understand its distribution and correlations. Following this, various machine learning models were trained and optimized. Through exploration and optimization, we find statistics features and slightly advanced model based on simple model improve the performance of the model. Then we reveals the reason why some of the classic method does not work in this problem. These findings can be inspiring towards the field of Data Mining.

Introduction

Floods are among the most devastating natural disasters, impacting millions of people worldwide. Accurate prediction models can mitigate the adverse effects of floods by enabling timely interventions. Our research aims to develop a robust flood probability prediction model that leverages data characteristics and advanced machine learning techniques. This report provides a detailed account of our methodology, experiments, and findings.

1 Data Exploration and Visualization

Understanding the dataset is fundamental to building a reliable prediction model. The dataset used in this study was sourced from Kaggle and comprises 1 million samples with 20 features, making it a relatively large and complex dataset. To evaluate and improve the model, we first present data exploration analysis and visualization on this data set

1.1 Box plot & Distribution exploration

To find the basic statistic information of the dataset, we began by visualizing the data using box plots:

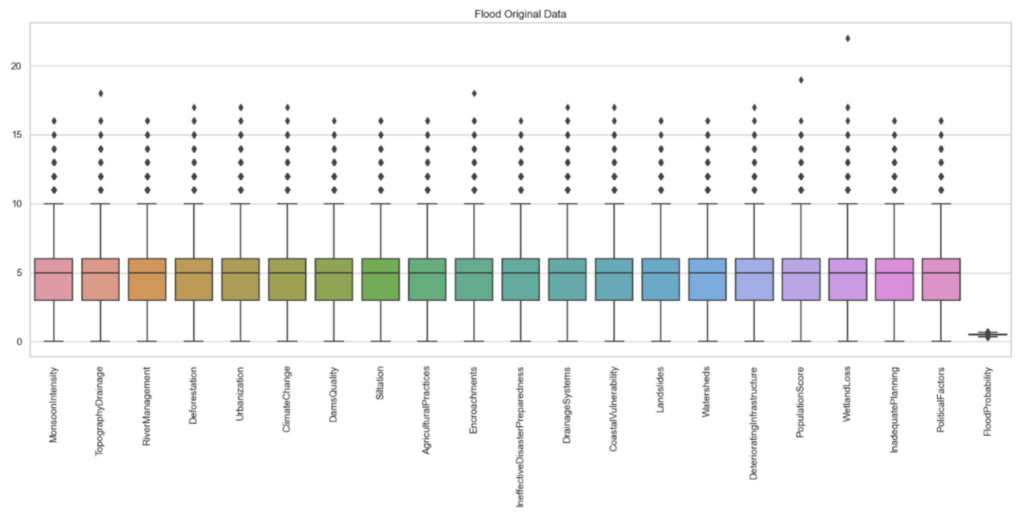
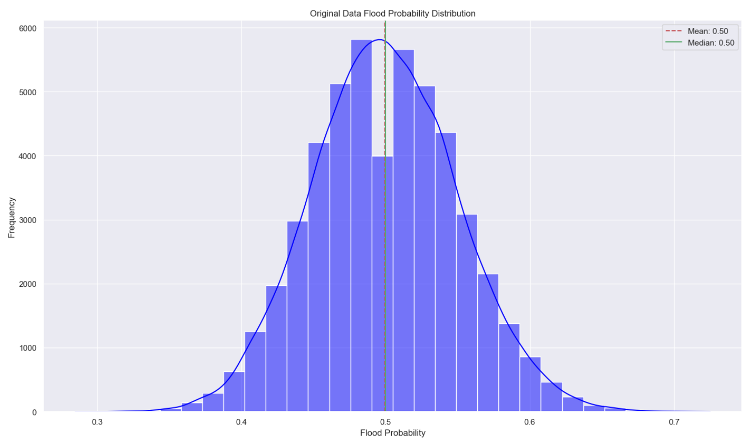


Figure 1: Box plot for the dataset

These plots revealed that most data points had an average value of approximately 5, with only a few outliers present. This observation suggested that the dataset was relatively uniform, with limited variability.

1.2 Distribution of the dataset

Next, we analyzed the data distribution through fitting curves, wondering whether these feature has similar distribution. The graph of feature distribution indicated a similarity to a Poisson distribution. Since there are multiple features, they are presented in the **”predict.ipynb” files**.Here we present the distribution of flood probability, more detailed figure will shown in the code files:



If it do follows a poission distribution, we can use the properties of poisson distribution to improve model performances. However, further statistical tests, including Kernel Density Estimation (KDE) plots and histograms, coupled with the Shapiro-Wilk test, demonstrated that the data did not strictly adhere to a Poisson distribution. The p-values obtained were significantly less than 0.05, rejecting the null hypothesis of a Poisson distribution.

