COMPOUND SIMILARITY PREDICTION

A PROJECT REPORT

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

COMPUTER SCIENCE AND ENGINEERING

BY

N. Lakshmi Renuka
Roll No: 17331A05A7

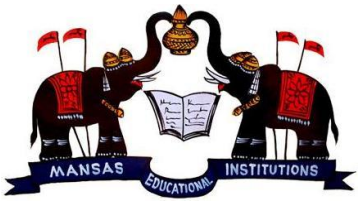
K. Hanisha
Roll No: 17331A0565

M. V. Rama Reddy
Roll No: 17331A0599

Lala P V K GOYAL
Roll No: 17331A0584

Under the Supervision of

Mrs. P. PARIMALA
Assistant Professor



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MVGR COLLEGE OF ENGINEERING (Autonomous)

VIZIANAGARAM-535005, AP (INDIA)

(Accredited by NBA, NAAC, and Permanently Affiliated to Jawaharlal Nehru Technological University Kakinada)

JUNE, 2021

COMPOUND SIMILARITY PREDICTION

A PROJECT REPORT

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

COMPUTER SCIENCE AND ENGINEERING

BY

N. Lakshmi Renuka
Roll No: 17331A05A7

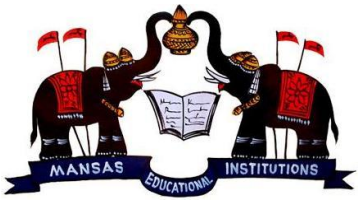
K. Hanisha
Roll No: 17331A0565

M. V. Rama Reddy
Roll No: 17331A0599

Lala P V K Goyal
Roll No: 17331A0584

Under the Supervision of

Mrs. P. PARIMALA
Assistant Professor



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MVGR COLLEGE OF ENGINEERING (Autonomous)

VIZIANAGARAM-535005, AP (INDIA)

(Accredited by NBA, NAAC, and Permanently Affiliated to Jawaharlal Nehru Technological University Kakinada)

JUNE, 2021

Vizianagaram

CERTIFICATE

This is to certify that the project report entitled **"Compound Similarity Prediction"** being submitted by **N. Lakshmi Renuka, K. Hanisha, M. V. Rama Reddy, LALA P V K Goyal** bearing registered numbers 17331A05A7, 17331A0565, 17331A0599, 17331A0584 respectively, in partial fulfillment for the award of the degree of "**Bachelor of Technology" in Computer Science and Engineering** is a record of Bonafide work done by them under my supervision during the academic year 2020-2021.

HOD CSE
Dr. P. Ravi Kiran Varma
Associate Professor
Department of CSE
MVGR College of Engineering

Supervisor
Mrs. P. Parimala
Assistant Professor
Department of CSE
MVGR College of Engineering

External Examiner

ACKNOWLEDGEMENTS

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

I wish to express my sincere and most profound gratitude to **Dr. P. Ravi Kiran Varma**, Head of the Department, Department of CSE, MVGR College of Engineering, Vizianagaram. He has always been a pillar of support for all our work both during the project or otherwise. It was privilege working with him.

I am thankful to **Mrs.P.Parimala**, Project In-charge, project guide, Assistant_Professor, Department of CSE, MVGR College of Engineering, Vizianagaram. For providing me an opportunity to do the project work in and giving us all support and guidance, which made me complete the project duly.

I also thank **Dr.K, V. L. Raju,** Principal, MVGR College of Engineering, Vizianagaram. For providing all provisions for successful completion of project.

I am thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staffs of computer science department which helped us in successfully completing our project work. Also, I would like to extend our sincere esteems to all staff in laboratory for their timely support.

