Table of Content

List of Figures

List of Tables

Acknowledgments

I thank above all the god who gave me the courage and the will to complete this work.

I deeply thank my supervisor Mrs. Seridi for guiding me with patience, and for all her efforts, advice and corrections.

My heartfelt thanks to my friends who have helped me.

Finally, a big thanks to my family for supporting me during my university course.

Dedication

To the whole family

To all my friends

Abstract

Flood susceptibility is the act of assessing and predicting the volume, timing, and length of floods based on known aspects of a river basin in order to avoid harm to people, property, and the environment. It's necessary for developing effective flood risk management plans, minimizing flood hazard and evacuating people from flood-prone areas.

In an effort to create a decision support system for flood susceptibility our main objective with this work is to compare the classification capability of advanced machine learning methods in this domain of flood susceptibility. We utilized the famous clustering algorithm known as K-means to achieve the class decomposition of our dataset with the aim to assess the benefits of diversity.

There is also is another concept precisely in flood susceptibility mapping where the intensity of the flood in a given area is manually calculated by hydraulic researchers from other variables like water elevation, then classified into different intensities like low, medium and high susceptibility, we aim to create a clustering mechanism that can capture these types of intensities in a non-supervised way.

Keywords: Flood susceptibility, Class decomposition, K-means, Machine learning

Résumé

L'étude de la vulnérabilité aux inondations est l'évaluation et la prédiction l'intensité, le moment et la durée des inondations en fonction des aspects connus d'un bassin hydrographique afin d'éviter de nuire aux personnes, aux biens et à l'environnement. Il est nécessaire pour élaborer des plans de gestion des risques d'inondation efficaces de réduire au minimum les risques d'inondation et évacuer les personnes des zones inondables.

Dans le but de créer un système d'aide à la décision pour la sensibilité aux inondations, notre principal objectif dans ce travail est de comparer la capacité de classification des méthodes avancées d'apprentissage automatique dans ce domaine de la sensibilité aux inondations. Nous avons utilisé le fameux algorithme de Clustering connu sous le nom de K-Means pour réaliser la décomposition de classe de notre ensemble de données dans le but d'évaluer les avantages de la diversité.

Il ya aussi un autre concept précisément dans la cartographie de la vulnérabilité aux inondations où l'intensité de l'inondation dans une zone donnée est calculée manuellement par les chercheurs hydrauliques d'autres variables comme l'élévation de l'eau, puis classés en différentes intensités comme faible, susceptibilité moyenne et élevée, nous visons à créer un mécanisme de regroupement qui peut saisir ces types d'intensités de manière non supervisée.

Mots-clés: Susceptibilité aux inondations, Décomposition des classes, K-means, Machine learning

Introduction

Flood susceptibility is the process of assessing and predicting the volume, timing, and length of floods based on known features of a river basin, with the goal of preventing harm to people, property, and the environment. It is required for establishing effective flood risk management strategies, reducing flood danger, evacuating people from flood-prone locations, and managing water resources systems. Knowing the different degrees of flood susceptibility helps prioritizing for the prevention and remediation of floods in different areas. In recent years, data-driven techniques/models for flood forecasting have gotten a lot of attention, incorporating modern machine learning techniques and algorithms like artificial neural network, decision trees, random forest, etc.

Floods are one of Algeria's most dangerous natural hazards. They result in several fatalities, property damage, and the destruction of roadways, public works, and infrastructure. Floods in Northwestern Algeria have caused significant damage in the past, such as at Mohammadia 1881, Mostaganem 1927, and El Asnam 1966 [1]. In this work we aim to create a decision support system with modern machine learning techniques by the classification of flood zones while implementing class decomposition to extract important insight from the datasets and improving the models performances.

There is also is another concept precisely in flood susceptibility mapping where the intensity of the flood in a given area is manually calculated by hydraulic researchers from other variables like water elevation, then classified into different intensities like low, medium and high susceptibility, we aim to create a clustering mechanism that can capture these types of intensities in a non-supervised way.

Structure of the thesis

Our work is organized as follows:

- Chapter 1 presents the state of the art in the field of Machine learning, Class decomposition and Flood susceptibility prediction.
- Chapter 2 details the conception and architecture of our system and all its components.
- Chapter 3 describes the implementation of the system as well as its results.
- And lastly a general conclusion and perspective.

Chapter 1: State of the art

1. Flood susceptibility

Flood susceptibility prediction is the act of assessing and predicting the volume, timing, and length of floods based on known aspects of a river basin in order to avoid harm to people, property, and the environment. It's necessary for developing effective flood risk management plans, minimizing flood hazard and evacuating people from flood-prone areas.

The analytical hierarchy process (AHP), the frequency ratio (FR) method, and the weights of evidence methodology have all been used to assess flood susceptibility. These methods, however, have drawbacks in terms of flood susceptibility; for example, the analytical hierarchy process produces uncertainties associated with ambiguous judgments based on expert knowledge, which is used to set the weights of influential factors, and the use of flood susceptibility is reliant on the sample size. Predefined assumptions about flood occurrence and the relevant influential elements are also disadvantages. Furthermore, many different types of models have been employed to simulate hydrological phenomena, including physically based, lumped, and statistical models; however, physical models have a number of flaws. Data restrictions cause uncertainties like this, and models require extensive and thorough field observations for parametrization. [2]. As a result, new techniques and methodologies for estimating flood risk in ungauged arid-region basins are required.

The physical mechanism of the hydrological process is not taken into account by data-driven models, which instead build a mathematical analysis of the time series and use the provided sample to determine the statistical or causal relationship between the hydrological variables. For resolving numerical prediction issues, recreating highly nonlinear functions, and examining time series, the data-driven approach offers particular advantages. Data-driven models have more applications now than ever before in the field of hydrological forecasting thanks to the quick advancement of computer technology. At the conclusion of the previous century, flood forecasting began to use artificial neural networks (ANNs), a popular data-driven methodology based on artificial intelligence [3].

Over the last two decades, the use of machine learning algorithms for flood susceptibility prediction has been intensively evaluated around the world. As a result, the recent development of machine learning methods has resulted in significant improvements in flood modeling, and these methods have become widely used due to their ability to capture data without using predefined assumptions and to process complex datasets with high levels of accuracy in short periods of time. Logistic regression (LR), artificial neural networks (ANNs), the adaptive neuro-fuzzy inference system (ANFIS), genetic algorithms (GAs), support vector machines (SVMs), and random forest (RF) models are some of the machine learning methods that have been used to predict flood vulnerability. For assessing flood risk, the RF model has been extensively employed. FFS has also been mapped using a variety of ML algorithms including fresh ML ensemble techniques. ML methods include a number of phases. First, accurate inventory datasets that can be used for both model training and model validation must be created. The second step is to choose the study area's geoenvironmental elements or possible flood conditions. Third, effective and appropriate machine learning models are used, and the performance of the models is evaluated using trustworthy evaluation indices [2].