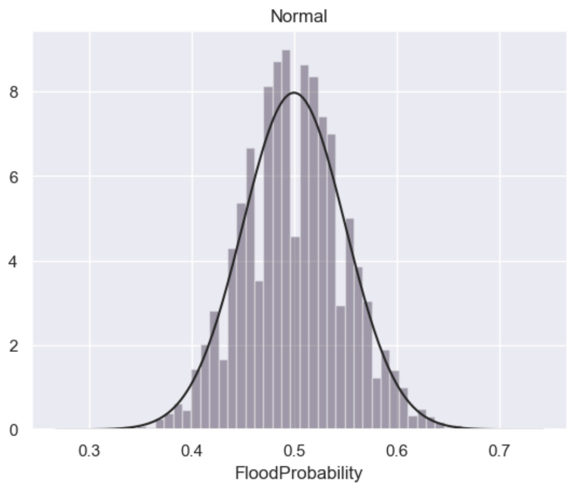


Figure 2: Distribution fit in normal distribution

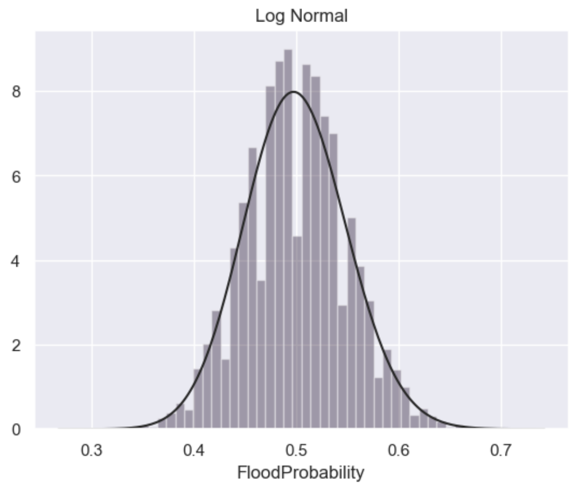


Figure 3: Distribution fit in logistic normal distribution

For the flood probability, we also add a simple test to decide which distribution mode can best fit the data. The distribution of flood probability can fit in many distribution including normal distribution, logistic normal distribution, poisson distribution and Johnson distribution. In comparsion, we find that the normal distribution slightly better than the other.

1.3 Pearson Correlation Coefficient

Since the distribution of the features are closed, we are curious about the relationship among these feature. To examine this, we generated a heat map of Pearson correlation coefficients. The Pearson correlation coefficient r between two variables X and Y is calculated as:

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2 \sum_{i=1}^{n}(Y_i - \bar{Y})^2}}, \tag{1}$$

where X_i and Y_i are the individual data points, $\bar{X}$ and $\bar{Y}$ are the means of X and Y, respectively, and n is the number of data points. After calculation, the result of the heatmap is shown below:

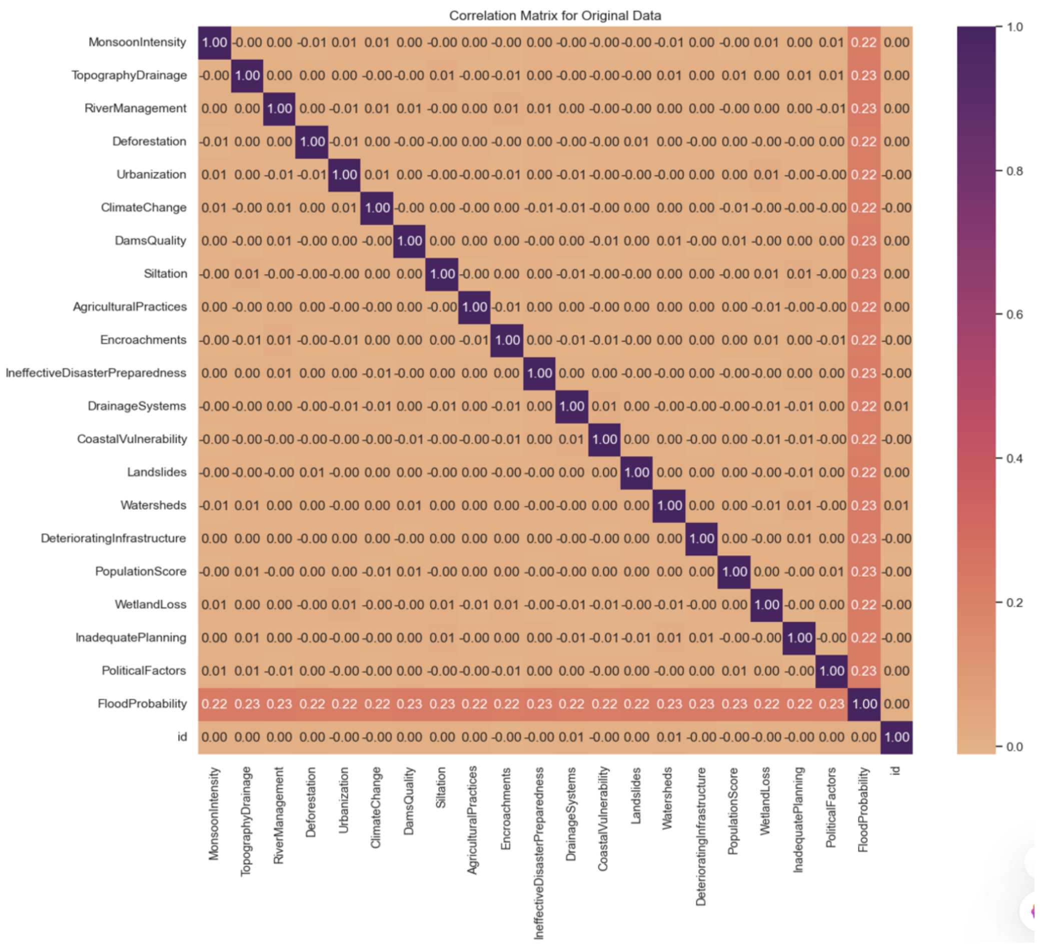


Figure 4: Pearson Heatmap

The correlation matrix revealed that most feature pairs had coefficients close to zero, indicating weak linear relationships. This lack of strong correlations posed a challenge for traditional linear modeling approaches.