N. Lakshmi Renuka(17331A05A7)
K. Hanisha(17331A0565)
M.V.Rama Reddy(17331A0599)
Lala P V K Goyal(17331A0584)

ABSTRACT

The drug industry is one of the major players guiding the development of the medicines, biotechnology and pharmacology field. Drug discovery is the process by which drugs are discovered and designed. It is a process which aims at identifying a compound therapeutically useful in curing and treating disease. Drug discovery and development pipelines are long, complex and depend on numerous factors. Drug designers mine chemical data from large databases to extract chemical compound that becomes the lead compound in drug discovery. The drug target designing combine machine learning and deep learning algorithm that improve the quality of drugs discovered. Drug discovery involves seven step process that includes disease selection, target hypothesis, lead identification, lead optimization, pre-clinical trial, clinical trial, pharmacogenic identification. Machine learning can be applied to identify the drug targets and in optimization of lead compound. Both supervised and unsupervised algorithms when applied to the databases increases the efficacy of identifying a new target and optimize the lead compound. The process failure rate when machine learning is applied will be very low compared to the traditional drug discovery process. Here we present a Compound Similarity Prediction by using k-means clustering and using dbscan. Through this compound similarity prediction, the compounds are clustered according to their similarity. So, if the reaction of the new compound is unknown, we could know its reaction through this compound similarity prediction. The main advantage of compound similarity prediction is to know the reaction of the new compound. If the reaction of the compound is known then we can speed up the process of drug discovery.

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Need of the Project

Drug discovery is a very complicated process as it involves a huge investment of time and money. On average, it takes 6 to 12 years with an investment of 500 million to 1 billion (in US dollars) to identify a drug for fighting against a target. However, even after a huge struggle, the success rate is very low. Many long-term research projects may end up fruitless resulting in wastage of enormous efforts. Blockbuster drugs are the drugs that are prescribed for the common medical problems like cold, diabetes, high blood pressure, asthma and flu. They are extremely profitable in the pharmaceutical industry. They bring revenues greater than 1 billion per year and a profit of more than 1 million a day (in dollars). However, it can also result in problems for the company if the drug shows any side effects. Usually, the patents on drugs expire resulting in competition from less expensive equivalents. The process of drug discovery is therefore highly complicated and risky activity but is always motivated by the benefits it could do to millions of people suffering from various diseases. The detailed process of drug discovery, illustrated, in below Figure

PROCESS OF DRUG DISCOVERY

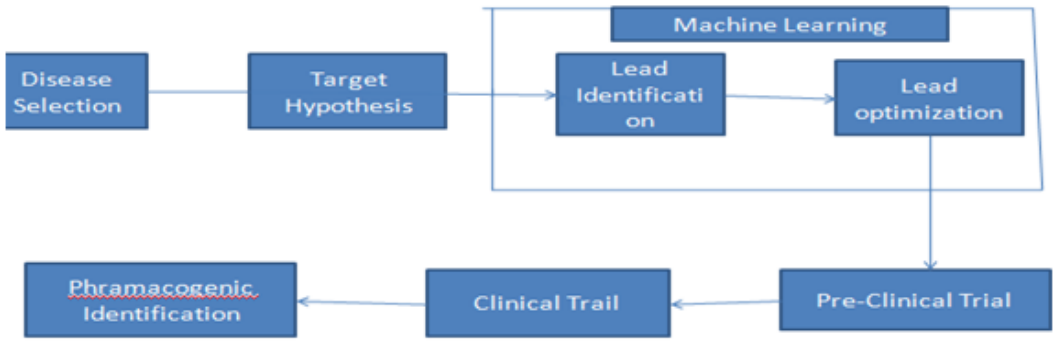


Figure 1.1 Drug Discovery Process

1. Pre-Drug Discovery Process (Disease identification)

2. Modern Drug Discovery Process: The discovery process includes four important processes such as, target identification and validation, lead identification, lead optimization and pre-clinical trials.

a) Target Identification & Validation

used to identify target molecule which can be either gene or protein

b) Lead Identification

Lead identification also helps to see which molecules bind strongly to the target.

c) Lead Optimization

This phase results in finding the drug candidate from the lead identified compound. The goal is a process of refining the chemical structure of a confirmed. Hit to improve its drug characteristics.

d)Pre- Clinical Trial

an important phase to check whether the compound is working correctly or not

e) Clinical Trial

primary phase which will be fastest and safest way to find treatments.

Trials can be done in five ways such as, prevention trials, screening trials, diagnostic trials, treatment trial and quality of life trials.

