USING GENETIC ALGORITHM FOR FEATURE SELECTION

A PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF

MASTER OF TECHNOLOGY
IN
INFORMATION TECHNOLOGY

Submitted by

KARAN (2K21/ISY/011)

Under the supervision of

Prof. KAPIL SHARMA

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi 110042

MAY, 2023

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, Karan Yadav, Roll No – 2K21/ISY/011 student of M.Tech (IT-ISY), hereby declare that the project Dissertation titled "Using Genetic Algorithm for feature selection" which is submitted by me to the Department of Information Technology, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Karan Yadav

Date: 2K21/ISY/11

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled “Using Genetic Algorithm for feature selection” which is submitted by Karan Yadav, Roll No – 2K21/ISY/011, IT-(ISY) ,Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Prof. KAPILSHARMA

Date: SUPERVISOR

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

I wish to express my sincerest gratitude to Prof. Kapil Sharma for his continuous guidance and mentorship that he provided me during the project. He showed me the path to achieve our targets by explaining all the tasks to be done and explained to me the importance of this project as well as its industrial relevance. He was always ready to help me and clear my doubts regarding any hurdles in this project. Without his constant support and motivation, this project would not have been successful.

Place: Karan Yadav

Date: 2K21/ISY/11

Abstract

With the rise of machine learning, we are discovering better more complex, and performant models and new data-sets to use on those models. But as the dataset keep getting bigger and bigger, the time to train the model also increases. Feature selection helps in keeping in check this training time all the while improving the model performance by avoiding overfitting. As discussed in this thesis there are many types of feature selection algorithms. Each of those types has its pros and cons. We are focusing on genetic algorithm as the feature selection method of our choice. We compare genetic algorithm-based feature selection algorithm with five other feature selection methods. We found that the genetic algorithm based feature selection method has comparable performance while having lower error. We also use genetic algorithm-based feature selection with KNN to detect the mechanism of action of drugs and find that the Cross-validation score increases substantially from first generation to the twentieth generation. Showing that genetic algorithm as a feature selection method of choice is comparable if not better to other feature selection methods available.

Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

In today's fast-paced world, advancements in technology have made way for transforma-tive innovations. Machine learning has emerged as a powerful tool that is revolutionizing many industries. Its impact ranges from improving user experiences to enabling ground-breaking discoveries.

Machine learning has unlocked the hidden potential of plethora of data we have. It enables computers to learn from patterns, make accurate predictions, and make valuable insights which are beyond human capabilities. With the growth of data in almost every domain, machine learning has allowed us to extract knowledge, optimize processes, and drive intelligent decision-making.

In the field of business, machine learning powers personalized recommendations (for example recommendations on Flipkart), fraud detection systems, and demand forecasting models. It allows companies like Amazon [1] and Netflix [2] to personalize their offerings to individual preferences, improving customer satisfaction and increasing revenue. Massive datasets are analysed by machine learning algorithms, which find patterns and trends that help companies make optimal decisions.

One area where machine learning holds amazing promise is healthcare. The capability of machine learning to improve patient prognosis, diagnosis, and treatment is tremendous. In medicine, timely and accurate decision-making is critical, and machine learning offers many opportunities in this regard.

Diagnosing diseases can be complex, often requiring expert knowldge and analysis of huge amount of patient data. Machine learning algorithms, when trained on comprehen-sive medical datasets, can assist doctors in making more accurate diagnoses [3][4] which in turn will reduce cost and decrease mortality. By analyzing symptoms, genetic markers, and medical imagng, machine learning models can identify patterns and provide valuable insights to helping in early detection and treatment planning. This can be the difference between life and death in diseases such as cancers where early detection [5][6] has a very good prognosis in most cases.

Machine learning also has great potential in personalized medications, where treat-

ments are tailored to an individual's unique characteristics. By analyzing genetic profiles, medical histories, and treatment outcomes, machine learning algorithms can help predict the effectiveness of different therapies for a particular patient. This approach can revolu-tionize the field, enabling more precise and targeted interventions all the while lowering the cost of healthcare.

Also, machine learning plays an important role in drug discovery and development [7]. Traditional methods for identifying new drugs are time-consuming and costly due to which we haven't discovered a new antibiotic in quite a while, increasing the chances of a superbug pandemic. Machne learning models can analyze vast amounts of data, including genetic information, molecular structures, and clinical trial results. This can help to identify potential drug candidates and accelerate the drug discovery process. This helps with more efficient drug development and the possibility for faster access to life-saving treatments all the while lowering the cost of healthcare.

As discussed above, Machine learning has become a game-changer in various fields, from helping us find the best products online to diagnoising diseases. But not all features or attributes in our data are equally important. Feature selection, is a critical step in machine learning. It helps us choose the right features and improve the performance of our machine learning models.

Let's say we have a dataset with many different features, each having some informa-tion. However, not all of them are relevant and equally to solving the problem at hand. Including irrelevant in our model can lead to poor performance.

Here's where feature selection comes handy. It helps us identify the most informative and important features that have a significant impact on the result we want to predict. By selecting the right features, we can reduce noise, imprve model accuracy, and speed up training and prediction processes.

Feature selection offers several benefits in machine learning: Improved Model Perfor-mance: By focusing on the most relevant features, we can avoid overfitting, in overfitting the model memorizes the training data instead of learning patterns. Feature selection helps our model more robust to new, unseen data and achieve higher accuracy.

Enhanced Interpretability: When we select important features, it becomes easier to understand the relationships between input variables and the predicted result. This is especially important in fields like healthcare, where doctors and researchers need to un-derstand, explain and trust the model's predictions.

Faster Training and Prediction: Including unnecessary/unimportant features in our models can slow down the training and prediction processes (wasting precious computa-tion resources to incorporate unimportant features). Feature selection helps us eliminate irrelevant features, reducing the computational resource wastage and improving efficiency.

Handling High-Dimensional Data: Many real-world datasets contain a large number of features. Feature selection allows us to handle such data by reducing it to a smaller

set of the most important features. This simplifies the problem, making it easier to train our models for the data.

Methods of Feature Selection: There are various methods for feature selection, includ-ing:

Univariate Selection: This method selects features based on statistical factors, and each feature is evaluated individually.

Recursive Feature Elimination:Features are recursively eliminated by training the machine learning model on subset of features and then comparing performance.

Regularization Techniques: A penalty is added to the model's objective function which forces the model to choose only most important/impactful features.

As discussed above, Feature selection is a crucial step in machine learning that helps us choose the most important features from our data. By selecting the right featurs, we improve model performance, interpretability, and efficiency. It also allows us to handle high-dimensional data and focus on the most important attributes. As machine learning continues to advance, feature selection is a hot topic among researchers who are continually coming up with new and more efficient ways to select features.

Machine learning's impact is particularly significant in the field of drug discovery and development. Machine learning helps researchers understand the mechanism of action of drugs, or how they interact with the body to produce therapeutic effects, which in turn is crucial for effective treatment.

Traditionally, determining a drug's mechanism of action was a time-consuming and costly process. However, with the rise of machine learning, researchers can utilize vast amounts of data to discover hidden patterns and insights that would otherwise be chal-lenging if not impossible to identify.

Machine learning algorithms are trained on many huge datasets containing information about the chemical properties of drugs, their interactions with biological targets, and the resulting therapeutic effects. Machine learning algorithms produce predictions and hypotheses about a drug's mechanism of action after examining this data.

The use of machine learning in determining the mechanism of action of drugs offers several advantages:

Predictive Modeling: Machine learning algorithms can predict how a drug will interact with specific biological targets based on its chemical structure and other relevant factors. Which in turn helps researchers identify potential drug-target interactions and greatly reduces the number of further investigation.

Efficient Screening: It takes time and money to conduct traditional experimental screening of huge chemical libraries for prospective therapeutic candidates. Machine learn-ing algorithms analyze chemical structures and biological data to narrow down the most promising candidates, thus reducing the screening process and reducing costs.

Uncovering Complex Relationships: Machine learning models are inherently very good

at identifying complex relationships and patterns within the data. They can discover connections between chemical features and target interactions that otherwise might not be apparent through traditional methods. This deep insight enhances our knowledge of drug mechanisms and helps with the discovery of new therapeutic targets.

Personalized Medicine: Machine learning algorithms can be trained on large-scale pa-tient data, including genetic information, clinical records, and treatment outcomes. This way machine learning enables the identification of patient subgroups that may respond differently to specific drugs, facilitating the development of personalized treatment ap-proaches for those patient subgroups.

As machine learning algorithms continues to improve, its impact on determininig the mechanism of action of drugs and other fields will only grow stronger. The availability of data like genomic sequence and patient data and real-world evidence, will further improve the accuracy and applicability of machine learning models in drug discovery. With improve in machine learning we can expect important discoveries and more effective treatments that also improve the lives of countless individuals. This will also enable healthcare providers to make more educated decisions and deliver individualized treatment by giving them the ability to analyze huge amount of information, spot trends, and make precise predictions. We are moving towards a time when technology and human skill could bring in a new era of improved health and welbeing for all the people as we learn more about and put machine learning to use.