It seems that the dataset has no particular mode, close to a standard normal distribution dataset. Yet we still doubt that if the sample in the dataset forms different shape. So we apply dimension reduction method into the dataset. Using the Principal Component Analysis(PCA) in both 2 dimension and 3 dimension,we gain the graph below. Sadly, it seems that there is no specific shape in the dataset. t-distributed stochastic neighbor embedding are used as well. Yet the effect remain the same.

After all this exploration and analysis, we found this dataset was different than the dataset we met before. So here the projectcame to several question: **How will model perform on this dataset? How to improve model on this dataset? Do traditional methods fit this dataset?** This leads to the next part of our work.

2 Model Training & Model Selection

2.1 Base Model

2.1.1 Training and evaluation

At first, four base models are trained: Linear Regression, Back Propagation(BP) Neural Network, Random Forest and Extreme Gradient Boosting Tree. Through training, we found that Linear Regression is the most accurate model. Based on this, we then apply Multilayer Perceptron(MLP) to this dataset as originally MLP was based on the linear combination of the neuron in layers.We set hidden layer to 3 and the neurons in layers are 64,32,16 respectively.As we expected, the model performs even better than linear Regression, with a R-square about 0.866. As far as this presentation are presented, MLP is still the best performed model on this dataset. The detailed information of these five models are listed in the table below:

Table 1: Detailed Performance Comparison of Models

Model	**Training Time**	**Train R2**	**Val RMSE**	**Val R2**	**Cross Val Score**
Linear Regression	0.521613	0.844968	0.020080	0.844877	0.844959
BP Neural Network	5448.013103	0.793134	0.023190	0.793099	*NaN*
Random Forest	935.080230	0.951190	0.029920	0.655605	0.652653
XGBoost	1.915260	0.818250	0.022278	0.809067	0.809204
MLP	295.451996	0.861312	0.018997	0.861165	0.859119

A viuslization of these models are presented.

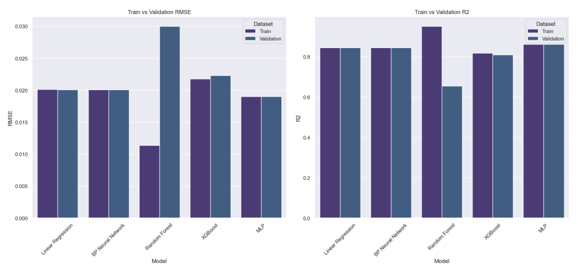


Figure 5: RMSE and R^2 on training and validation dataset

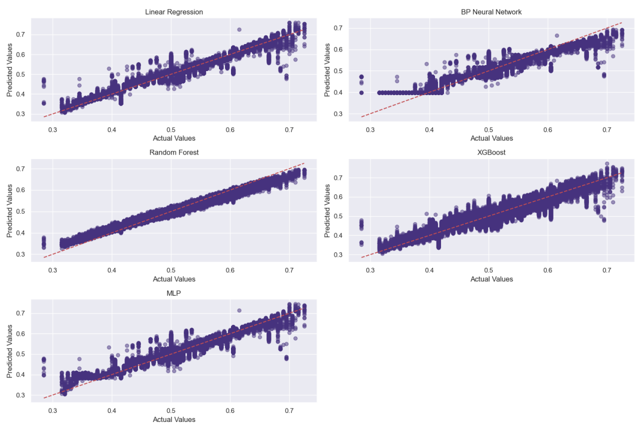
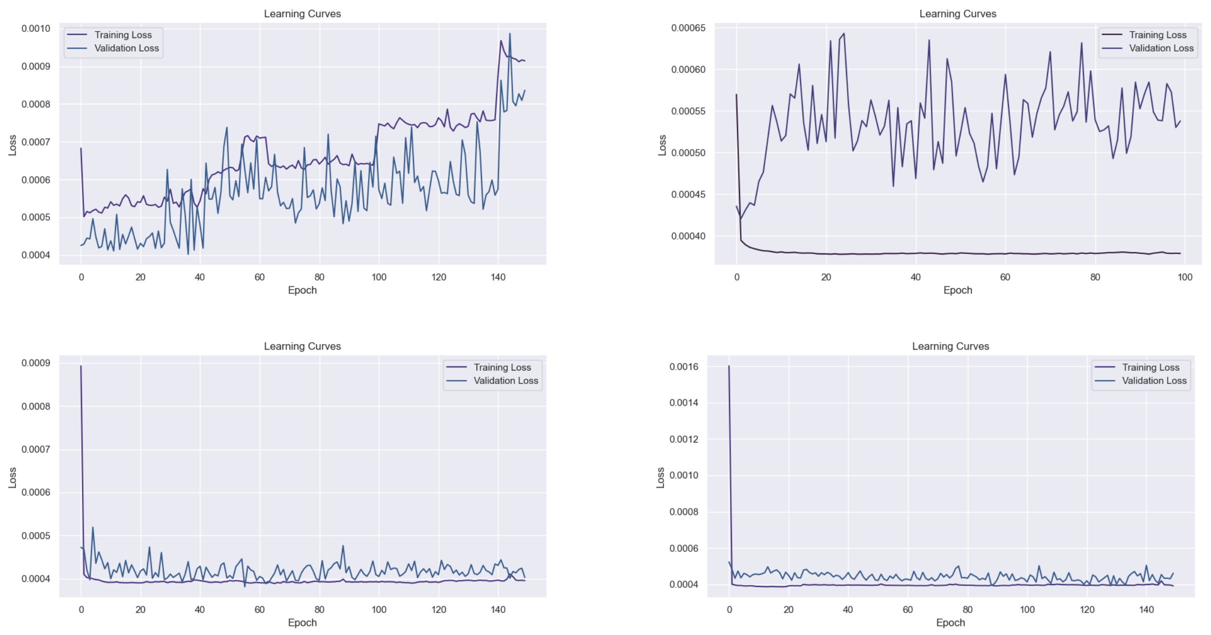