In this project we work on compounds similarity which is a part of lead identification. Here we take a structure and generate smiles and from smiles we produce chemical fingerprint generation using jupyter notebook and we find similarity between compounds and based on similarity we perform clustering. Drug discovery using traditional methods takes years to discover a new drug. Using machine learning algorithms, we can reduce the time taken to discover a new drug. This motivates us to take drug discovery project. As biomedical data are highly complex, using algorithms in designing new drugs has become more possible than it has ever been. Machine learning can enhance many stages of the drug discovery process. preliminary but crucial stages including designing a drug’s chemical structure. investigating the effect of a drug – both in basic preclinical research and clinical trials, in which a lot of biomedical data is produced. Finding new patterns in those data can be facilitated by machine learning. There are different kinds of data, including genetic and imaging ones. Each of them can be analyzed with machine learning and further used to build novel solutions for drug

discovery. First of all, we need to know the reaction of the new compound in order use that compound in any process so Compound similarity prediction is used to speed up the process.

1.2 About Compound Similarity Prediction

The different compounds are taken and generated smiles from the structure of the compound. Now similarity is found among the smiles. The **simplified molecular-input line-entry system** (**SMILES**) is a specification in the form of a line notation for describing the structure of chemical series using short ASCII strings. SMILES strings can be imported by most molecule editors for conversion back into two-dimensional drawings or three-dimensional models of the molecules.

Typically, a number of equally valid SMILES strings can be written for a molecule. For example, CCO, OCC and C(O)C all specify the structure of ethanol. Algorithms have been developed to generate the same SMILES string for a given molecule; of the many possible strings, these algorithms choose only one of them. This SMILES is unique for each structure, although dependent on the canonicalization algorithm used to generate it, and is termed the canonical SMILES. These algorithms first convert the SMILES to an internal representation of the molecular structure; an algorithm then examines that structure and produces a unique SMILES string. Various algorithms for generating canonical SMILES have been developed and include those by Daylight Chemical Information Systems, OpenEye Scientific Software, MEDIT, Chemical Computing Group, MolSoft LLC, and the Chemistry Development Kit. A common application of canonical SMILES is indexing and ensuring uniqueness of molecules in a database. After generating smiles by considering some features like no of carbons, no of nitrogen etc we generated chemical fingerprint. After knowing chemical fingerprint, we normalized the data and after that using elbow method, we found number of clusters required for clustering the compounds. After knowing the clusters using kmeans clustering technique all compounds are divided into four clusters based on their similarity.

1.3 Hardware and Software requirements

1. Windows 8 or above version
2. Dual Quad-core CPUS
3. Python recent version
4. 4-8Gbs of memory per processor core, with 60% overhead for virtualization

CHAPTER 2

LITERATURE SURVEY

Machine learning plays a crucial role in the field of pharmaceutical industries. Machine learning algorithm most commonly used in drug discovery and development. Machine learning algorithm will be applied to group similar compounds that have similar chemical, biological and physical characteristics of lead compounds in drug discovery. Machine learning approaches are two types i.e., supervised and unsupervised learning. Supervised learning in which goals is to predict the label of new observation given in a large database of labeled example. Supervised learning algorithm are classification, regression, navies bayes, random forest, support vector machine, neural network and deep learning. unsupervised learning and it aims at detecting underlying relationship or patterns in unlabeled data. This method includes dimensionality reduction technique such as principal components analysis, independent component analysis and clustering algorithm etc. Different datasets are used for drug discovery. Depending upon nature of the data set, the machine learning algorithms are used. Machine learning algorithm can be applied to various application of bioinformatic and cheminformatics key to computer-aided drug design is hence the design of an efficient, accurate and highly scoring function using machine learning techniques.