In academic and applied scientific circles, there is increased interest in learning methods as a result of recent developments in the field of computer science. Modeling sequential data through a recurrent neural network (RNN), which is particularly ideal for hydrological prediction and provides an accurate and timely prediction of time series in systems, is one of the most active study areas in deep learning. The long short-term memory (LSTM), which has been effectively used for image recognition, the Internet of Things, text translation, and market prediction, is one of the more contemporary RNN architectures that have been developed since the late 1990s. The LSTM neural network is intended to address the issue of gradient disappearance in long-term dependent time series of simple RNN neural network; however, each LSTM unit comprises four affine transformations and each time step needs to be run once, which can easily take up the memory available [3].

2. Machine learning

Machine learning, also known as artificial learning, is a form of artificial intelligence (AI) that allows a system to learn from data and not through explicit programming. As algorithms ingest training data, it becomes possible to create more accurate models based on this data. A machine learning model is the result generated when you train your machine learning algorithm with data. After the training, when you provide input to a model, you receive an output result. a predictive algorithm creates a predictive model, when data is provided to the predictive model, it results in a forecast that is determined by the data that formed the model.

2.1. Artificial Neural Networks

An Artificial Neural Network (ANN) is a data or signal processing system made up of a large number of simple processing pieces linked together by direct linkages to execute parallel distributed processing in order to complete a specific computational goal. Neural networks act in a similar fashion to the human brain when it comes to processing information. Neural networks learn via example, similar to how biological nerve systems, such as the brain work. In comparison to traditional computing, ANN takes a distinct approach to problem solving. In order to solve a problem, traditional computer systems employ an algorithmic method, which entails following a series of instructions. This restricts our problem-solving abilities to problems that we are familiar with and can solve. Neural networks and traditional algorithmic computers, on the other hand, do not compete but rather complement each other. There are activities that are better suited to an algorithmic method, such as arithmetic operations, and tasks that are better suited to a neural network approach, such as image processing.

2.2. Decision Tree

A decision tree is a very simple model. Given several characteristics, the decision begins with one of these characteristics; if this is not enough, another one is used, and so on. It is widely known and used in many companies to facilitate the decision-making process and risk analysis. It was widely used in the 1960s and 1980s for the construction of expert systems. The rules are introduced manually, for this reason this model lost its popularity after the 80s. The emergence of mathematical methods to build decision trees has brought this model back to the battle of algorithms of automatic apparent drawing.

2.3. Random Forest (RF)

Random Forest is a decision tree ensemble technique in which each tree fits a piece of data taken separately via bootstrapping [4]. Due to the random selection at each split node based on the two data items, Out-Of-Bag and proximities, RF is known to deliver a resilient error rate with regard to outliers in predictors. As trees are added to the forest, OOB data is utilized to estimate variable relevance and an internal unbiased OOB error (the classification error), while bagging is used to randomly choose samples of variables as the training dataset for model calibration. If the values of each variable are permuted throughout the OOB data, the function calculates the model prediction error for that variable. On the other hand, proximity are used to replace missing data, finding outliers, and providing illuminating low-dimensional representations of the data can only be determined after each tree is fitted on for each pair of instances, then normalized by dividing the total number of fitted trees over it.

2.4. Support Vector Machine (SVM)

SVM is a novel mathematical tool that is utilized as a general constructive learning technique based on statistical learning theory rather than loose comparisons with natural learning systems [5]. SVMs solve nonlinear regression and classification problems by translating the input variables into a large-dimension space whose inner product is supplied by positive definite kernel functions, and then training them using dual optimization approaches with constraints [6].

3. Class decomposition

The technique of segmenting each class into a number of homogeneous sub-classes is known as class decomposition. Clustering is a natural way to accomplish this. Using class decomposition in supervised learning, particularly ensembles, can provide a variety of advantages. It can be a computationally efficient way to generate a linearly separable dataset without the requirement for feature engineering, which is required by techniques such as Support Vector Machines (SVM) and Deep Learning. Decomposition is a natural technique for ensembles to promote diversity, which is a fundamental aspect in the effectiveness of ensemble classifiers [7].

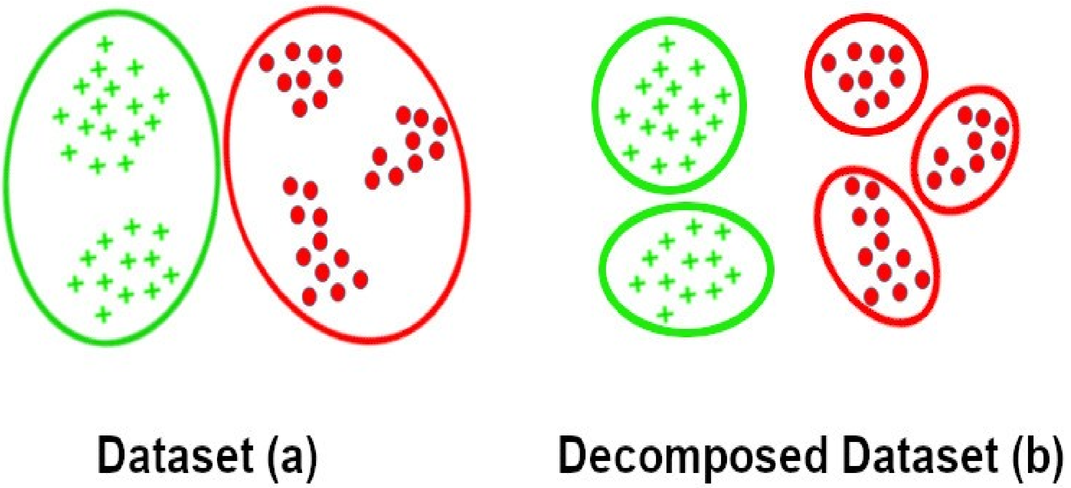

Figure 1: Class decomposition example

3.1. Problem

Many datasets, including the ones we used in this work, are binary datasets that have only two classes which are represented by the values: 0, False or 1, True. They are created after some classification is applied to the data, and thus in the process of predicting, which is called binary classification, the output is restricted to only two classes, but in some cases having more diverse classes in the dataset helps our model better understand and classify our data by learning the more in-depth patterns.

Having the output restricted to only two classes, we may miss important insight that might be found in our data, like sub-classes in medical datasets might represent a sub-type of a disease, knowing which sub-type of a disease a patient has helps doctors give a more precise and effective treatment to their patients.

3.2. Decomposition

Class decomposition facilitates learning class boundaries of a dataset for a machine learning model and consequently may improve the precision of the given model, as well as uncovering valuable insight from the dataset, there many ways of achieving class decomposition.

3.3. Related works

In 2019 Samih M. Mostafa and Hirofumi Amano [8] published an article where they highlighted the positive effect that clustering has on machine learning models accuracy, their work presents a technique where the data is clustered using the K-means algorithm, the number of clusters was determined by the elbow method which is a heuristic method of validation and interpretation of symmetry within cluster analysis, in this method the clustering processing step was done before applying the prediction algorithm.

In the experimentation phase the authors used MLR, Ridge, Lasso and ElasticNet machine learning algorithms and the results showed that the proposed method achieves significant improvement from the point of view of RMSE, and coefficient of determination R^2. The work done by these individuals that there are potential benefits to data clustering before applying a prediction algorithm.