Figure 6: Training vs Predicted

Figure 7: Model Performances

2.2 Process in training

Even though the training process seems very easy. It has go through a process of evolution. At first, the Back Propagation neural network could not converge well.So we consider lower the learning rate, making it 10 times smaller from 0.01 to 0.001. After this adjustment, we find that the training loss quickly converges yet validation loss still not converges. Considering it is a large dataset with 1 million samples. We increase the batch size from 32 to 64. This not only boosting the speed of training but also make it converges better. This time, the validation loss converges yet we still try to improve it. Then we increase the batch size to 128. We apply the Adam optimizer and the MSE loss function, setting the training epochs to 150. All of these leads to the final learning curves.



2.3 Model Improvment and Advanced models

Though the two best model has a accuracy around 0.86, we still want to improve the model's performances. Through this process, we try multiple method. While discovering some interesting solutions, we find some of the traditional method did not work on this dataset. The following text will discuss it in detail.

2.4 Attempt on Complex Neural Network

When basic neural network does not live up to researchers' expectation, they usually try to use a more complex neural network. Yet this method seems useless on this dataset, indicating that training is not a very simple process. Take ResNet(Residual Neural Network) as an example, it solves the problem that degradation and are more complex than BP Neural Network. Disappointedly, the model performed even worse with a R^2 below 0.75:

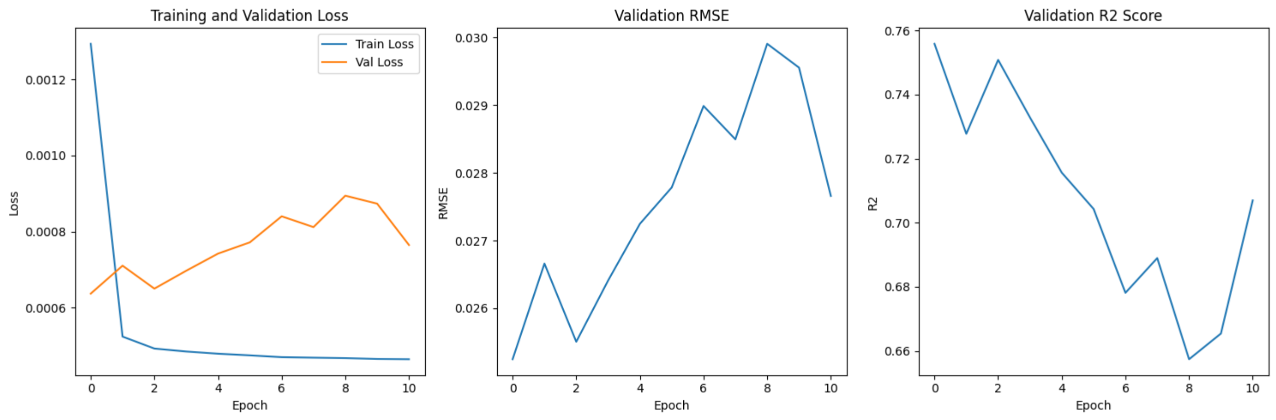


Figure 8: Residual Neural Network Performance

This indicates us that not all the problems can be solved by deeper and more complex network. To improve some of the work in data mining, choosing proper model after analyzing the specific problem is more important.