Producing new drugs and marketing them with a complete drug profile is a challenging task as it is a long process and requires a large investment of time and money. Drug repositioning or drug repurposing is the process of identifying new therapeutic uses for existing drugs. It can reduce the time, costs and risks of the traditional drug discovery process. The main goal of drug repositioning is to increase the therapeutic use of the existing drugs in the clinical and medical domain. It is believed that drugs having similar profiles are more likely to share similar behavior in presence of similar targets. There is also evidence that computational drug repositioning can be improved by heterogeneous data analysis. In contrast to laborious in-vivo and in-vitro experiments, computational methods for drug repositioning have become popular as effective and efficient approaches for drug repositioning. These methods focus on identifying new uses for existing drugs and finding new associations between other contributing entities like proteins, genes, diseases and side effects to approach this problem. Pharmacological data can be represented in homogeneous or heterogeneous graphs/networks. Therefore, most of the drug repositioning approaches can be seen as hybrid methods of

graph/network theory concepts and machine learning. Graph clustering is such hybrid approach where graphs of homogeneous and heterogeneous objects can be grouped into small clusters based on their associations. Since pharmacology networks are large and complex, partitioning large networks produces an abstraction which simplifies their complex interaction structure. Realizing the importance of simplifying drug-data network, research has approached partitioning pharmacological networks using various graph theory concepts.

Two compounds that are ‘similar’ to one another will have feature vectors that, when considered as position vectors in the chemical space spanned by the descriptors, are close to each other. If the mapping function varies reasonably slowly and smoothly across chemical space, then we expect similar molecules to have similar values of the relevant chemical (or biological) property. This is the basis of the similar property principle: ‘Similar molecules have similar properties’. While this may be considered a central principle of chemoinformatics, it is far from universally valid. In the case of ‘activity cliffs’ for instance, the mapping function varies dramatically over a small distance in chemical space, corresponding perhaps to a change of one functional group which might prevent a ligand from binding effectively to a protein, and apparently similar molecules can have very different bioactivities.

An artificial neural network (ANN), often simply called a neural network where confusion with biology is unlikely, is a mathematical model used for pattern WIREs Computational Molecular Science Machine learning methods in chemoinformatics recognition and machine learning. The network’s architecture is based on connected neurons in an input layer, a hidden layer or layers, and an output layer. In a typical design, each connection between Neurons carries a weight. The weights are varied during the training phase as the network learns how to connect input and output data, before being tested on unseen instances. While the ANN is inspired by the structure and function of the human brain, it is massively simpler in design and in no way simulates higher brain function.

Algorithm	Description
Ant Colony[87]	Uses virtual pheromones based on ant behavior for optimization
Relevance Vector Machine (RVM)[88]	Sparse probabilistic binary classifier related to SVM; gives probabilities rather than all-or-nothing classification
Parzen-Rosenblatt Window[82,83,89]	Kernel density estimation method that allows molecular similarities to be transformed into probabilities of class membership
Fuzzy Logic[90]	Designed to give interpretable rules based on descriptor values
Rough Sets[91]	Rule-based method designed to give interpretable rules
Support Vector Inductive Logic Programming (SVILP)[84]	Rule-based method incorporating SVM ideas
Winnow[47,85,92,93]	For every class, Winnow learns a vector of weights for each feature. Test instances are compared with these using score thresholds
Decision Tree[23,76,94,95]	Like one tree from a Random Forest, but without randomization
Linear Discriminant Analysis (LDA)[96,97]	Models statistical differences between classes in order to make a classification
kScore[98]	Analogous to a weighted kNN scheme in which the weights are optimized by Leave-One-Out cross-validation
Projection to Latent Structures (PLS)[29,52,68]	Obtains a linear regression by projecting *x* and *y* variables to a new space. Also called Partial Least Squares

figure 2.1 Algorithms

2.1 Data Collection

The Dataset name is HSP90.This will involve collection of data with chemical structure and activity data after this preprocessing is applied to extract feature. The records in the dataset are around 1000 records. The main fields in the dataset are structure, smiles, compound id, molecular weight, H-bond donors, H-bond acceptors, Chemical Series, target. We collected the dataset from https://drugdesigndata.org/about/datasets/408. It was added at 08:57am on September 2,2016.

Source: drugdesigndata

Data Size: 1100 KB

Data Volume: 1000 records

Column Details:

1. Structure
2. Smiles
3. Compound id
4. Molecular weight
5. H-bond donors
6. H-bond acceptors
7. Lipinski violation
8. Rotatable bonds
9. Chemical series
10. PDB ID

11. FRET Assay1: Target
12. FRET Assay 1: IC50
13. Thermofluor Assay: Target
14. Thermofluor Assay: Kd
15. Thermofluor Assay: SEM
16. ITC: Target
17. ITC: Kd
18. ITC: SEM

Expected Output:

Finally, compounds are clustered according to their similarity.