Apart from the possible performance benefits that class decomposition may offer to a machine learning model, there are other benefits that can be that can be gained that are related to the insight we can extract from our datasets, the work done by Suchi Saria, Anna Goldenberg in 2015 [9] highlights the problem with binary medical datasets and that some additional important information can be extracted from them that can helps with doctors decision making and treatment choices, this approach is called precision medicine, which is the main focus of this article, the discovery and refinement of disease sub-types can benefit both the practice and science of medicine. Clinically, by refining prognoses based on similar individuals, disease sub-types help reduce uncertainty in an individual's expected outcome. Accurate prognoses can thereby improve treatment decisions.

They showed examples of binary datasets of different diseases where decompositing the positive class into multiple classes using statistical and machine learning approaches such as nonnegative matrix factorization, hierarchical clustering, and probabilistic latent factor analysis, reveals the presence of sub-types related to that disease.

4. K-means

Clustering is an unsupervised machine learning technique that divides the population or data points into several groups or clusters such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups.

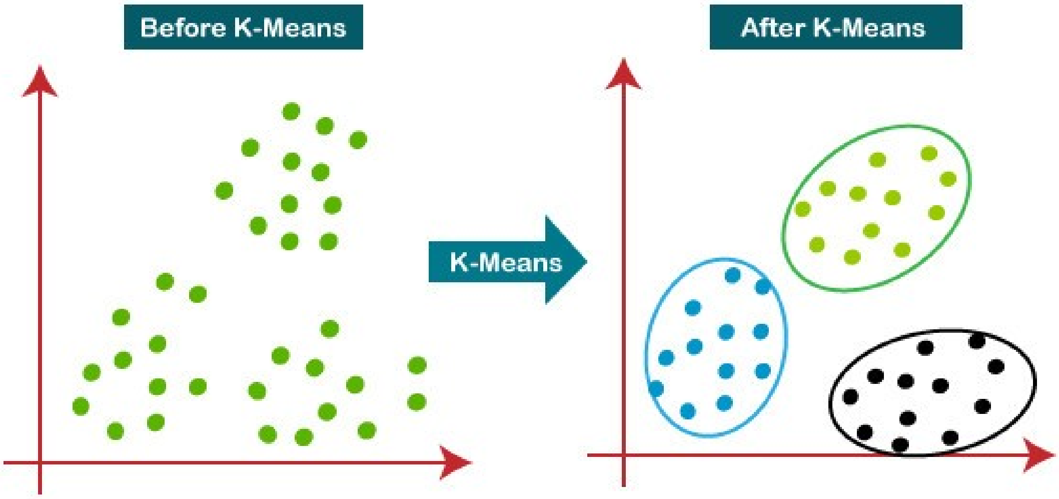

Figure 2: K-means clustering example [10]

K-means is a clustering technique that divides samples into subsets with the purpose of maximizing intra-subset similarity and inter-subset dissimilarity, where similarity is used to evaluate the link between two samples. The K-means clustering method is one of the most widely used unsupervised learning techniques, and it has been employed in a range of fields including artificial intelligence, data mining, biology, psychology, marketing, and medicine.

There are three parameters for the k-means algorithm that must be defined: the number of clusters, the k cluster initialization, and the distance metric.

4.1. The distance parameter

The distance between data points and cluster centers is commonly computed using the Euclidean distance metric in the k-means algorithm. K-means finds spherical or ball-shaped clusters in data as a result of this. Other distance metrics that k-means can use include the L1 distance, Itakura-Saito distance, Bergman distance, and Mahalanobis distance. The clusters become hyper-ellipsoidal when the Mahalonobis distance metric is used.

So the distance metric that should be used for k-means depends solely on the type of dataset we are using and it's shape, for example in the figure 3, if our dataset has spherical or ball-shaped clusters like dataset (a) it is better to use the Euclidean distance to better capture the ball-shaped clusters, and if our dataset contains ellipsoidal-shaped clusters, such as dataset (b), Mahalanobis distance should be used.

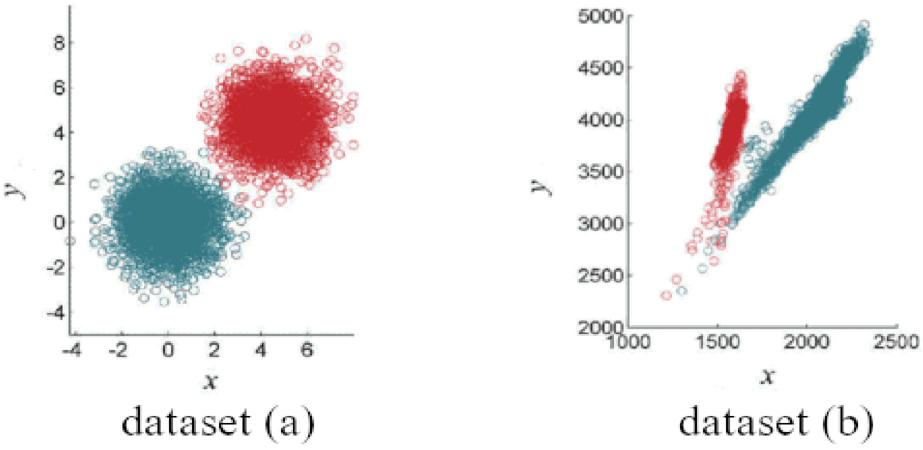

Figure 3: K-means datasets example

4.2. The K parameter (number of clusters)

A good choice of k results in better clustering and less iteration time for the centroids to converge. A poor choice of k, on the other hand, increases the number of iterations required for the centroids to converge, lowering performance. As a result, running the procedure on large datasets or with a poor choice of k is inefficient because it necessitates multiple iterations. The centroids may move a little in the last few iterations. Because extending such expensive iterations substantially affects performance, there must be some convergence criteria in place to ensure that the iteration ends when the criteria are met.

There also is a technique to address the number of clusters k issue called the elbow method, this method runs k-means clustering on the dataset for a range of values for k (say from 1-10) and then for each value of k computes an average score for all clusters. By default, the distortion score is computed, the sum of square distances from each point to its assigned center. When these overall metrics for each model are plotted, it is possible to visually determine the best value for k. If the line chart looks like an arm, then the “elbow” (the point of inflection on the curve) is the best value of k, an example is shown in figure 4. The “arm” can be either up or down, but if there is a strong inflection point, it is a good indication that the underlying model fits best at that point. [11]

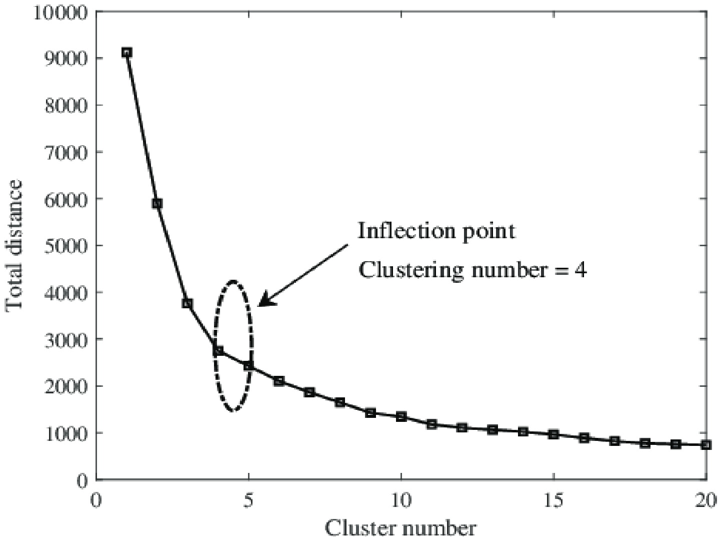

Figure 4: K-means elbow method

4.3. K-means++ (K-means plus plus)

K-means++ is a variant of k-means that improves clustering results through more clever seeding of the initial cluster centers, this approach assures the stability of the clusters and has many performance benefits [12],

In typical k-means the algorithm starts with random initial points (or seeds) then the algorithm finds it’s way towards the best cluster and centroids, now K-means++ tries to find the best initial

initial points/seeds instead of selecting them randomly, with this method it was proven that the algorithm gives better results and within a smaller time-frame. [13]

5. Conclusion

As a result of recent breakthroughs in the field of computer science, there has been a surge in interest in learning methods in academic and practical scientific circles.