2.5 Training on Dimension reducted Dataset

So does dimension reduction helps this problem? We know that too many features will cause "Dimension curse" in KNN and Decision Tree. So we wonder if we apply PCA, will the model improve? However, dimension reduction works bad on this dataset. After applying PCA, the two most accurate models seem nearly loss all its original advantages. We may can say that model reduction does not work well on all the models.Similarly, model does not perform well on BP Neural Network, it does not become better and the converges worse Here we can directly present the comparsion of before and after using PCA in training:

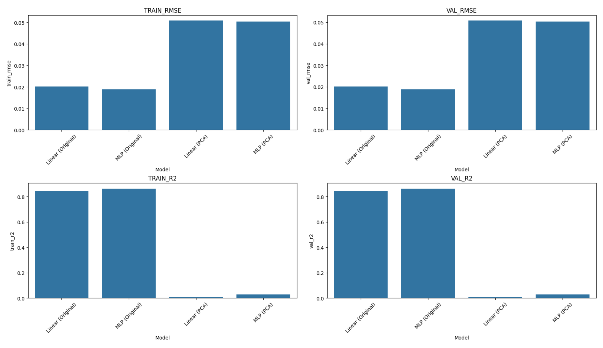


Figure 9: RMSE and R^2 on training and validation dataset

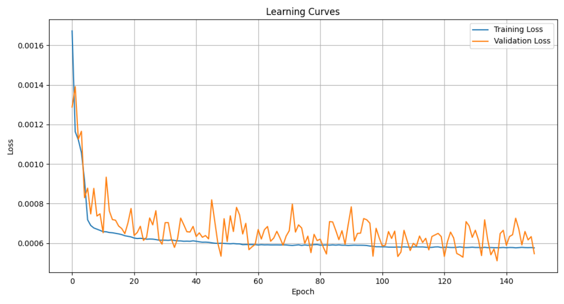


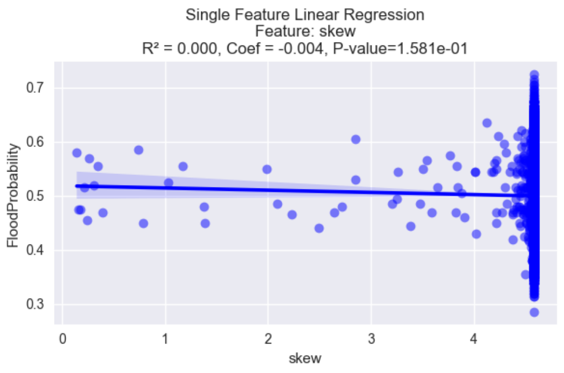
Figure 10: Training vs Predicted

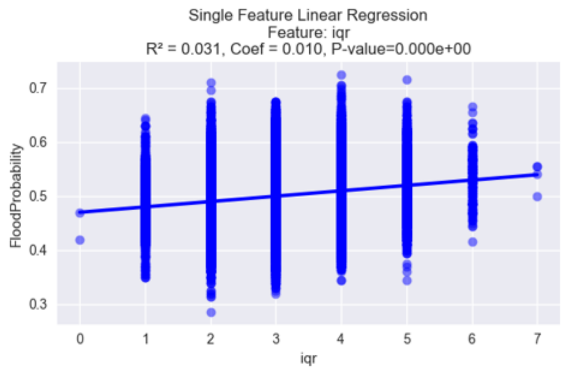
Figure 11: Model Performances

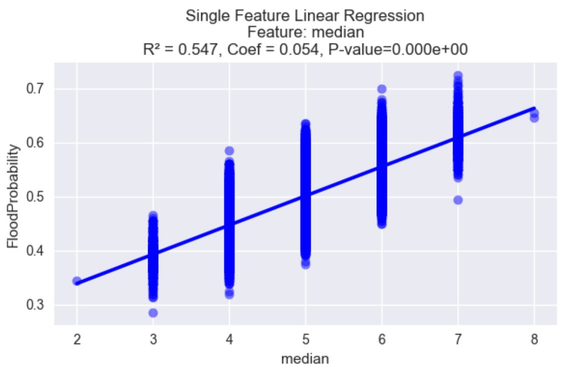
Simultaneously, after applying PCA on XGBoost, Random Forest, CatBoost and LightGBM, the model did not perform better, even if they are model based on Decision Tree.

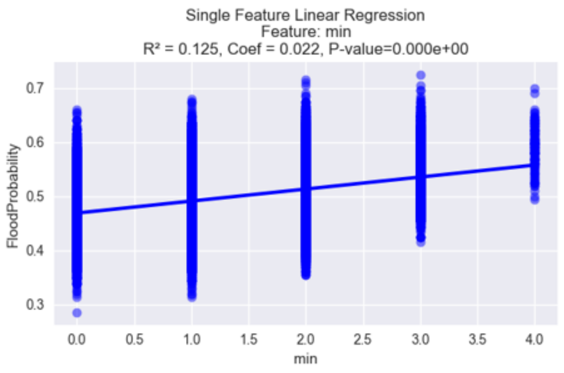
3 Improvement by applying Statistical features

The difficulty in improvement from above force us to think deeper towards the dataset. We discover that the some of the statistics features like minimum value, median value and inter quartile range have strong linear relationship with flood probability:









Considering this, we think they can help models to capture extra information and pattern inside the dataset. So we take these stats features as extra input into the models. Surprisingly, Tree models all improve their performances, especially random forest with R^2 increased by 20%. The extra statistics features did not improve the two best models. But the cross validation of MLP accuracy improved, meaning that model has stronger generlization ability after taking statistic feature as input. Besides,if only statistics features are used as input of linear regression, it still has a R-square of about 0.84. The table below shows the model performance after using statistical feature.