CHAPTER 3

THEORETICAL BACKGROUND

3.1 Machine Learning

Machine learning (**ML**) is the study of computer algorithms that improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks. A subset of machine learning is closely related to computational statistics, which focuses on making predictions using computers; but not all machine learning is statistical learning. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics. Machine learning involves computers discovering how they can perform tasks without being explicitly programmed to do so. It involves computers learning from data provided so that they carry out certain tasks. For simple tasks assigned to computers, it is possible to program algorithms telling the machine how to execute all steps required to solve the problem at hand; on the computer's part, no learning is needed. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. In practice, it can turn out to be more effective to help the machine develop its own algorithm, rather than having human programmers specify every needed step.

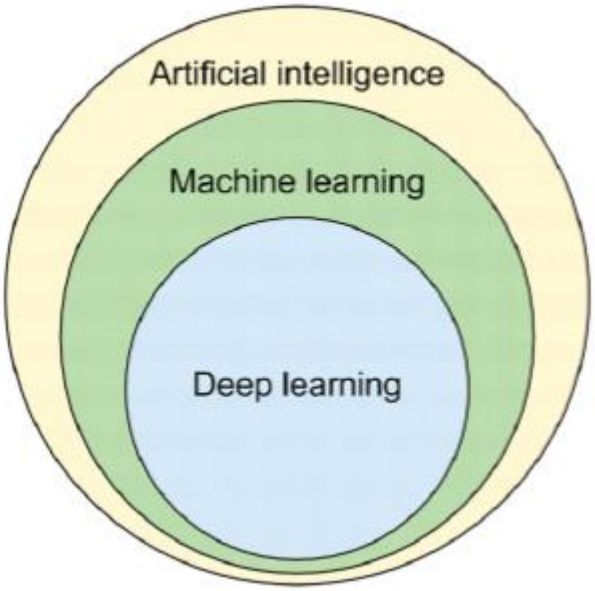


Figure 3.1 Machine Learning

3.2 Supervised learning

Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs. The data is known as training data, and consists of a set of training examples. Each training example has one or more inputs and the desired output, also known as a supervisory signal. In the mathematical model, each training example is represented by an array or vector, sometimes called a feature vector, and the training data is represented by a matrix. Through iterative optimization of an objective function, supervised learning algorithms learn a function that can be used to predict the output associated with new inputs. An optimal function will allow the algorithm to correctly determine the output for inputs that were not a part of the training data. An algorithm that improves the accuracy of its outputs or predictions over time is said to have learned to perform that task.

1. **Multiple regression analysis**- A statistical process to find relationships between dependent variables and one or more independent variables. Results of Multiple Regression Analysis (MRA)Recall that MRA is a statistical procedure that assesses the relationship between a dependent variable and several predictor variables. The estimates generated by MRA are called coefficient. Using MRA, we can calculate the amount of variance in the dependent variable that is accounted for (= explained) by the variation in each of the independent variables. This calculation shows the relative importance of each independent variable to the relationship.