Chapter 2

LITERATURE REVIEW

2.1 Feature Selection

2.1.1 Introduction to Feature Selection

Feature selection is an important step in machine learning that involves choosing the most relevant and important features from a dataset. It improves the model performance by reducing complexity, increasing efficiency, and enhancing generalization all the while reducing the time needed to train the model.

In machine learning, datasets often contain numerous features, but not all of them are equally important for the model. Using irelevant and unimportant features can lead to overfitting, wasted computational resources, and lower model performance. Feature selec-tion addresses these challenges by identifying the most important and relevant features, resulting in improved model performance and faster training time.

As suggested by Ang et el., [8] there are multiple stages of feature selection:

- Search Direction: Ang et el. [8] State that, first stage for a FS process is finding the starting point and search direction. In the search process, there are three dif-ferent ways to find the best features. One way is called forward search, where new features are added step by step until the desired set is reached. Backward elimina-tion is another method in which features are deleted one by one from a complete set until the desired set is obtained. The third method is random search, which includes iteratively adding and removing features. After determining the search direction, a search strategy can be used to locate the best feature subset.

- Determine the search strategy: Based on what we have learned from previous studies, there are different types of search strategies: exponential, randomized, and sequential search. However, the exponential search has a drawback. It needs to consider 2^N combinations when selecting features from N available features. This approach is exhaustive and considered a chalenging problem known as NP-hard. As

2^N increases exponentialy with the number of features, randomized search strategies were proposed. In sequential search, we add features to an empty set one by one or delete features from a complete set one by one. When we add features, it's called Sequential Forward Selection (SFS), and when we remove features, it's called Sequental Backward Selection (SBS). However, a drawback of these methods is that once a feature is eliminated, it won't be considered for the remaining iterations. This drawback/phenomena is called nesting effect. Ferri and Pudil [9]presented the Plus-l-minus-r (l-r) search strategy to address this limitation. For picking l features from a set of M features, these approaches have a $\theta(2M)$ time complexity.

- Evaluation Criteria: We employ assessment methods to assess the efficiency of a given feature selection strategy. Wrapper techniques, Filter methods, Embedded methods, and Hybrid methods are the four categories of feature selection methods based on the assessment criterion [10].

- Stopping criteria: When the feature selection algorithm should stop? This is decided by the stopping criteria. Stopping criteria is important as it affects the computational complexity in overcoming overfitting and finding the best set of fea-tures. Some of the most common stopping features are:

  – Pre-defined Number of features

  – Pre-defined Number of iterations

  – Performance improvement as a percentage over two successive iteration stages

  – Based on the evaluation function

- Validate the results: To validate our results, we use validation techniques for the feature sets. Some of these techniques include confusion matrix, Jaccard similarity-based measure, cross validation, and Rand Index. The most commonly used method is Cross-Validation (CV), which provides an unbiased estimate of errors or an es-timate of the real world performance of the model. The Confusion Matrix helps evaluate how well the classifier is performing.

2.1.2 Types of Feature Selection

Based on evaluation criteria, we can divide feature selection methods into the following three types:

Filter Methods: These techniques recognizes the relevance and importance of features independently of the chosen learning algorithm. Correlation analysis, statistical tests, and information gain are some of the ways filter methods achieve their goal of finding relevant

features. Filter methods rank features based on their indivdual importance and relevance and select the top-ranked features for the model. Chi squared and LDA are some of the common filter type featuer selection methods. Rasim Cekik and Alper Kurset Uysal [11]proposed a novel filter based feature selection method called Proportional Rough Fea-ture Selector (PRFS) for selecting important features. PRFS uses rough set theory to distinguish documents that exactly belong to a class or might belong to a class based on the value set of a term. They penalized documents that might belong to a class by multiplying them with a coefficient called ”a”. Additionally, they considered the spar-sity in the term vector space using rough set theory. To evaluate PRFS, they compared it with other popular feature selection methods such as Gini index, information gain, distinguishing feature selector, max-min ratio, and normalized difference measure. The comparison was done using various feature sizes on four different datasets with short text. The performance was measured using the Macro-F1 score. Experimental results showed that PRFS performed either better or at least as well as other methods in terms of Macro-F1. Wrapper Methods: Wrapper methods evaluate subsets of features by using a specific learning algorithm. They create multiple models, each with a different subset of features, and then select the subset that gives the best performance. Wrapper methods are computationally expensive but usually produce quite accurate feature subsets. Recur-sive feature elimination and backward elimination are some of the commonly used feature selection algorithms. Babak et al. [12]proposed a novel wrapper feature selection method called “Forest Optimization Algorithm” (FOA). Their algorithm used grid, archive, and region-based selection to achieve multiple goals at once. They developed two versions of the algorithm, one using continuous representation and the other using binary represen-tation. They also tested the performance of our algorithms on nine UCI datasets and two microarray datasets. They compared the results with seven traditional single-objective methods and five other multi-objective methods. The findings showed that both their algorithms performed as well as or even better than the single-objective methods.

Embedded Methods: Embedded methods have the feature selection algorithm in the learning algorithm itself. Algorithms like “Lasso (Least Absolute Shrinkage and Se-lection Operator)” and “Ridge regression” use regularization (L1 and L2 regularization respectively) techniques to automatically select relevant and important features during the model training process itself. Lie et al. [13]introduced a FS method called the weighted Gini index (WGI). They compared the performance of WGI with other feature selection methods such as F-statistic, Chi2, and Gini index. The results show that when lesser number of features are selected, Chi-2 and F-statistic Chi2 perform the best. However, as the number of selected features increased, WGI achieved the best performance. They evaluate the performance using the area under the receiver operating characteristic curve (ROC AUC) and F-measure. The experimental results on two datasets showed that even

with a small number of selected features, the ROC AUC performance was high and only changes slightly when more features are selected. However, the F-measure achieves excellent performance only when 20% or more of the features are chosen.

We can also segregate feature selection methods on the presence or absence of the class labels. In absence of class label we can use unsupervised feature selection while in the presence of feature selection we can use unsupervised feature selection. If the dataset has both labelled and unlabelled data, we can use semi supervised feature selection.

Supervised feature selection: In this approach, we use the class label to choose important features. However, most of the times it leads to a problem called overfiting because of noisy data in the dataset. There are several popular supervised feature selec-tion methods such as the Fisher score, Fisher Criterion, Hilbert-Schmidt Independence Criterion (HSIC), trace ratio criterion, Pearson Correlation Coefficient, and mutual in-formation. Tutkan et al. [14]proposed Meaning Based Feature Selection (MBFS), a novel feature selection method. MBFS uses both supervised and unsupervised learning. MBFS is based on the Gestalt theorem of human perception and Helmholtz principle. This fea-ture selection was proposed for text mining and it uses the helmholtz principle for putting a meaning score to each word of the document.

Graph based unsupervised feature selection: In unsupervised F.S., we don't have the class labels. This makes it a more difficult task compared to semi-supervised and supervised feature selection. We use similarty measures to find and remove redundant features from the dataset. When features are similar to each other, we remove one of them. Likewise, if a feature doesn't contribute to clustering or findng patterns in the data, we eliminate it during the feature selection process. This is important for studying biological data and can help identify new types of diseases. However, there is one disadvantage to unsupervised feature selection: it does not take into account the association between different features. "Variance score", "Feature Selection using Feature Similarity" (FSFS), "Laplacian Score for Feature Selection" (LSFS), and others are well-known unsupervised feature selection algorithms.

S. Wang et al. [15]proposed a low rank approximation and structure learning based unsupervised feature selection. By using "low-rank approximation", they accurately determined the connected components in graphs when performing structure learning. As the first step, feature selection must be represented as a matrix factorization having low-rank constraints.This is done by using a self-representation of a data matrix. To capture the sparsity of the feature selection matrix, the l2,1-norm method is employed. With the help of structured learning and low-rank approximation techniques, an efficient algorithm has been developed. However, there is a limitation to this method, which is figuring out how to adaptively learn the feature subsets.

Y. Liu et al. [16]introduced a new method called "Diversity-Induced Self-Representation"

(DISR) for unsupervisd feature selection. They used the “Self-representation property” and an optimization algorithm called “Augmented Lagrange Method” (ALM) to make the process more eficient. By considering the diversity of features, which captures more information about the data, they were able to discard similar features. This led to a significant reduction in redundant features. The similarity between two features can be calculated using the dot product weight. A higher similarity value indicates that the features are more similar. In order to select the most valuable features, DISR takes into account diversity properties and representativeness.

Hu et al. [17]proposed a new method for unsupervised feature selection called “Graph Self Representation Sparse Feature Selection” (GSR-SFS). They increased feature se-lection stability by integrating a subspace-learning model (dubbed LPP) with a self-representation technique at the feature level. As a result, they developed a loss function that aids in the interpretation of the selected features. They also employed l2,1-norm regularisation to stabilise the learning process of subspace.