Machine learning methods have been extensively examined for flood susceptibility prediction during the last two decades all around the world. As a result, the recent development of machine learning methods has resulted in significant improvements in flood modeling, and these methods have become widely used due to their ability to capture data without relying on predefined assumptions and to process complex datasets with high levels of accuracy in a short amount of time.

This prompted us to develop a decision support system that would allow concerned authorities to achieve better prevention and remediation of floods in different areas.

Chapter 2: Class decomposition and Flood susceptibility

1. Introduction

The main objective of this study is to assess and compare the classification capability of advanced machine learning methods in flood susceptibility for a decision support system while incorporating class decomposition with K-means, before reaching the decomposition and classification phase the datasets are ran through normalization then dimensional reduction processes this ensures the integrity of the data we give to our models for maximum accuracy, consistency, and context.

2. The Architecture

The system's architecture consists of the datasets passing through a treatment pipeline where the data is processed and normalized then feed to the machine learning algorithms, the algorithm's results are saved then used for visualization.

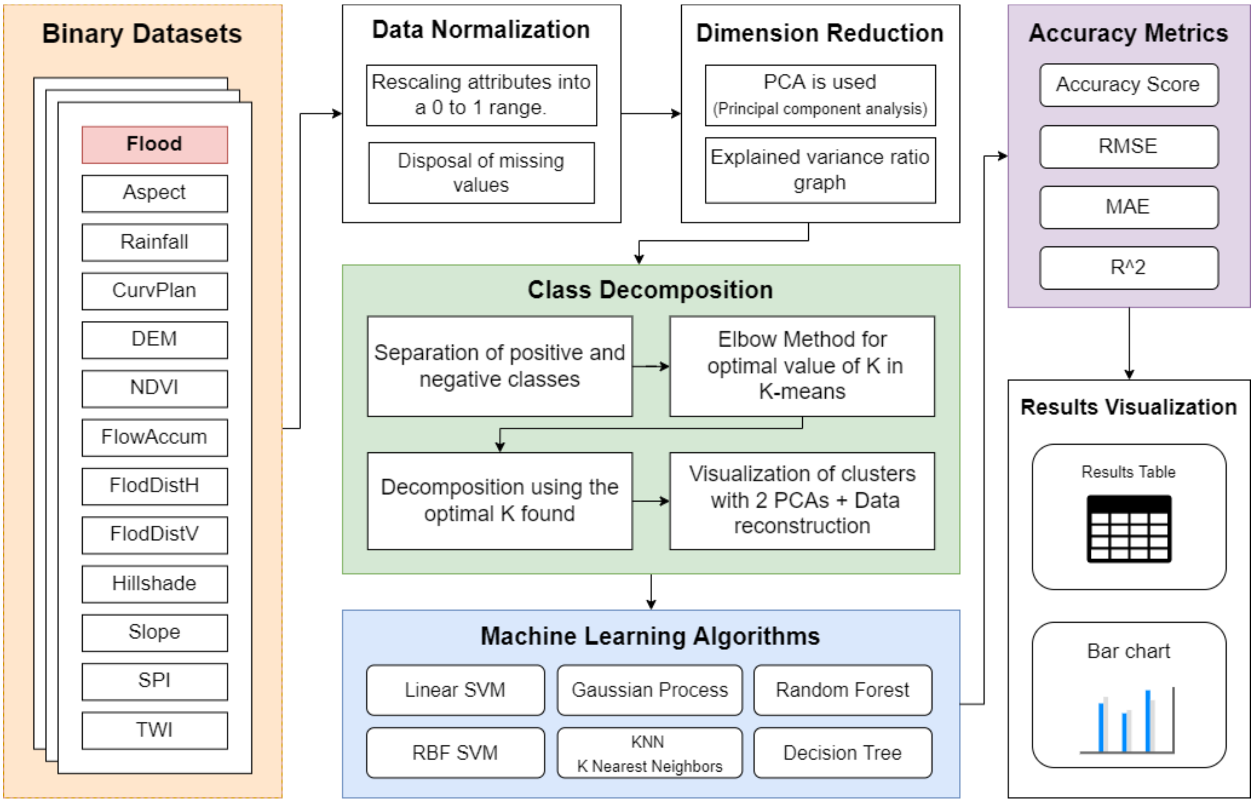
Binary Datasets
Flood
Aspect
Rainfall
CurvPlan
DEM
NDVI
FlowAccum
FlodDistH
FlodDistV
Hillshade
Slope
SPI
TWI
Data Normalization
Rescaling attributes into a 0 to 1 range.
Disposal of missing values
Dimension Reduction
PCA is used
(Principal component analysis)
Explained variance ratio graph
Accuracy Metrics
Accuracy Score
RMSE
MAE
R^2
Class Decomposition
Separation of positive and negative classes
Elbow Method for optimal value of K in K-means
Decomposition using the optimal K found
Visualization of clusters with 2 PCAs + Data reconstruction
Machine Learning Algorithms
Linear SVM
Gaussian Process
Random Forest
RBF SVM
KNN
K Nearest Neighbors
Decision Tree
Results Visualization
Results Table
Bar chart

Figure 5: Architecture's diagram

Figure 5 shows the architecture's diagram, each box represents a phase in the pipeline, the process starts from datasets (the left side) to finally the visualization of results (the right)

There are two points in the pipeline that require a supervised (manual) decision, the first one is the number of principal components for PCA in the dimensionality reduction phase, the second is in the class decomposition phase which is the K of K-means, in both cases a relevant graph is provided to help with the decision, Elbow Method graph for K-means and explained variance graph for PCA.

2.1. Normalization Phase

Normalization is a data preparation technique that is used in machine learning. The process of transforming the columns in a dataset to the same scale is referred to as normalization. Every dataset does not need to be normalized for machine learning. It is only required when the ranges of characteristics are different.

The most widely used types of normalization in machine learning are:

- Min-Max Scaling – Subtract the minimum value from each column's highest value and divide by the range. Each new column has a minimum value of 0 and a maximum value of 1.
- Standardization Scaling – The term "standardization" refers to the process of centering a variable at zero and standardizing the variance at one. Subtracting the mean of each observation and then dividing by the standard deviation is the procedure.

In this work we used the Mix-Max Scaling Normalization.