Table 2: Performance Comparison of Models with Statistical Features

Model	Training Time	Train R2	Val RMSE	Val R2	Cross Val Score
Random Forest with Stats	946.390741	0.980985	0.018698	0.865491	0.865355
XGBoost with Stats	4.570753	0.871007	0.018532	0.867873	0.868071
MLP with Stats	592.057320	0.861079	0.019016	0.860886	0.861895

Figure below directly shows the improvment of tree model.

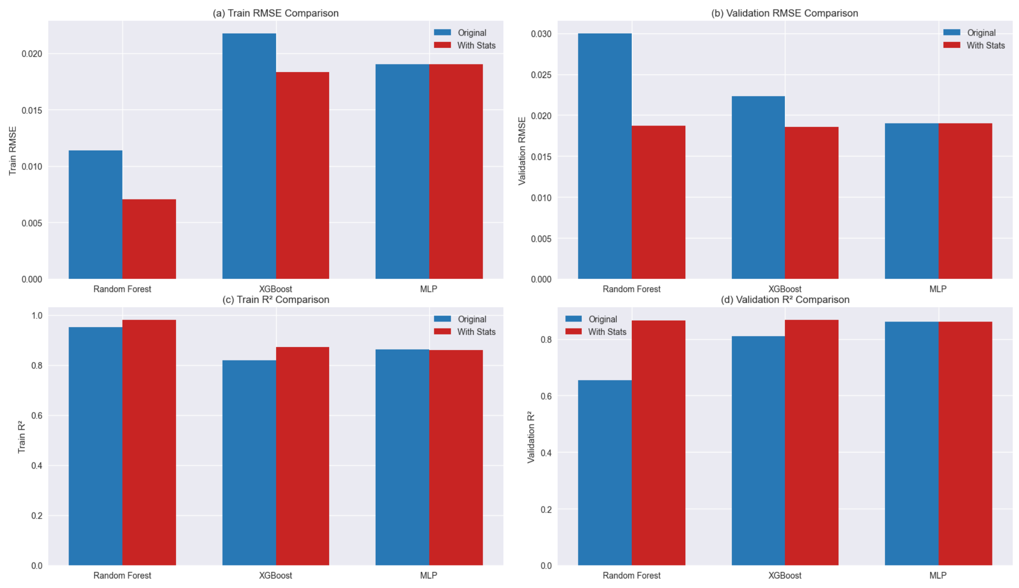


Figure 12: Models after applying stats

4 Ensemble learning

4.1 Training using Ensemble learning method

After the process above, we tried ensemble learning as in most of the time ensemble learning leads to a better performance. However, this time it disappointed us. The ensemble models perform not only worse but even cause overfitting problem.