Du et al. [18]proposed “Robust Unsupervised Feature Selection through Matrix Fac-torization” as an unsupervised feature selection approach (RUFSM). They divided the data matrix into two matrices: one with latent cluster centres and one with sparse repre-sentation using the l2,1-norm. They were able to accomplish highly accurate discrimina-tive feature selection by predicting the orthogonal cluster centres.

Qi et al. [19]developed a novel technique known as “Regularized Matrix Factorization Feature Selection” (RMFFS). Using matrix factorization, this method finds the correlation between features. They consider the absolute values of the inner product of feature weight matrixto make the feature weight matrix sparse. For matrix factorization, they employ a combination of l1-norm and l2-norm.

Semi-supervised feature selection: The training set in Semi-Supervised feature selection contains both labelled and unlabeled data. Many scholars working on semi-supervised approaches are interested in the use of graph Laplacian methods in feature selection. For feature selection, a weighted graph is built using the available data. The graph depicts three levels of Semi Supervised Learning (GSSL). To begin, the user can select a kernel or similarity function to determine the affinity of a pair of samples.

Belkin et al. [20]used a Gaussian kernel model as a similarity function, and the findings demonstrate the method’s effectiveness. Second, the user must choose an algorithm for constructing the sparse weighted subgraph from the completely weighted graph between all nodes. The sparse subgraph is often created using KNN and $\epsilon - N$ eighbourhood. Finally, the user must utilise a graph-based semi-supervised learning technique to dis-tribute the class labels from known graph nodes to unknown data nodes. There are a variety of graph-based semi-supervised learning algorithms available, including the Gaus-sian fields, graph min-cut method, and harmonic methods, global and local consistency methods, manifold regularisation, and the alternating graph transduction method. Several

methods in “Graph-based Semi-Supervised Learning” (GSSL) make use of neighbourhood algorithms such as k-NN. Yet, research show that these methods frequently produce ir-regular and imbalanced graphs for actual and synthetic data. The problem stems from the greedy method of adding nodes to the graph based on the k nearest points. To solve this shortcoming, a method known as maximum weight b-matching was devised. Each node in the graph is precisely connected to b other nodes using this strategy.

2.2 Genetic Algorithm

2.2.1 Introduction to genetic algorithm

Genetic algorithms are computer models that copy the process of evolution. They use a gene-like structure to represent potential solutions to a problem. These algorithms apply recombination operations to the structures, preserving important information. Genetic algorithms are commonly used to optimize functions, but they can also be used in a wide range of problems. Figure 2.1 has a flowchart of genetic algorithm for reference.

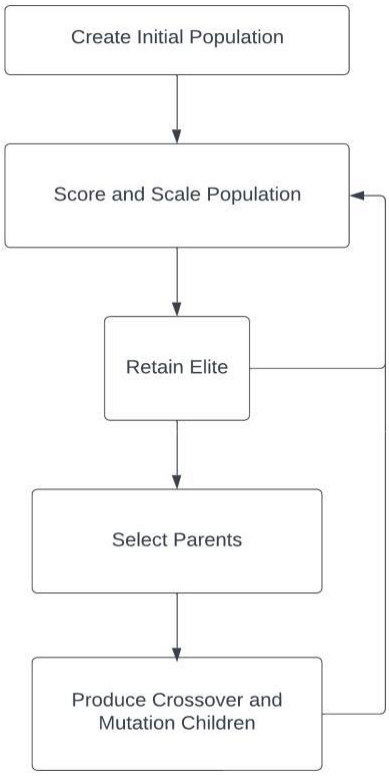

Figure 2.1: Genetic Algorithm, A Flow chart

To implement a genetic algorithm, we start with a set of genes, which are usually randomly created. We evaluate these genes and these genes are then provided reproductive opportunities based on their performance. Gene that represent better solutions have

higher chances to propagate/reproduce, while worse solutions have fewer opportunities. The quality of a gene is typically measured relative to the other genes in the current population.

In a more general sense, a genetic algorithm is a type of computer model that selects and then recombines (also called as corssover) operations to create new gene. Researchers have developed various genetic algorithm models, often focusing on practical applications and optimization. The specific way a genetic algorithm works can vary depending on the problem at hand, with two important components being the way the problem is represented (encoding) and how it is evaluated (evaluation function).

2.2.2 The Initial Population

To start a genetic algorithm, we first create a group of individuals called the initial population. Each individual is represented by a string of 1s and 0s, which is usually randomly generated and has a specific length determined by the problem. This string is like a unique identifier or DNA for the individual. After creating the initial population, we evaluate each individual's string and calculate a fitness score for them based on a fitness function.

2.2.3 Evaluation and Fitness

"Evaluation" and "fitness" it's important to understand their distinct meanings in the context of a genetic algorithm. In our case, the fitness function, also known as the objec-tive function, measures how well a set of parameters performs (We check how accurate the model is and find the best accuracy among all the models). The fitness function then takes the evaluation results and determines the reproductive opportunities for each individual. Evaluating a gene which is the representation of the parameters is done independently, without considering other strings. However, the fitness of that gene is always relative to the other individuals in the current population.

2.2.4 Generating Off-Spring

Mainly Genetic Algorithm is composed of two steps. First, we have the current popu-lation. We create an intermediate population by applying selection on the current pop-ulation. Then, we recombine and mutate the intermediate population to generate next population. This process in which we created a new population from current population is one generation in genetic algorithm. We will focus on the simple genetic algorithm implementation.

We begin with the initial population in the first generation. Then the fitness of each gene in the current population is calulated. Once that is complete we perform selection. In

the genetic algorithm, the probability of copying and placing strings into the intermediate population is based on their fitness level.

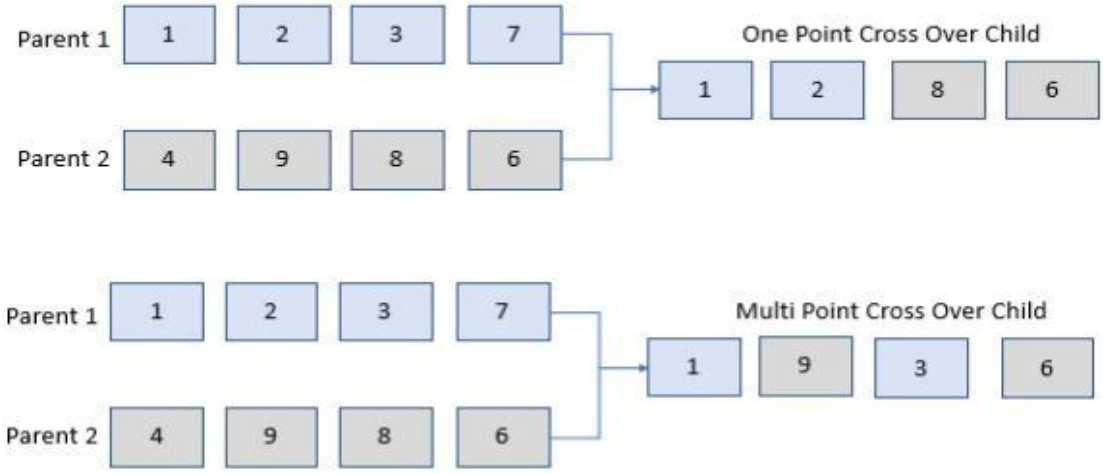

Figure 2.2: Crossover in genetic algorithm

Once the selection is completed, we move on to recombination. We create the next population using the intermediate population. Randomly chosen genes undergo crossover with probability proportional to their fitness value. (The is already shuffled sufficiently through the selection process which is random.) Figure 2.2 has represents crossover of two parents. For each pair, there is a chance that the genes will recombine, resulting in two new genes that are added to the next population.

We introduce mutation after recombination. Each bit in the gene has a very low probability of being mutated. Generally, the mutation rate is less than 1%. Depending on the interpretation, mutation can involve randomly generating a new bit, in which case there's a 50% chance of the bit value changing. In other cases, we can flip the bit. Both implementations are correct as long as the reader understands the difference and recognizes that one form of mutation results in bit change only half as often as the other.

Selecting the best features for a task is a difficult problem (NP hard problem as the best way isto try every possible set of the features). It involves finding the ideal combination of features from a given set, and this process is quite complex and time-consuming.

Genetic algorithms provide a solution by mimicking the process of evolution. In feature selection, the first step is to create a group of potential feature combinations called a population.

Each combination in the population is evaluated using a predictive model for the spe-cific task at hand. After evaluating all the combinations, we use the fitness score (provided by the fitness function) is conducted to determine which combinations will move on to the next generation. The next generation consists of the winners from the tournament, with some features being exchanged between the winning combinations and random changes.

Following are the steps performed:

- Create an initial population of feature combinations.
- Assign a score to each combination based on its performance.
- Conduct a tournament to select combinations for reproduction.
- Choose genetic material to pass on to the next generation.
- Introduce random changes (mutations).
- Repeat these steps for multiple generations.

After a set number of generations, the best combination of features is selected as the optimal solution. By following this iterative process, the genetic algorithm finds the best set of features for the task at hand.