2.2. Dimensionality Reduction Phase

Reducing the number of feature variables for a dataset is referred to as dimensionality reduction. The input variables are the columns that are provided as input to a model to forecast/classify the target variable if the data is represented using rows and columns, because With a high number of dimensions in the feature space, the volume of that space can be rather huge, and the points (rows of data) that we have in that space frequently reflect a tiny and non-representative sample, this can have a significant influence on the performance of machine learning algorithms, therefore, reducing the amount of input features is typically beneficial.

PCA, or Principal Component Analysis, might be the most popular technique for dimensionality reduction. "PCA can be defined as the orthogonal projection of the data onto a lower dimensional linear space, known as the principal subspace, such that the variance of the projected data is maximized" [14], we opted to use PCA for dimensionality reduction in this work because it's very fast (easy to compute) and reduces noise.

The number of principal components is chosen manually, the explained variance graph is presented to aide with the choice, for example in figure 6 seven components (out of seven-teen) are explaining about 95% of the variance, so we can select seven as our number of principal components and effectively reduce the data dimension from seven-teen to seven without losing much information (only around 5%).

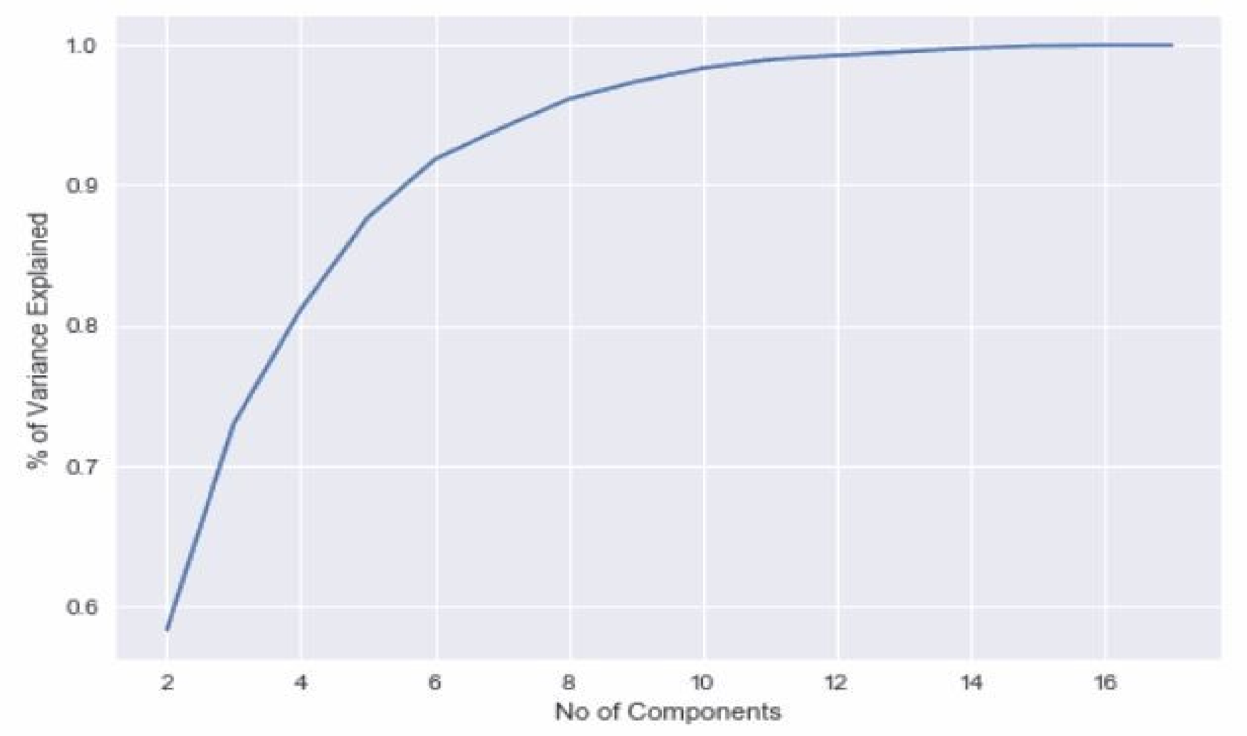

Figure 6: Explained variance graph example

2.3. Class Decomposition Phase

Class decomposition is the technique of separating each class into a number of meaningful sub-classes, there many ways to achieve class decomposition but in this work we used clustering with K-means, with K-means++ initialization.

The binary dataset is first separated by positive and negative classes then the decomposition is applied with K-means to each one of them separately, which means we will have multiple sub-classes originating from either from positive or negative classes, this ensures that a sub-class cluster containing rows from both positive and negative classes is avoided because it typically hinders the machine learning algorithm's performances. A good representation of this is shown in figure 1.

For the distance parameter we choose to use the Euclidean distance because our dataset contains mostly spherical or ball-shaped clusters and using the Euclidean distance helps better capture these shapes and therefore a better clustering, we determined that our data has spherical clusters by using TENE to plot a 3D graph then inspecting the result, upon inspection it's clear that there are spherical shaped clusters. Figure 7 shows the TSNE result graph.

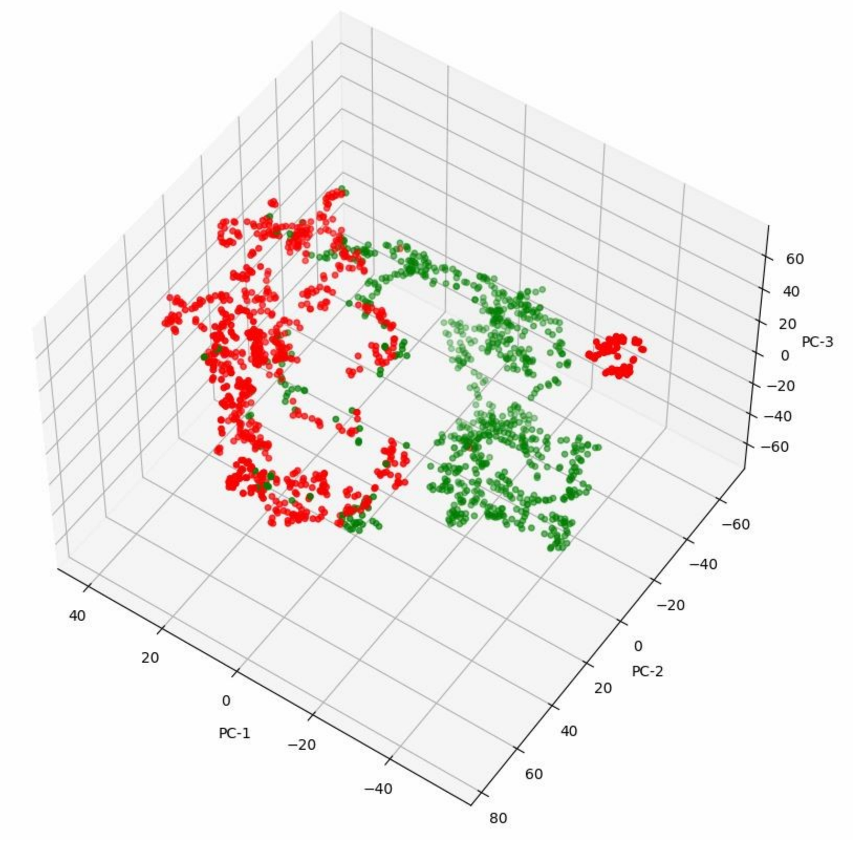

Figure 7: TSNE 3D graph

The number of clusters or K in K-means is picked manually in a supervised manner, the Elbow Method is used and the elbow graph is presented, the graph is plotted from the result of running k-means clustering on the dataset for a range of values for k, we choose the range 2 to 10, and then for each value of k we compute an average score for all clusters, this score is the Inertia which measures how well a dataset was clustered by K-Means, Inertia is calculated by measuring the distance between each data point and its centroid then squaring the distance, and summing them across one cluster. We would choose a value of k where we see an inflection point and the line begins to flatten out.