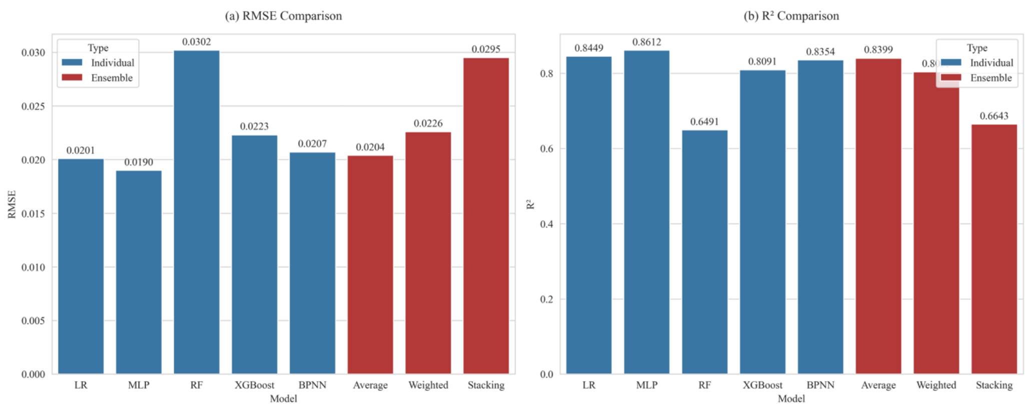


Figure 13: Model Comparsion

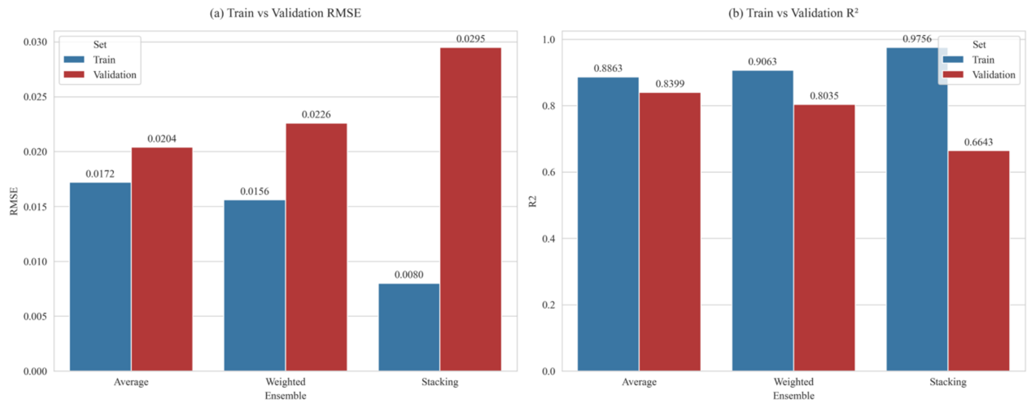


Figure 14: Ensemble model performance

We calculate the overfitting ratio of three ensemble model. The overfitting ratio of weighted ensemble method is larger than 1 and the ratio of stacking of model are larger than 3. These indeices shows the overfitting problem in these models. So ensemble learning appears to be a imappropriate choice for this problm

4.2 Why Ensemble learning failed?

We are curious about the cause of bad performance and overfitting, so we visualize the prediction distribution of each model on the test dataset:

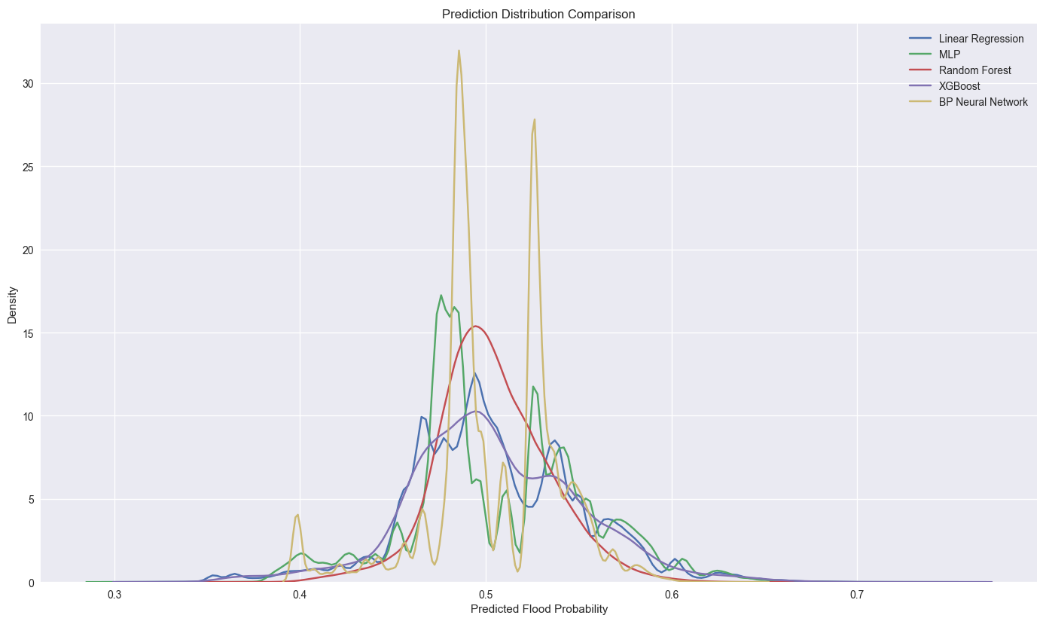


Figure 15: Prediction distribution of different models

As shown in the figrue, different model presents very different distribution on the test dataset. In comparison, BP neural network and MLP share a similar distribution but not the same. While tree models predict a distribution of normal distribution. So we think maybe different distribution sometimes make different models hard to reach to agreement. These fundamental differences in how models interpreted and predicted the data made it difficult for ensemble methods to effectively combine their predictions. Another possible point is that the dataset is unbalanced at some point, making ensemble learning not good for the model.

Besides reason above, when the underlying relationships are relatively simple or linear, combining multiple complex models may introduce unnecessary noise.The initial data exploration showed that most features had an average value of around 5 with limited variability, suggesting a relatively uniform dataset structure that might not benefit from diverse model combinations.

5 Future Work

In the future, some further improvement can be applied:

1. Applying ensemble learning using improved tree model: we wonder if the ensemble learning method is effective or not after the tree model is improved by using statistics feature. Besides, we can take a look at new prediction dataset as well.

2. We can also use the Autogluon library to improve the performance of model. In fact most of the best performed model on Kaggle are trained by Autogluon library. So this popular new library may be our next step of exploration.

6 Conclusion

This study on flood probability prediction models has revealed several important insights about model selection and improvement strategies. Our initial exploration showed that the dataset possessed unique characteristics - uniform feature distributions, weak correlations, and potentially simple underlying relationships. These characteristics led to unexpected results where simpler models like Linear Regression and MLP outperformed more complex approaches.

In this flood prediction case, theprediction patterns of different models and dataset characteristics led to ensemble methods actually degrading model performance and introducing overfitting. The research suggests that focusing on individual model optimization (such as incorporating statistical features) might be more effective than combining multiple models for this specific problem. This finding challenges the common assumption that ensemble methods consistently improve model performance.