Tahir et al. [21] introduced a new optimization method called the “Binary Chaotic Genetic Algorithm (BCGA)” to improve the performance of the traditional genetic algo-rithm (GA). They used Chaotic maps to modify the initial population and reproduction operations in the algorithm. Then they applied BCGA to a feature selection task using three datasets: AMIGOS and two healthcare datasets with many features. Then they compared BCGA’s performance with the traditional GA and two other FS methods. The comparison was based on number of selected features and classification accuracy. The results showed that the BCGA found the best subset of features with improved fitness values.

Jiang et al. [22] proposed a new approach known as modified genetic algorithm (MGA), which is aimed to choose the most beneficial characteristics by taking into account their combinations and the information offered by filter-based feature selection. The forecasting model was a deep neural network, and its initial parameters were produced using stacked autoencoders during a pre-training procedure. The methodology was tested on an OPD in Northeast China and compared to other FS methods and demand forecasting models. Their results showed that MGA improved the performance and efficiency of feature selec-tion compared to other methods, leading to higher forecast accuracy with fewer selected features. The deep neural network enhanced the advantages of MGA, outperforming other forecasting models.

Rostami et al. [23] proposed a new method for selecting features using community detection based on genetic algorithm. The method involves three steps. First, the simi-larities between features are calculated. Then, community detection algorithms are used to group similar features into clusters. Lastly, the features are chosen using a genetic

algorithm with a community-based repair operation. The authors evaluated the perfor-mance of their approach on nine benchmark classification problems and compared it to four other feature selection algorithms. The results revealed that the ethod outperformed three alternative methods based on the PSO, ACO, and ABC algorithms.

Xue et al. proposed introduced a new method called MOBGA-AOS, which is a genetic algorithm for feature selection. It uses five different crossover operators that have different search behaviors. The algorithm dynamically selects the best crossover operator based on its performance during the evolution process. This selection is done using a roulette wheel method that considers the probabilities assigned to each operator. The authors compared MOBGA-AOS with five other well-known algorithms and tested on ten datasets. The re-sults showed that MOBGA-AOS is effective in removing many features while maintaining a low classification error. It also performed well on large datasets, indicating its ability to handle high-dimensional feature selection problems.

2.3 Cross-Validation

2.3.1 Introduction to Cross-Validation

In machine learning, it's important to make sure that our model performs well not just on the data it was trained on, but also on new, real-world data. To do this, we use a technique called cross-validation.

Cross-validation allows us to measure how well our model will work on unseen data. We divide our available data into several parts, or folds. One fold is used to test the model, while the remaining folds are used to train it. We repeat this process multiple times, using a different fold as the validation set each time. Finally, we calculate the average performance of the model across all the folds.

Cross-primary validation's purpose is to prevent overfitting. Overfitting occurs when a model gets overly specialised in the training data and fails to perform well on fresh, previously unknown data. Cross-validation provides a more accurate approximation of how the model will perform on fresh data by assessing it on numerous validation sets that have a balanced representation of the target classes.

Cross-validation approaches include leave-one-out cross-validation, k-fold cross-validation, and stratified cross-validation. The technique used is determined by elements like as the size and nature of the data, as well as the unique requirements of the modelling task. In conclusion, cross-validation is a critical stage in machine learning since it ensures that the model chosen is robust and capable of performing well on new data. Cross validation flow chart is given in figure 2.3.

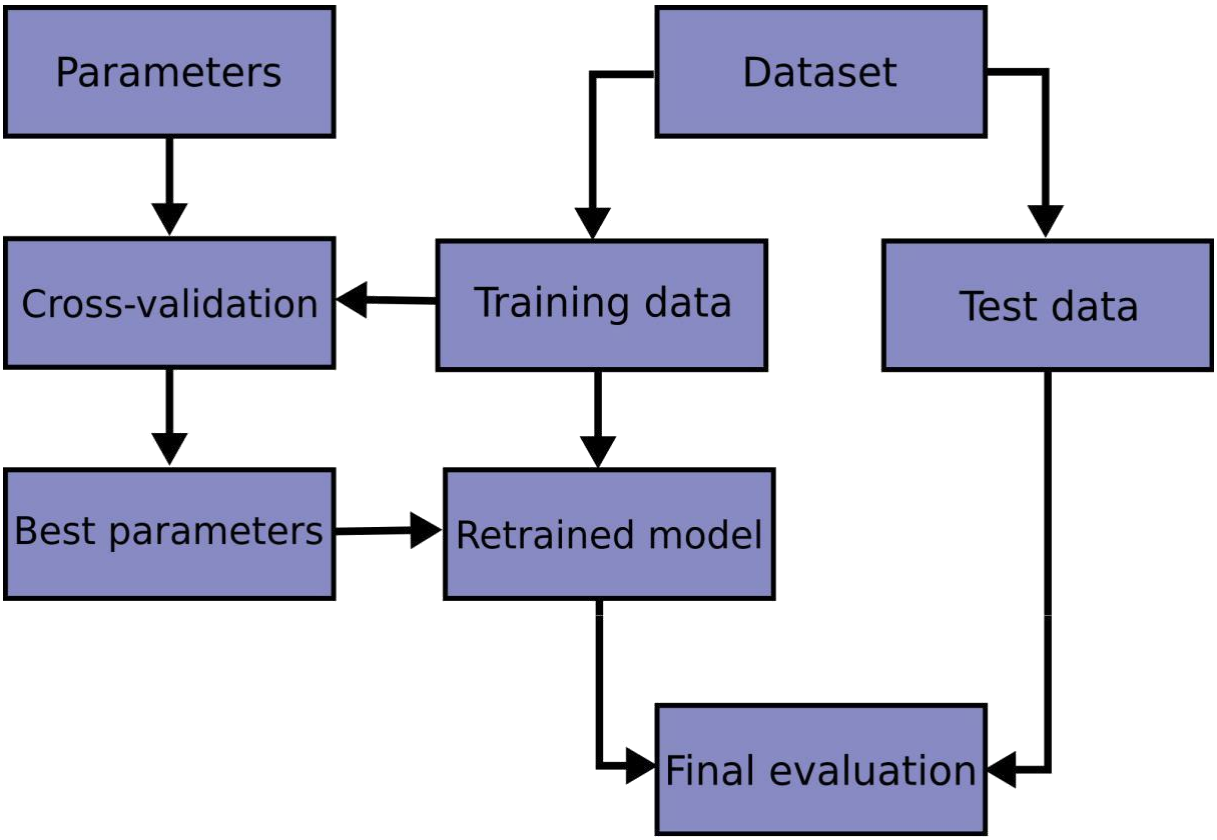

Figure 2.3: Cross Validation Flow

2.3.2 Cross-Validation Methods

Below are the types of cross validation [24]

Validation: In this method, we split the dataset into two parts: one for testing (50%) and the other for training (50%). However, a disadvantage of this approach is that we only train our model on half of the available data. This means that we might be missing out on important information contained in the other half, leading to higher bias in our model. It's important to note that this method often results in an underfitted model, which may not capture the complexity of the data accurately.

LOOCV (Leave One Out Cross Validation): In this method, we train our model using the entire dataset but leave out one data point at a time during each iteration. This approach has both advantages and disadvantages. One advantage is that we utilize all the available data points, making it less biased. However, a major drawback is that it can result in higher variation in the testing process because the testing is only done against a single data point. If that data point is an abberation, it can greatly affect the results. Additionally, this method takes longer to execute as it requires iterating over each data point individually.

K-Fold Cross Validation: In this method, the data is divided into K smaller groups called folds. The model is trained using the k-1 folds, and we save one fold for testing.

This process is repeated multiple times, and each time we use a different fold for testing. Note that as there are k folds and we are using each fold once for testing, we iterate k times. The following diagram (Figure 2.4) depicts K-Fold cross validation with K=5.