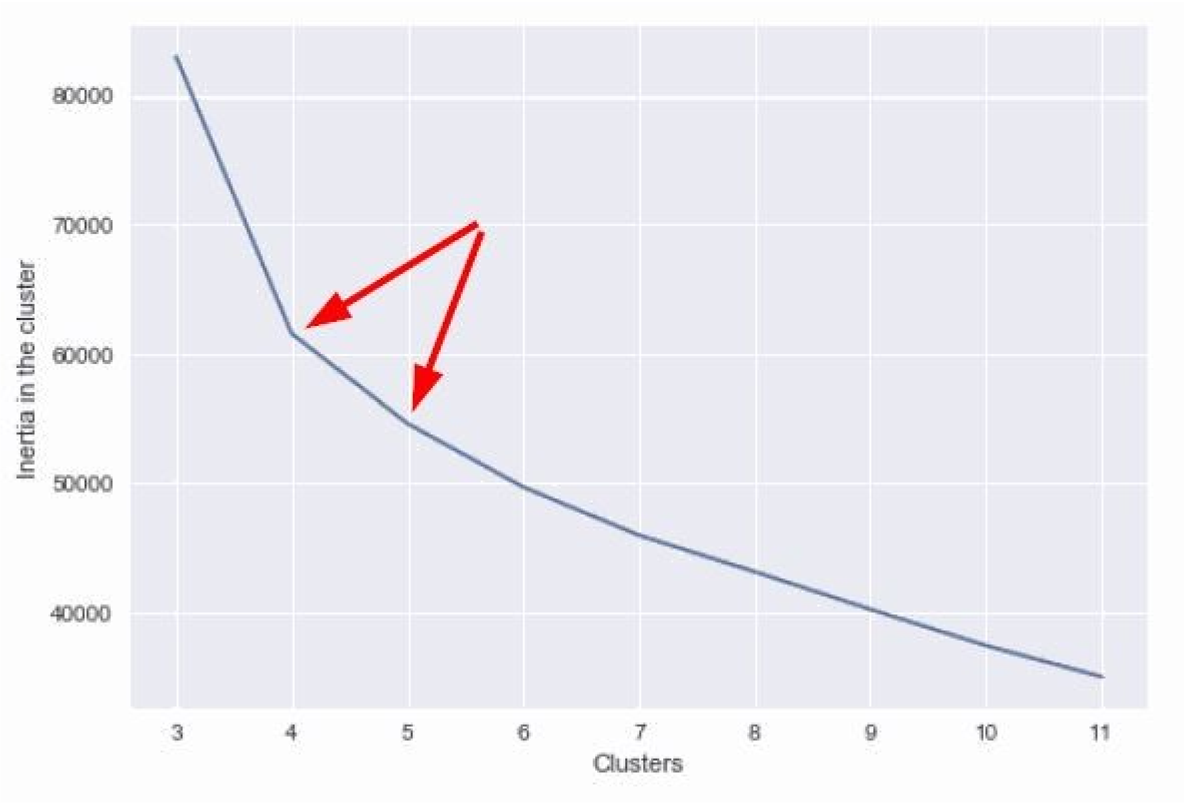

Figure 8: Elbow Method Graph with Inertia

The figure 8 show an example of the Elbow method graph, we can see an inflection in the points four and five and the line starts to flat out, we can try both of the values and evaluate the cluster to find with one of them works best.

2.4. Algorithms Phase

In this phase the machine learning algorithms are fit with the decomposed data, then tested, the data is first randomly shuffled then split into training and testing sets where 70% of the data is for training, and the remaining 30% for testing. We used multiple classifiers in this work and they are:

- Decision Tree
- Random Forest
- Artificial Neural Network
- Gaussian Process
- Nearest Neighbors
- Linear SVM
- RBF SVM

The metrics saved are accuracy score (ACC), root mean squared error (RMSE), mean absolute error (MAE) and coefficient of determination score (R^2). These results are saved then used in the Metrics/Visualization phase.

3. Conclusion

With this architecture we will be able to easily assess and compare the classification capability of multiple machine learning methods in flood susceptibility with different datasets, the pipeline like concept makes it easy to separate the different phases of this process and see feedback of each phase incrementally while the system is executing.

In the next chapter we'll go through the architecture's implementation, as well as its performance and outcomes

Chapter 3: Experimentation and Results

1. Introduction

In this chapter we will go over the implementation of the system, the architecture consists of multiple phases where the data passes through.

2. Technologies Used

2.1. Python (Programming language)

Python is an open-source programming language that is interpreted, interactive, and object-oriented. Modules, exceptions, dynamic typing, extremely high-level dynamic data types, and classes are all included. Python is a powerful programming language with a simple syntax. It is expandable in C or C++ and offers interfaces to numerous system functions and libraries, as well as multiple window systems. It can also be used as an extension language for programs that require a configurable user interface [15].

Python is cross-platform, running on a variety of Unix (Linux), Mac OS and Windows systems. It's also garbage-collected and dynamically typed. It supports a variety of programming paradigms, including structured (especially procedural) programming, object-oriented programming, and functional programming. Because of its extensive standard library, Python is frequently referred to as a "batteries included" language.

2.2. Scikit-learn Library

Scikit-learn is a free and open-source Python library for machine learning. It was developed by many contributors and is used by many individuals especially in the academic world, institutes of higher education and research. This library contains a wide collection of machine learning algorithms and datasets and can be used to achieve various tasks including classification, regression, preprocessing, clustering, dimensionality reduction and model selection [16].

This library provided the machine learning algorithms needed for our system.

2.3. Visual Studio Code

VS Code in short, is an integrated development environment (IDE) and code editor that provides all the necessities to write and run python code, we used it’s integrated support for Jupyter Notebooks to write our system in a notebook style where each phase is separated into its own code block.

3. Dataset

The used dataset used in this work is a binary datasets containing 12 features plus the class feature which a contains either 1 for positive case and 0 for the negative case for flood and nonflooded points respectively, these features 12 features are Slope, Aspect, Rainfall, Plan curvature (CurvPlan), Hillshade, Topographic wetness index (TWI), Sediment transport index (STI), Flow accumulation (FlowAccum), Flow distance (FlodDistH/FlodDistV), Digital Elevation Models (DEM) and Normalized Difference Vegetation Index (NDVI).

This dataset that we will be running our evaluation on is of a Vietnamese basin, this dataset contains 1675 records, 827 are positive (flood) and the other 848 are negative (non-flood)

3.1. Features description

Slope: Slope has a significant influence on flooding (Meraj et al. 2018) due to its effects on water velocity and surface runoff (Torabi Haghighi et al. 2018). Steep slopes contribute to a high water velocity and increase the flow volume in downstream areas (Chen et al. 2020).