The key finding is that model selection and improvement strategies should be guided by careful analysis of dataset characteristics rather than following conventional wisdom. In this case, the dataset's properties favored simpler, more direct modeling approaches over complex ensemble methods. Our research suggests that researchers should rather understand dataset characteristics than simply applying complex ensemble methods.

Appendix

Complete information are in the code file, here we present some detailed information related to the text above. Model performance comparsion in for tree model, linear regression and MLP.

Model	Before Dim. Reduction (R^2)	After Dim. Reduction (R^2)
BP Neural Network	0.793134	0.789800
Linear Regression	0.844968	0.007320
MLP Neural Network	0.861895	0.028577

Table 3: Model Performance Comparison Before and After Dimensionality Reduction

Model	Train R²	Val RMSE	Val R²
XGBoost (Original)	0.024512	0.025036	0.758849
LGBM (Original)	0.025238	0.025547	0.748902
CatBoost (Original)	0.762727	0.024871	0.762026
XGBoost (PCA)	0.050268	0.050519	0.018132
LGBM (PCA)	0.050400	0.050519	0.018137
CatBoost (PCA)	0.050569	0.050548	0.016976

Table 4: Model Performance Comparison Before and After Dimensionality Reduction

Part of the feature distribution:

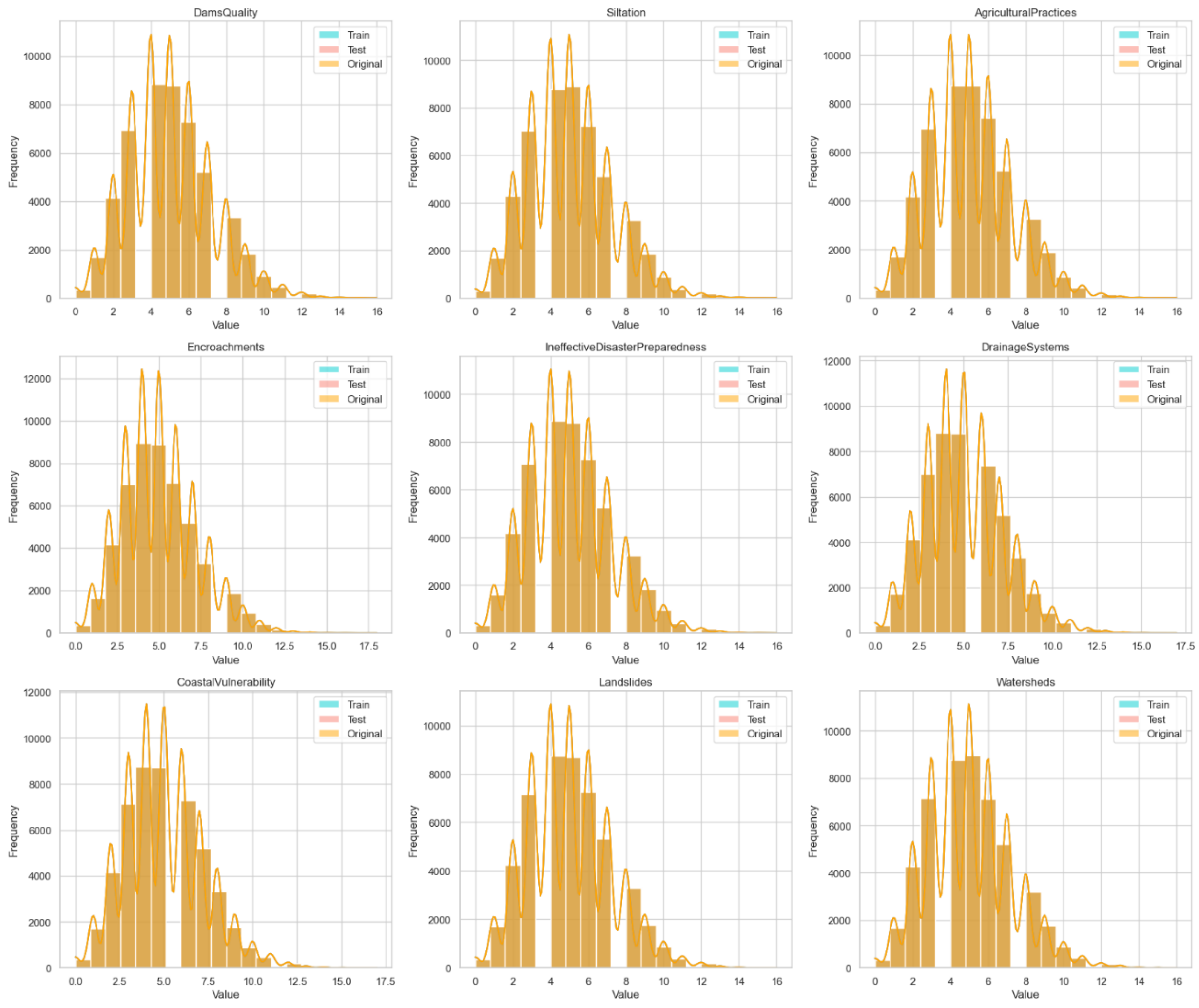


Figure 16: Caption