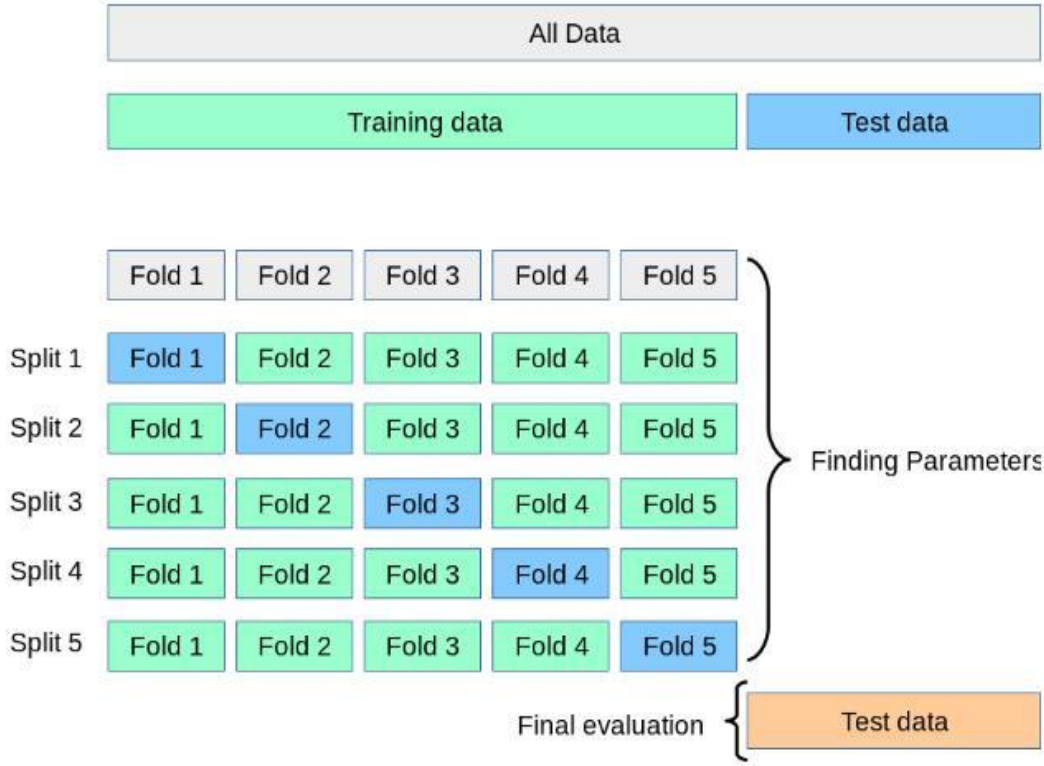

Figure 2.4: K-fold cross validation illustration, here K=5

Stratified K-Fold Cross Validation: Stratified k-fold cross-validation is an en-hanced version of regular “K-fold cross-validation”, designed for classification tasks. In regular k-fold cross-validation, the data splits are random. However, in ”Stratified K-fold cross-validation”, the distribution of target classes in each fold is maintained to be the same as the overall dataset.

To understand this, consider an example where we want to classify images of bike and car. Suppose our dataset contains 75% bike images and 25% car images. By using stratified k-fold cross-validation, we ensure that each fold we create will have a similar proportion of bike and car images, closely reflecting the 75/25 ratio of the full dataset.

2.4 Mechanism of Action

2.4.1 Introduction to Mechanism of Action

The mechanism of action refers to the specific way in which a drug interacts with our bodies to produce its desired effects. Drugs have important targets in our bodies, such as

receptors, enzymes, or signaling pathways. By binding to these targets, drugs can either activate or inhibit them, leading to various therapeutic outcomes. Knowing the MoA helps us understand how a drug works, predict its effects, and develop new drugs with specific targets.

Machine learning, has become a valuable tool for scientists to uncover the MoA of drugs. It helps them make sense of complex data and make predictions all the while reducing there efforts. Following are some ways machine learning is helping researchers in discovering/analyzing the MOA of drugs:

- Analyzing Diverse Data: Machine learning algorithms handle large amounts of biological and chemical data. Scientists input different types of information, like genetic data, chemical structures, and protein interactions. By analyzing these vast datasets, patterns and relationships can be discovered, providing clues about a drug's MoA.

- Making Predictions: Machine learning algorithms create models by learning from existing data about drugs with known MoAs. These models can then be used to predict the MoA of new drugs based on their chemical properties, genetic informa-tion, or other relevant factors. This helps speed up the drug discovery process by narrowing down potential targets for further study.

- Network Pharmacology: Network pharmacology is an approach that combines machine learning and biological networks. By studying the connections between genes, proteins, and drugs, scientists can gain insights into drug MoAs. Machine learning help analyze and interpret these complex networks, uncovering hidden re-lationships that is crucial for understanding drug actions.

Using machine learning to determine drug MoAs accelerates the drug discovery pro-cess, reduces costs, and improves treatment outcomes. By understanding how drugs work, researchers can also identify potential side effects and drug interactions, ensuring safer and more effective treatments.

Gururaj et al. [25] proposed a machine learning model to predict the mechanism of action of a drug. They used "Multi-label K-Nearest Neighbors", "Binary Relevance K Nearest Neighbors (Type A and Type B)", and a custom neural network. Then they evaluated these models on mean column-wise log loss. In their findings they found out that custom neural network had the best performance with a log loss of 0.01706. They also created a web app where a user can upload a dataset and then the web app will provide the top classes of drugs.

Scott J. Warchal et al. [26] in their paper compared ensemble based tree classifier with a deep learning classifier based on CNN for predicting compound MOA when transferred across cell lines. They found that CNN performed equally well as the ensemble based.

tree classifier when predicting within cell lines. Whereas CNN performed worse than the ensemble model when trained on multiple cell lines at predicting the compound MOA.

Chapter 3

METHODOLOGY

In this chapter, we will be discussing my methodology. We'll be discussing the dataset I used, the process I followed and other theoretical aspects i used in my project.

3.1 Comparing Genetic algorithm based feature selection against other common feature selection algorithms

3.1.1 Dataset used

I used the breast cancer dataset from sklearn. The dataset is also known as the breast cancer Wiconsin dataset. The dataset has 569 samples and two classes denoted by M and B (Malignant and Benign respectively). Where M represents the cancerous samples and B represents the non cancerous samples. Each sample has 32 attributes ranging from the id, diagnosis (target class), radius, texture, perimeter, area, smoothness, etc. The dataset is given in figure 3.1.

3.1.2 Data Pre-Processing

We are standardizing the data using StandardScaler(). We are standardising the data as different columns/features are measured at different scales and they might not contribute equally to the model training. This can create bias in our model and to solve that problem we standardise our model such with mean = 0 and standard deviation 1.

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.07871	...
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	...
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.05999	...
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.09744	...
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.05883	...

5 rows × 30 columns

...	worst radius	worst texture	worst perimeter	worst area	worst smoothness	worst compactness	worst concavity	worst concave points	worst symmetry	worst fractal dimension
...	25.38	17.33	184.60	2019.0	0.1622	0.6656	0.7119	0.2654	0.4601	0.11890
...	24.99	23.41	158.80	1956.0	0.1238	0.1866	0.2416	0.1860	0.2750	0.08902
...	23.57	25.53	152.50	1709.0	0.1444	0.4245	0.4504	0.2430	0.3613	0.08758
...	14.91	26.50	98.87	567.7	0.2098	0.8663	0.6869	0.2575	0.6638	0.17300
...	22.54	16.67	152.20	1575.0	0.1374	0.2050	0.4000	0.1625	0.2364	0.07678

Figure 3.1: The dataset used. The dataset has 32 columns.

3.1.3 Classifier used

Logistic regression: Logistic regression is a statistical method used to predict the like-lihood of an event happening based on input variables. It is commonly used in situations where the outcome we want to predict is binary, meaning it can only have two possible outcomes (e.g., yes or no, true or false).

In logistic regression, we start with a set of input variables, such as age, gender, and income, that we believe might influence the outcome. We assign weights to each input variable based on their importance in predicting the outcome. These weights help us calculate a probability score for each individual, indicating the likelihood of the event occurring.

To make predictions, logistic regression uses a mathematical function called the logistic function (also known as the sigmoid function, depicted in figure 3.2) to map the probability score to a value between 0 and 1. If the probability score is above a certain threshold (usually 0.5), we predict that the event will occur. If it is below the threshold, we predict that it won't occur.

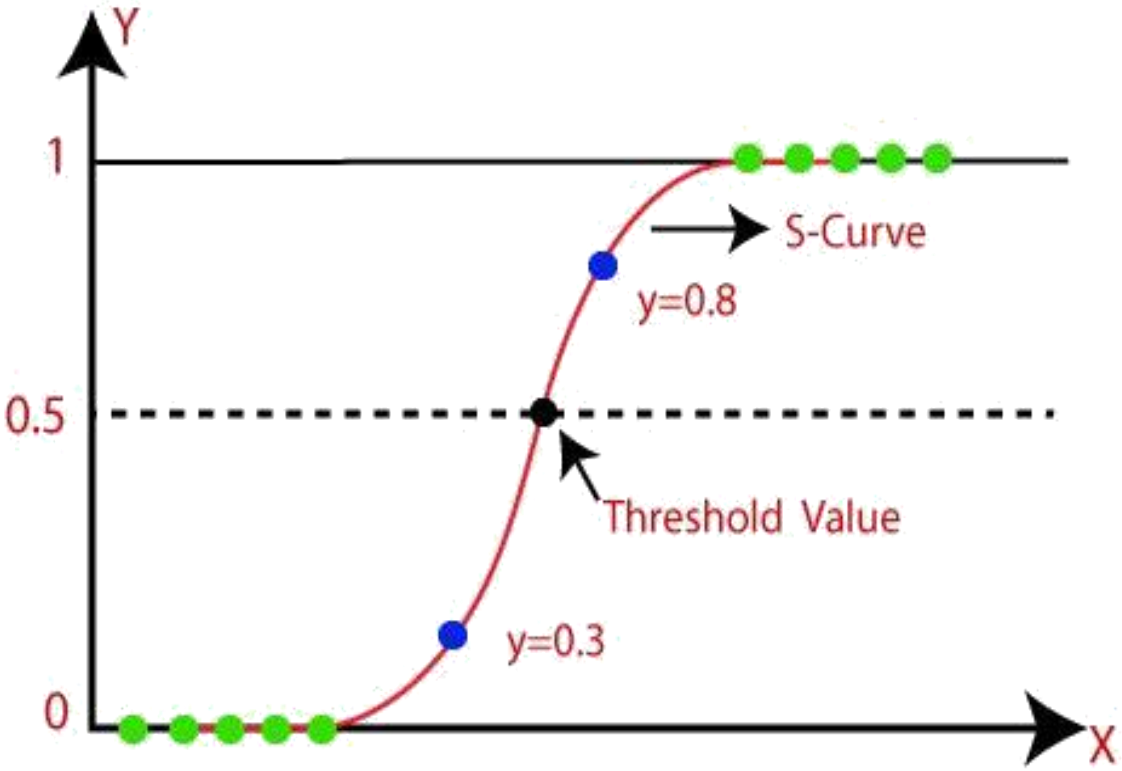

Figure 3.2: The sigmoid curve of logistic regression.