Aspect: Aspect is important for flooding (Choubin et al. 2019), and many hydrologic parameters are influenced by aspect. When an aspect receives low and intense sun, soil moisture will increase; consequently, the moist slope will generate runoff, contributing to flooding risks.

Plan curvature: Plan curvature is considered an important and essential floodinfluencing factor by many researchers (Hong et al. 2018) and affects hyporheic conditions and heterogeneity.

Hillshade: Hillshade or toposhade is directly related to the shade and length of hillslopes, which may affect the convergence of overland flow (Aryal et al. 2003).

Flow accumulation: Flow accumulation can be estimated from flow direction parameters to show the accumulation of flows among pixels, thus, this factor is important in FSM.

Topographic wetness index (TWI): the TWI expresses the water accumulation quantity per unit (or cell) in a watershed considering the downstream flow trends due to the gravitational forces. (Proposed by Beven and Kirkby (1979)).

Sediment transport index (STI): The processes of erosion and deposition can be reflected by the STI, The changes in a riverbed caused by these processes (Tehrany and Jones 2017).

Flow distance: The distance from main rivers or streams has a considerable impact on the flooding occurrence in a given area (Glenn et al. 2012).

Rainfall: Precipitation has a considerable effect on flooding; notably, without rainfall, flooding would not occur.

Digital Elevation Models (DEM): Open-access global Digital Elevation Models have been crucial in enabling flood studies in data-sparse areas.

Normalized Difference Vegetation Index (NDVI): Normalized Difference Vegetation Index data is extracted from Moderate-Resolution Imaging Spectroradiometer (MODIS).

4. Implementation

4.1. Normalization

In the normalization phase we applied the min-max normalization technique on our dataset, min-max is the simplest method of normalization which transforms and scales each feature individually such that it is in the given range. we used Scikit-learn's "`MinMaxScaler`" processing module and we set the range to be between 0 and 1.

4.2. Dimension Reduction with PCA

In Dimension Reduction we used Scikit-learn's "`PCA`" decomposition module which takes as an input the number of components and the data.

We first need to plot the explained variance graph, we achieve this by running the PCA algorithm for a range of values that is from 2 to the number of features, in our case 12, then we sum each time the explained variance ratio and finally plot the result in a line chart.

By inspecting the line chart we pick the optimal value for the number of components and pass it again to PCA. The resulting dataset is then passed to the next phase for further treatment.

4.3. Class decomposition

In this phase we start by separating the positive and negative classes into their own variables, we are using the library pandas's DataFrame which makes this process very easy.

After that, we plot the Elbow Method's graph and we achieve this by running K-means through a range of values, we choose the range 2 to 10, and then saving the inertia for each iteration, we plot the values into a line chart which results in an elbow shaped graph.

After picking picking a value for K by inspecting the elbow graph, we pass it to K-means again and save the resulting clusters, we then reconstruct our dataset with the new classes that are represented by the clusters. The data is passed to the next phase.

4.4. Applying Machine Learning Algorithms

We first need to split our dataset into two sets, training and testing set, we used Scikit-learn's "`train_test_split`" handy function to do just that, the training set contains 70% of the data while the testing set contains the remaining 30%.

In this phase we have a collection of machine learning algorithms that we run through and fit them to the new decomposed dataset, after fitting them to the training set we then use testing set to evaluate and them record multiple evaluation metrics which are the accuracy score (ACC), root mean squared error (RMSE), mean absolute error (MAE) and coefficient of determination score (R^2).

The machine learning algorithms we used in this phase, which all of them are classifiers, are Decision Tree, Random Forest, Gaussian Process, Nearest Neighbors, Linear SVM and RBF SVM each implementation of these algorithms is provided by the Scikit-learn library

These result metrics are saved and then passed to the next phase.

4.5. Results Metrics and Visualization

In this phase the we use the resulting metrics from the models evaluation to visually represent the outcome, this includes a comparative performance table and bar chart, we used Python's matplotlib pyplot library which provides an implicit, MATLAB-like, way of plotting.

5. Results

We ran our evaluation on the default binary dataset which is a simple binary classification then used the decomposition method and ran the evaluation again but this time as multi-class classification, we will be comparing the different results of the binary and decomposed datasets.

We did a Dimension Reduction with PCA for both decomposed and binary datasets where the number of principal components is 7, we choose the number 7 because from the Explained variance graph, 7 components explained 95% of the variance in our data.

In the decomposed dataset we separatly clustered the negative and positive classes into 3 clusters each, which means we now have 6 classes where 3 of them are originating from the positive class and the other 3 are originating from the negative class.

We choose the number of 3 clusters (or the number K of K-means) after inspecting the Elbow Method graph which showed an apparent inflection in the 3rd point coincidentally for both the positive and negative classes

5.1. Binary dataset

Accuracy bar chart

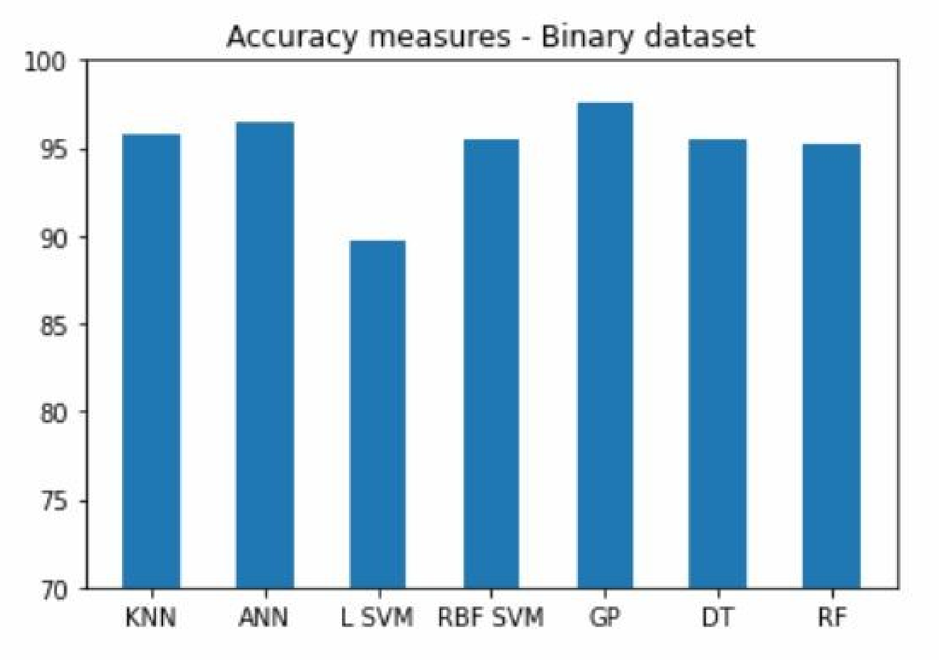

Figure 9: Bar chart accuracy measures of binary dataset

Comparative table

Model	ACC	RMSE	MAE	R2
KNN	95.83%	0.0417	0.0417	0.8329
ANN	96.42%	0.0358	0.0358	0.8568
L SVM	89.66%	0.1034	0.1034	0.5863
RBF SVM	95.43%	0.0457	0.0457	0.8170
GP	97.61%	0.0239	0.0239	0.9045
DT	95.43%	0.0457	0.0457	0.8170
RF	95.23%	0.0477	0.0477	0.8091