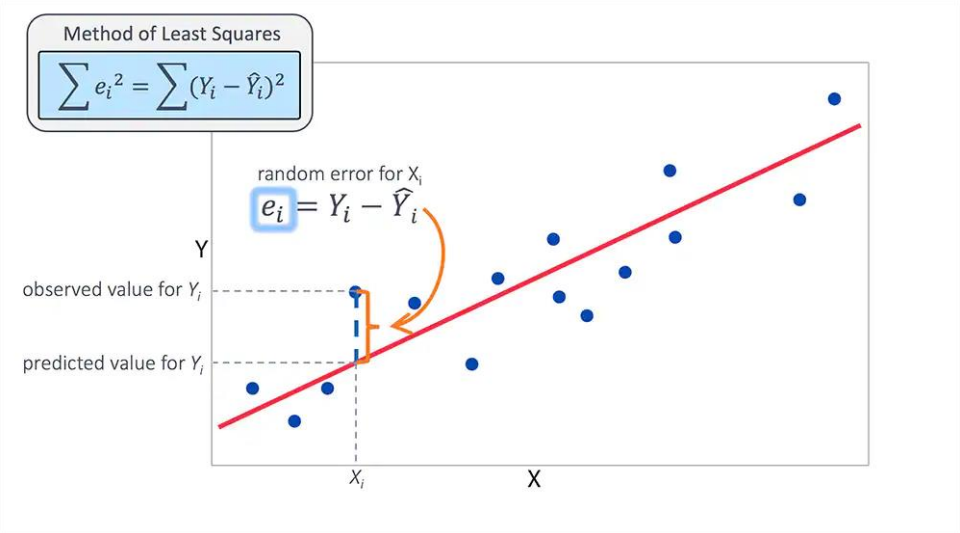


Figure 3.2 Multiple regression analysis

2. **k-nearest neighbor** -An instance-based learning where an object is classified by the majority rule among its k nearest neighbor, where k is an integer. The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other.

The KNN Algorithm

1.Load the data
2.Initialize K to your chosen number of neighbors
3. For each example in the data
3.1 Calculate the distance between the query example and the current example from the data.
3.2 Add the distance and the index of the example to an ordered collection
4. Sort the ordered collection of distances and indices from smallest to largest (in ascending order) by the distances
5. Pick the first K entries from the sorted collection

6. Get the labels of the selected K entries

7. If regression, return the mean of the K labels

8. If classification, return the mode of the K labels

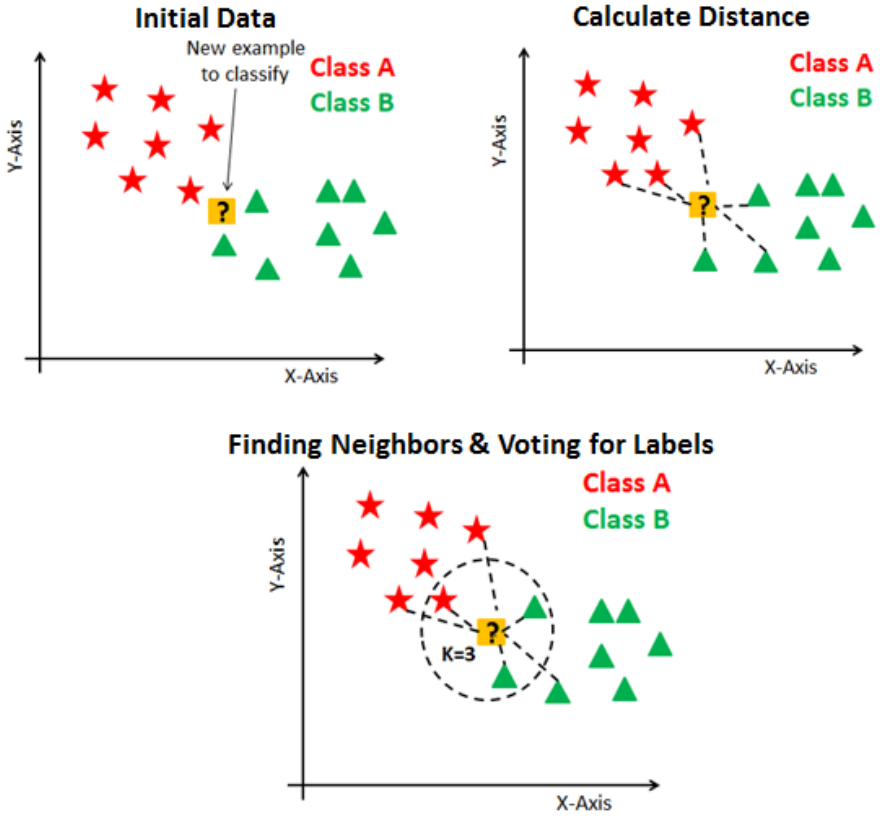


Figure 3.3 k-nearest neighbor

3. **Naive bayes**- A probabilistic approach that uses probability prior and Bayes rule to predict membership by assuming feature independency. Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. There is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness, and diameter features.