The logistic regression model is trained using a labeleld dataset. It adjusts the weights assigned to the input variables during training to find the best fit that maximizes the accuracy of the predictions. The process of finding the optimal weights is typically done using an algorithm called maximum likelihood estimation.

3.1.4 Feature selection methods used

Genetic algorithm: We create a population of gene where the gene represents the weight of each feature. In each generation of multiple genes, each gene is evaluated on a fitness function and provided with a fitnes score. The gene's probability to be propagated to the next generation is directly proportional to it's fitness score. This way it is ensured that the most fit genes have the highest chances of propagating to the next generation. Note that the offspring is generated after combining two genes and then applying mutation to the gene.

Recursive feature elimination: Recursive Feature Elimination (RFE) is a tech-nique used to select the most important features from a set of variables in a step-by-step manner.

Following are the steps used in RFE:

- Start with all the features: Initially, RFE considers all the variables or features available for selection.
- Build a model and evaluate feature importance: RFE builds a model using the selected features and evaluates the importance of each feature based on a predefined criterion. This criterion could be the coefficient values in a linear regression model or the feature importance scores in a decision tree model.
- Eliminate the least important feature(s): RFE identifies the least important fea-ture(s) based on the evaluation in the previous step. These features are removed from the set of variables.
- Repeat the process: Steps 2 and 3 are repeated iteratively until a specified number of features remains or a desired level of accuracy is achieved.

By eliminating the least important feature(s) at each iteration, RFE aims to find the subset of features that are most relevant and informative for the problem at hand. RFE is a recursive process because it repeatedly builds and evaluates models with different subsets of features. It prioritizes the features based on their importance and eliminates the ones that contribute less to the model's performance.

Feature importance using Random forest regressor: Random forest regressor calculates feature importance for the dataset features and the same can be accessed by using "randomForest.feature importances ". Random forest regressor calculates these importances as the decrease in node impurity weighted with the probability of reaching that node. The probability of reaching that node is calculated by dividing the number of samples reaching the particular node with the total number of samples.

Dimensionality reduction using factor analysis: Dimensionality reduction using factor analysis is a statistical technique used to simplify a large set of variables into a

smaller set of underlying factors. It aims to uncover the latent factors that explain the correlations among the observed variables.

Here's how factor analysis works:

- Data correlation: The first step is to examine the correlations between the variables in the dataset. If there are high correlations among variables, it suggests that they share common underlying factors.

- Factor extraction: Factor analysis tries to extract a smaller number of factors that explain the maximum amount of variance in the dataset. It identifies linear combinations of the original variables that represent these underlying factors. The extraction process is based on statistical methods such as principal component analysis (PCA) or maximum likelihood estimation.

- Factor rotation: Once the initial factors are extracted, factor rotation is performed to improve interpretability. Rotation aims to obtain simpler and more meaningful factor structure by minimizing the number of variables that load heavily on each factor. Common rotation methods include varimax, oblimin, and quartimin.

- Factor loading interpretation: After rotation, each variable is assigned a factor loading, which indicates the strength and direction of its relationship with each factor. Variables with higher factor loadings on a specific factor are considered more strongly associated with that factor.

- Factor selection: Researchers typically choose factors based on their interpretabil-ity and relevance to the research question. Factors with eigenvalues greater than 1 (Kaiser's criterion) or a scree plot showing a steep drop in eigenvalues can be considered for further analysis.

- Dimensionality reduction: Once the factors are identified, the original variables can be replaced by the factor scores. These scores represent the contribution of each factor to each observation in the dataset. By using the factor scores, the dimensionality of the dataset is reduced as the original variables are now represented by a smaller number of factors.

Dimensionality reduction using Principal Component Analysis (PCA):Dimensionality reduction using Principal Component Analysis (PCA) is a widely used technique to sim-

plify high-dimensional data by transforming it into a lower-dimensional space. PCA aims to capture the most important patterns and variability in the data while minimizing in-formation loss.

Here's how PCA works:

- Data preprocessing: The first step is to standardize the data by subtracting the mean and dividing by the standard deviation of each variable. This ensures that all variables have the same scale and prevents dominance by variables with larger magnitudes.

- Covariance matrix calculation: PCA computes the covariance matrix, which rep-resents the relationships between pairs of variables in the dataset. The covariance indicates how changes in one variable are associated with changes in another. It helps identify the directions of maximum variability in the data.

- Eigendecomposition: The next step is to perform eigendecomposition on the covari-ance matrix. This process calculates the eigenvectors and eigenvalues. Eigenvectors represent the principal components (PCs), and eigenvalues quantify the amount of variance explained by each PC. The eigenvectors are orthogonal to each other, meaning they are uncorrelated and capture independent directions of variability in the data.

- PC selection: The PCs are ranked based on their corresponding eigenvalues, with higher eigenvalues indicating greater variance explained. Typically, the top-ranked PCs are selected for further analysis, as they capture the most significant patterns and explain a large portion of the total variance. Researchers often set a threshold, such as keeping PCs that explain a certain percentage of the total variance (e.g., 90

- Dimensionality reduction: The original high-dimensional data can now be projected onto the selected PCs to obtain lower-dimensional representations. This is achieved by multiplying the standardized data by the selected eigenvectors (PCs). The re-sulting transformed data has fewer dimensions equal to the number of selected PCs.

- Interpretation and analysis: The transformed data can be used for subsequent anal-ysis, visualization, or modeling. The reduced-dimensional representation provides a compressed view of the original data, capturing the most salient features while discarding less important variations. It simplifies data interpretation, visualization, and can potentially improve the performance of downstream tasks like clustering, classification, or regression.

Dimensionality reduction using isometric mapping: Dimensionality reduction using Isometric Mapping (Isomap) is a technique that aims to preserve the geometric relationships between data points in a high-dimensional space while projecting them into a lower-dimensional space. It focuses on maintaining the local similarities and distances between points, making it useful for non-linear data structures.

Here's an explanation of how Isomap works:

- Constructing a neighborhood graph: Isomap starts by defining the neighborhood relationships between data points. It connects each point to its nearest neighbors based on a chosen metric, such as Euclidean distance. The number of neighbors to consider is a parameter that needs to be determined.

- Calculating geodesic distances: Next, Isomap computes the geodesic distances be-tween all pairs of points in the neighborhood graph. Geodesic distance refers to the shortest path distance along the graph, considering the connections between points. It takes into account the neighborhood structure and can capture non-linear rela-tionships.

- Constructing the low-dimensional embedding: Isomap applies classical multidimen-sional scaling (MDS) to embed the data points into a lower-dimensional space. MDS finds a configuration of points in a lower-dimensional space that best approximates the pairwise geodesic distances obtained in the previous step. The number of di-mensions in the embedding is a user-defined parameter.

- Visualization and analysis: The resulting low-dimensional embedding can be visu-alized and analyzed to understand the structure and relationships within the data. It provides a compressed representation of the original high-dimensional data while preserving the important global and local similarities.

Isomap offers a dimensionality reduction by incorporating the intrinsic geometry of the data. By considering the neighborhood relationships and geodesic distances, it can capture non-linear structures and preserve the underlying data relationships better than linear techniques like PCA.