Table 1: Comparative results table of binary dataset

5.2. Decomposed dataset

Accuracy bar chart

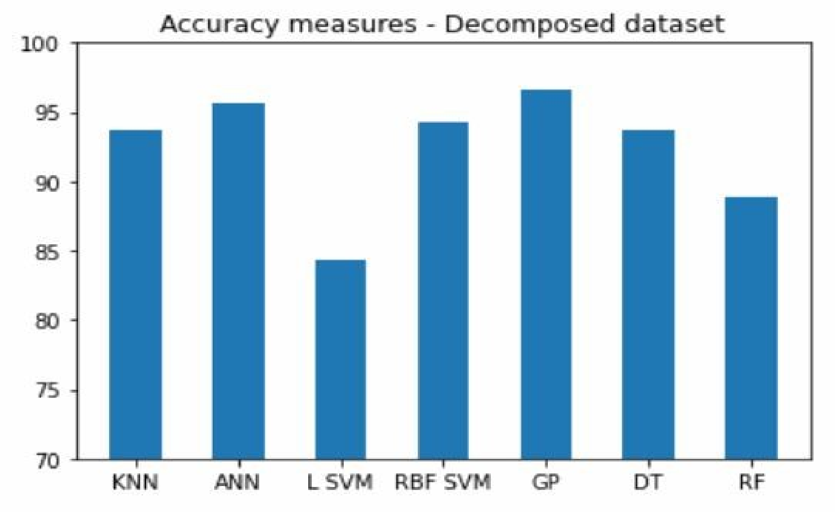

Figure 10: Bar chart accuracy measures of decomposed dataset

Comparative table

Model	ACC	RMSE	MAE	R2
KNN	93.64%	0.3201	0.1252	0.8518
ANN	95.63%	0.3797	0.1173	0.8242
L SVM	84.29%	1.3698	0.4314	0.3657
RBF SVM	94.23%	0.4215	0.1392	0.8048
GP	96.62%	0.3101	0.0954	0.8564
DT	93.64%	0.3380	0.1312	0.8435
RF	88.87%	0.6064	0.2286	0.7192

Table 2: Comparative results table of decomposed dataset

Conclusion and Perspectives

From the previews results we can note from the accuracy metrics that the best preforming machine learning algorithms are the Gaussian Process Classifier and Artificial Neural network, both have an above 95% accuracy for the two datasets, this is probably due to the complex nature of our dataset, we can see from the TSNE graph (Figure 7) many complex patterns that a standard classifier like SVM would fail to grasp, which explains the below 90% accuracy of the L SVM for both binary and decomposed datasets, 89.66% and 84.29% respectively.

We can also see that our top preforming classifiers are have almost the same accuracy in both binary and decomposed datasets, going from 2 classes to 6 classes with only 1% difference, in which we conclude that there’s a pattern or meaning in the generated clusters that the classifiers are able to apprehend and perceive.

To assess that matter and find out what the generated clusters represent, our colleges from the department of hydraulics helped us with the projection and mapping of our decomposed dataset, the result is a map of the area with our classes projected in-top of it represented with different color shades.

…

There is room for further enhancements and optimizations on the models from our ends for them to reach their optimal performance.

From this experiments we conclude that there is more insight that we can uncover from datasets through clustering.

References

[1]: Sardou Miloud, Said Maouche, Hanifi Missoum, Compilation of historical floods catalog of northwestern Algeria: first step towards an atlas of extreme floods, 2016, https://www.researchgate.net/publication/303478344_Compilation_of_historical_floods_catalog_of_northwestern_Algeria_first_step_towards_an_atlas_of_extreme_floods

[2]: Mohamed Saber, Tayeb Boulmaiz, Mawloud Guermoui, Karim Abdrabo, Examining LightGBM and CatBoost Models for Wadi Flash Flood Susceptibility Prediction, 2021, https://www.researchgate.net/publication/354236626_Examining_LightGBM_and_CatBoost_Models_for_Wadi_Flash_Flood_Susceptibility_Prediction

[3]: Yuanhao Xu, Caihong Hu, Qiang Wu, Zhichao Li, Application of temporal convolutional network for flood forecasting, 2021, https://www.researchgate.net/publication/353096681_Application_of_temporal_convolutional_network_for_flood_forecasting

[4]: L. Breiman, Random forests, 2001, https://link.springer.com/article/10.1023/A:1010933404324

[5]: N. Cristianini, B. Schölkopf, Support Vector Machines and kerNel Methods: The New Generation ofLearning Machines, 2002, http://dx.doi.org/10.1609/aimag.v23i3.1655

[6]: X. Yao, F.C. Dai, Support Vector Machine Modeling of Landslide Susceptibility Using a Gis: A Case Study, 2006, https://www.researchgate.net/publication/238431179_Support_vector_machine_modeling_of_landslide_susceptibility_using_a_GIS_A_case_study

[7]: Eyad Elyan, Mohamed Medhat Gaber, A Genetic Algorithm Approach to Optimising Random Forests Applied to Class Engineered Data, 2016, https://www.researchgate.net/publication/305875413_A_Genetic_Algorithm_Approach_to_Optimising_Random_Forests_Applied_to_Class_Engineered_Data

[8]: Samih M. Mostafa, Hirofumi Amano, Effect of clustering data in improving machine learning model accuracy, 2019, https://kyushu-u.pure.elsevier.com/en/publications/effect-of-clustering-data-in-improving-machine-learning-model-acc

[9]: Suchi Saria, Anna Goldenberg, Subtyping: What It is and Its Role in Precision Medicine, 2015, https://jhu.pure.elsevier.com/en/publications/subtyping-what-it-is-and-its-role-in-precision-medicine-3

[10]: docsity, Study notes for Data Mining, Chapter-3 Linear Regression, K-means, -, https://www.docsity.com/en/chapter-3-linear-regression-1/7480857/

[11]: Sklearn, Sklearn guides: Elbow Method, -, https://www.scikit-yb.org/en/latest/api/cluster/elbow.html

[12]: David Arthur, Sergei Vassilvitskii, k-means++: the advantages of careful seeding, 2007, https://dl.acm.org/doi/10.5555/1283383.1283494

[13]: Sindhuja Ranganathan, Improvements to k-means clustering, 2013, https://www.semanticscholar.org/paper/Sindhuja-Ranganathan-Improvements-to-k-means-Master-Ranganathan-Elomaa/8b2d83a9ec94ca6a235a2b5e354a29202ff416ab

[14]: Christopher M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), 2006, https://link.springer.com/book/9780387310732

[15]: Kuhlman, Dave, A Python Book: Beginning Python, Advanced Python, and Python Exercises, 2009, https://books.google.dz/books/about/A_Python_Book.html?id=mrjOygAACAAJ

[16]: Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Scikit-learn: Machine Learning in Python, 2012, https://www.researchgate.net/publication/51969319_Scikit-learn_Machine_Learning_in_Python