3.2 Determining Mechanism of action of drug

I used the Mechanism of action dataset from kaggle. The dataset has combined the gene expression and cell viability data. The dataset has a total of 27796 samples and it is split into two parts, training set (23814 rows) and testing set (3982 rows). The dataset has 876 attributes. The dataset is given in figure 3.3.

sig_id	cp_type	cp_time	cp_dose	g-0	g-1	g-2
id_000644bb2	trt_cp	24	D1	1.0620	0.5577	-0.2479
id_000779bfc	trt_cp	72	D1	0.0743	0.4087	0.2991
id_000a6266a	trt_cp	48	D1	0.6280	0.5817	1.5540
id_0015fd391	trt_cp	48	D1	-0.5138	-0.2491	-0.2656
id_001626bd3	trt_cp	72	D2	-0.3254	-0.4009	0.9700
id_001762a82	trt_cp	24	D1	-0.6111	0.2941	-0.9901
id_001bd861f	trt_cp	24	D2	2.0440	1.7000	-1.5390
id_0020d0484	trt_cp	48	D1	0.2711	0.5133	-0.1327
id_00224bf20	trt_cp	48	D1	-0.3014	0.5545	-0.2576
id_0023f063e	trt_cp	48	D2	-0.0630	0.2564	-0.5279
id_002452c7e	trt_cp	72	D2	-0.2875	0.0322	-0.8863
id_0024bcd70	trt_cp	48	D2	-0.3864	-0.5551	-0.8978
id_0025c5949	trt_cp	48	D1	0.0030	0.7189	1.8890

Figure 3.3: The drug mechanism of action dataset.

3.2.1 Data Pre-Processing

I’m using Stratified K-fold Cross validation. Therefore the data is divided into K (which is 5 in my case) folds.

3.2.2 Feature selection method used

Genetic Algorithm was used to assign weights to the feature. Each gene/individual had the encoding for all the features of the dataset and had the weight as the value for that particular feature. The weights varied from 0 to 1. The algorithm ran for 20 generations and KNN was used for calculating the fitness of the genes. Figure 3.4 represents the weight assigned to different features in a gene.

	cv	0	1	2	3	4	5	6	7	8	...
97	0.018899	100.0	6.0	2.5	0.602275	0.414047	0.573935	0.229658	0.628431	0.000215	...
80	0.018901	100.0	6.0	2.5	0.331359	0.414047	0.573935	0.229658	0.628431	0.240256	...
2	0.018902	100.0	6.0	2.5	0.300750	0.414047	0.573935	0.229658	0.628431	0.125681	...
50	0.018905	100.0	6.0	2.5	0.300750	0.414047	0.012844	0.502364	0.628431	0.537829	...
22	0.018905	100.0	6.0	2.5	0.000512	0.414047	0.573935	0.229658	0.183366	0.125681	...

...	865	866	867	868	869	870	871	872	873	874
...	0.001794	0.393178	0.002074	0.980315	0.102086	0.002192	0.093255	0.918856	0.982162	0.879228
...	0.022740	0.393178	0.016678	0.944321	0.102086	0.002192	0.730433	0.918856	0.982162	0.455154
...	0.022740	0.004025	0.016678	0.832634	0.102086	0.002192	0.730433	0.918856	0.982162	0.391908
...	0.258061	0.104964	0.016678	0.980315	0.102086	0.002192	0.730433	0.381929	0.982162	0.391908
...	0.707034	0.393178	0.016678	0.980315	0.102086	0.002192	0.730433	0.918856	0.982162	0.391908

Figure 3.4: The drug mechanism of action gene.

3.2.3 Classifier used

KNN: KNN (K Nearest Neighbours) is a supervised learning classifier. It can be used as both classifier as well as regression method. Though note that it is mostly used as a classifier. When classifying a datapoint, KNN considers the K nearest neighbours of the datapoint and assigns the majority class to the datapoint. Note that KNN is a lazy learning method, meaning that it performs most of the calculation at the time of assigning class to the dataset. Also note that KNN is highly sensitive to dimensionality.

Chapter 4

RESULTS and DISCUSSION

4.1 Comparing Genetic algorithm based feature se-lection against other common feature selection algorithms

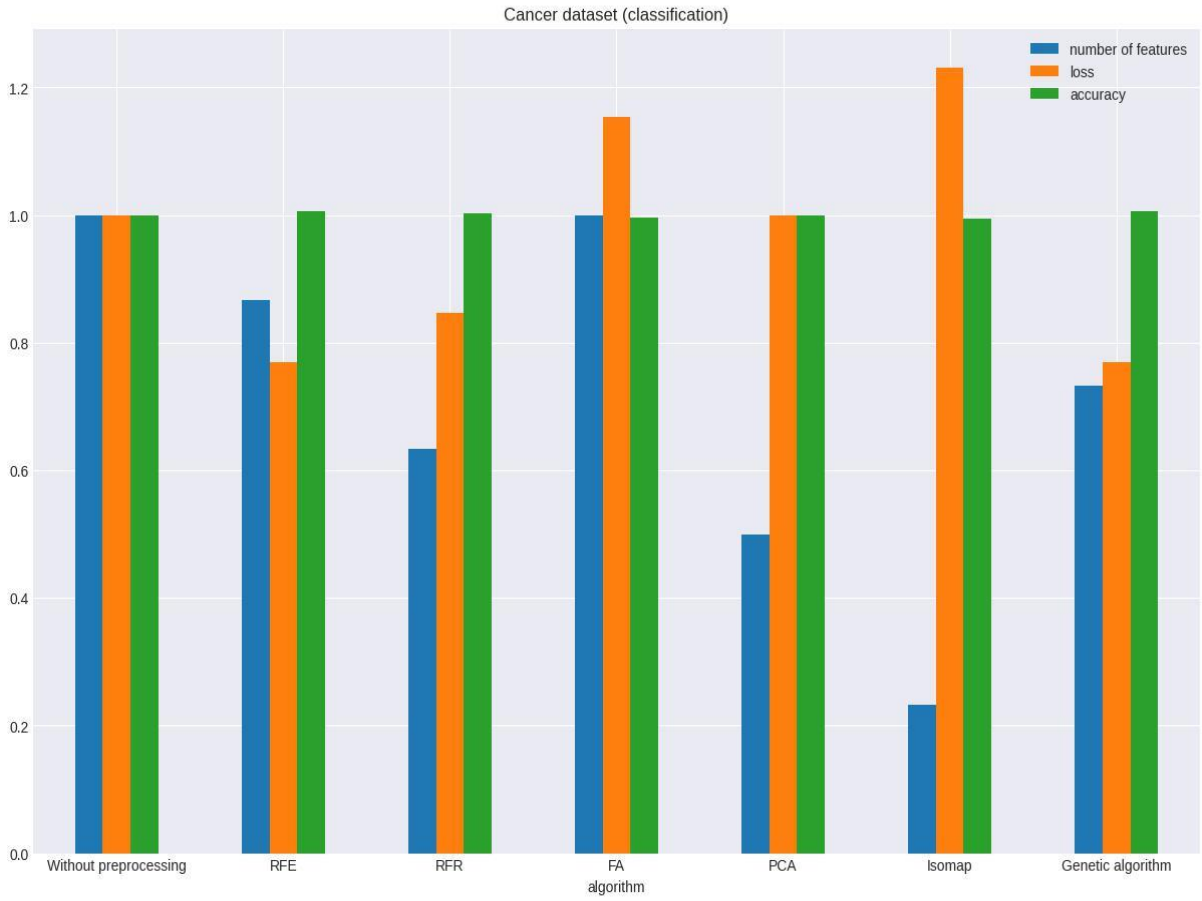

Figure 4.1: Comparision graph of all the feature selection methods considered.

Feature Selection method	Number of features	Error	Accuracy
No Feature Selection	30	0.023	0.977
Recursive Feature Elimination	26	0.018	0.982
Random Forest Regressor	19	0.019	0.981
Factor Analysis	30	0.026	0.974
Principal Component Analysis	15	0.023	0.977
Isometric Mapping	7	0.028	0.972
Genetic Algorithm	22	0.018	0.982

Table 4.1: Table with the number of features, error and accuracy of all the feature selection methods considered.

We have compared multiple commonly used feature selection algorithms with genetic algorithm. Although the accuracy of the Logistic regression model was comparable for all the feature selection algorithms. The loss was minimum when compared to all the other feature selection models. Do note that Principal component analysis, Isomap and Random Forest regressor had lesser features when compared to the genetic algorithm. The following comparison graph summarizes the results obtained. figure 4.1 represents the graph comparing the accuracy, error and loss for all the feature selection methods considered. Table 4.1 has all the values for accuracy, error and loss for all the methods considered.

4.2 Determining Mechanism of action of drug

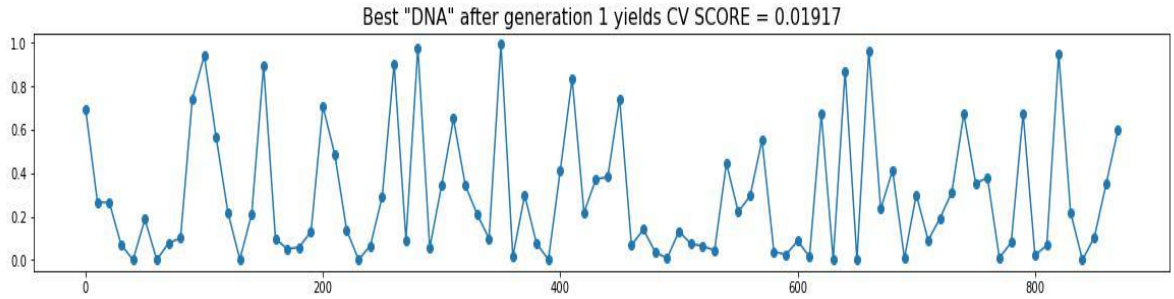

Figure 4.2: The CV score and the weights of the first generation's best DNA.

We are using genetic algorithm for feature selection. The first generation had a CV score of 0.01916. This score improved to 0.01890 in the twentieth generation. Figure 4.2 and figure 4.3 show the respective genes for generation 1 and generation 20. Another observation that can be made from considering multiple best DNA's is that a group of features have a high weight assigned to them among multiple generation's best DNA.

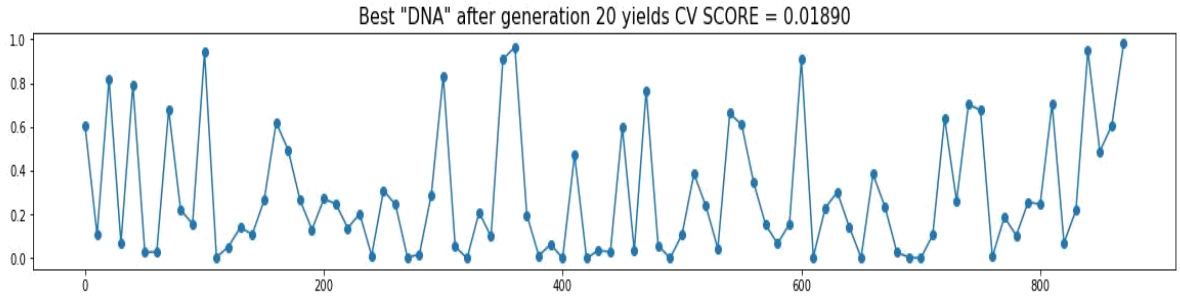

Figure 4.3: The CV score and the weights of the twentieth generation's best DNA.

(This can also be seen from the plots of best DNA's of first and twentieth generation, both have assigend high weight to last few attributes).

Chapter 5

CONCLUSION AND FUTURE SCOPE

We compared the performance of genetic algorithm based feature selection with Random forest regressor feature importance, Principal Component analysis, Recursive Feature Elimination, Factor Analysis and Isometric Mapping. We found out that genetic algo-rithm performed comparably in accuracy but had better performance (when we keep both loss and number of feature in mind) that the other feature selection methods considered. This goes on to show that after availability of many new feature selection methods, ge-netic algorithm based feature selection still remains relevant as a feature selection method.

Then we put our above findings to work and we used genetic algorithm for feature se-lection in Method of Action estimation. We saw that the CV Score/performance improved substantially from the first generation to the twentieth generation.

While this thesis has shed light on the effectiveness of genetic algorithm as a feature selection method, still a lot of opportunities exist for improvement. Future studies may focus on many issues, including:

- A bigger study with more feature selection methods and more classifiers should be conducted and the affect of selecting a type of feature selection method on a particular should be studied.

- Genetic algorithm combined with another technique (tweaked genetic algorithm) should also be studied.

References

[1] B. Smith and G. Linden, “Two decades of recommender systems at amazon. com,” Ieee internet computing, vol. 21, no. 3, pp. 12–18, 2017.

[2] C. A. Gomez-Uribe and N. Hunt, “The netflix recommender system: Algorithms, business value, and innovation,” ACM Transactions on Management Information Systems (TMIS), vol. 6, no. 4, pp. 1–19, 2015.

[3] Z. Wu, L. Li, R. Jin, L. Liang, Z. Hu, L. Tao, Y. Han, W. Feng, D. Zhou, W. Li et al., “Texture feature-based machine learning classifier could assist in the diagnosis of covid-19,” European journal of radiology, vol. 137, p. 109602, 2021.

[4] M. Reed, T. Le Sou¨ef, and E. Rampono, “Pilot study of a machine-learning tool to assist in the diagnosis of hand arthritis,” Internal Medicine Journal, vol. 52, no. 6, pp. 959–967, 2022.

[5] J. Lu, E. Song, A. Ghoneim, and M. Alrashoud, “Machine learning for assisting cervi-cal cancer diagnosis: An ensemble approach,” Future Generation Computer Systems, vol. 106, pp. 199–205, 2020.

[6] M. J. Iqbal, Z. Javed, H. Sadia, I. A. Qureshi, A. Irshad, R. Ahmed, K. Malik, S. Raza, A. Abbas, R. Pezzani et al., “Clinical applications of artificial intelligence and machine learning in cancer diagnosis: looking into the future,” Cancer cell in-ternational, vol. 21, no. 1, pp. 1–11, 2021.

[7] C. R´eda, E. Kaufmann, and A. Delahaye-Duriez, “Machine learning applications in drug development,” Computational and structural biotechnology journal, vol. 18, pp. 241–252, 2020.

[8] J. C. Ang, A. Mirzal, H. Haron, and H. N. A. Hamed, “Supervised, unsupervised, and semi-supervised feature selection: a review on gene selection,” IEEE/ACM transac-tions on computational biology and bioinformatics, vol. 13, no. 5, pp. 971–989, 2015.

[9] F. J. Ferri, P. Pudil, M. Hatef, and J. Kittler, “Comparative study of techniques for large-scale feature selection,” in Machine intelligence and pattern recognition. Elsevier, 1994, vol. 16, pp. 403–413.

[10] M. Dash and H. Liu, “Feature selection for classification,” Intelligent data analysis, vol. 1, no. 1-4, pp. 131–156, 1997.

[11] R. Cekik and A. K. Uysal, “A novel filter feature selection method using rough set for short text data,” Expert Systems with Applications, vol. 160, p. 113691, 2020.

[12] B. Nouri-Moghaddam, M. Ghazanfari, and M. Fathian, “A novel multi-objective forest optimization algorithm for wrapper feature selection,” Expert Systems with Applications, vol. 175, p. 114737, 2021.

[13] H. Liu, M. Zhou, and Q. Liu, “An embedded feature selection method for imbalanced data classification,” IEEE/CAA Journal of Automatica Sinica, vol. 6, no. 3, pp. 703–715, 2019.

[14] M. Tutkan, M. C. Ganiz, and S. Akyoku¸s, “Helmholtz principle based supervised and unsupervised feature selection methods for text mining,” Information Processing & Management, vol. 52, no. 5, pp. 885–910, 2016.

[15] S. Wang and H. Wang, “Unsupervised feature selection via low-rank approximation and structure learning,” Knowledge-Based Systems, vol. 124, pp. 70–79, 2017.

[16] Y. Liu, K. Liu, C. Zhang, J. Wang, and X. Wang, “Unsupervised feature selection via diversity-induced self-representation,” Neurocomputing, vol. 219, pp. 350–363, 2017.

[17] R. Hu, X. Zhu, D. Cheng, W. He, Y. Yan, J. Song, and S. Zhang, “Graph self-representation method for unsupervised feature selection,” Neurocomputing, vol. 220, pp. 130–137, 2017.

[18] S. Du, Y. Ma, S. Li, and Y. Ma, “Robust unsupervised feature selection via matrix factorization,” Neurocomputing, vol. 241, pp. 115–127, 2017.

[19] M. Qi, T. Wang, F. Liu, B. Zhang, J. Wang, and Y. Yi, “Unsupervised feature selection by regularized matrix factorization,” Neurocomputing, vol. 273, pp. 593– 610, 2018.

[20] M. Belkin and P. Niyogi, “Towards a theoretical foundation for laplacian-based manifold methods,” Journal of Computer and System Sciences, vol. 74, no. 8, pp. 1289–1308, 2008.

[21] M. Tahir, A. Tubaishat, F. Al-Obeidat, B. Shah, Z. Halim, and M. Waqas, “A novel binary chaotic genetic algorithm for feature selection and its utility in affective computing and healthcare,” Neural Computing and Applications, pp. 1–22, 2020.

[22] S. Jiang, K.-S. Chin, L. Wang, G. Qu, and K. L. Tsui, “Modified genetic algorithm-based feature selection combined with pre-trained deep neural network for demand forecasting in outpatient department,” Expert systems with applications, vol. 82, pp. 216–230, 2017.

[23] M. Rostami, K. Berahmand, and S. Forouzandeh, “A novel community detection based genetic algorithm for feature selection,” Journal of Big Data, vol. 8, no. 1, pp. 1–27, 2021.

[24] D. Berrar, “Cross-validation.” 2019.

[25] H. Gururaj, F. Flammini, H. C. Kumari, G. Puneeth, and B. S. Kumar, “Classifi-cation of drugs based on mechanism of action using machine learning techniques,” Discover Artificial Intelligence, vol. 1, pp. 1–14, 2021.

[26] S. J. Warchal, J. C. Dawson, and N. O. Carragher, “Evaluation of machine learning classifiers to predict compound mechanism of action when transferred across distinct cell lines,” SLAS DISCOVERY: Advancing Life Sciences R&D, vol. 24, no. 3, pp. 224–233, 2019.