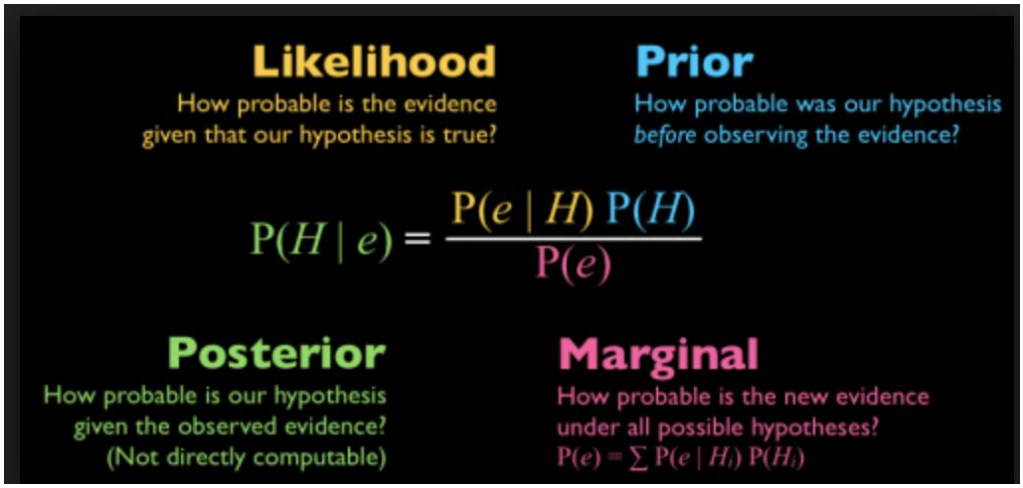


Figure 3.4 Naive bayes

4. **Random Forest** -A classification technique based on the assemble of multiple decision trees and majority voting rules. **Random forests** or **random decision forests** are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean/average prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

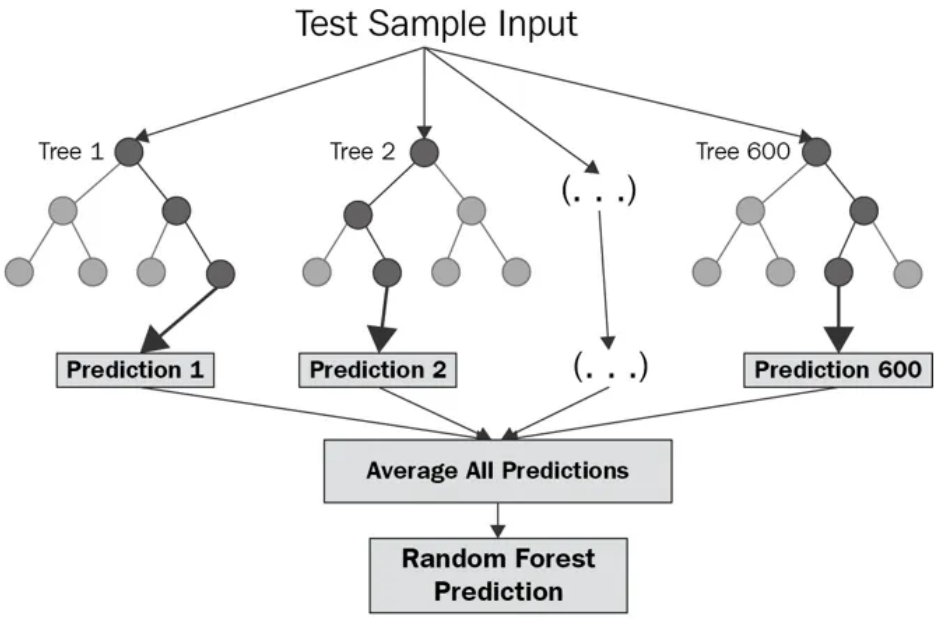


Figure 3.5 Random Forest

5.**Neural network and deep learning**- A model-based learning method that learns from input data based on layers of connected neurons consisting of input layers, multiple hidden layers (for deep learning) and output layers. Deep learning is a class of machine learning learning algorithms that uses multiple layers to progressively extract higher-level features from the raw input. For example, in image processing, lower layers may identify edges, while higher layers may identify the concepts relevant to a human such as digits or letters or faces.

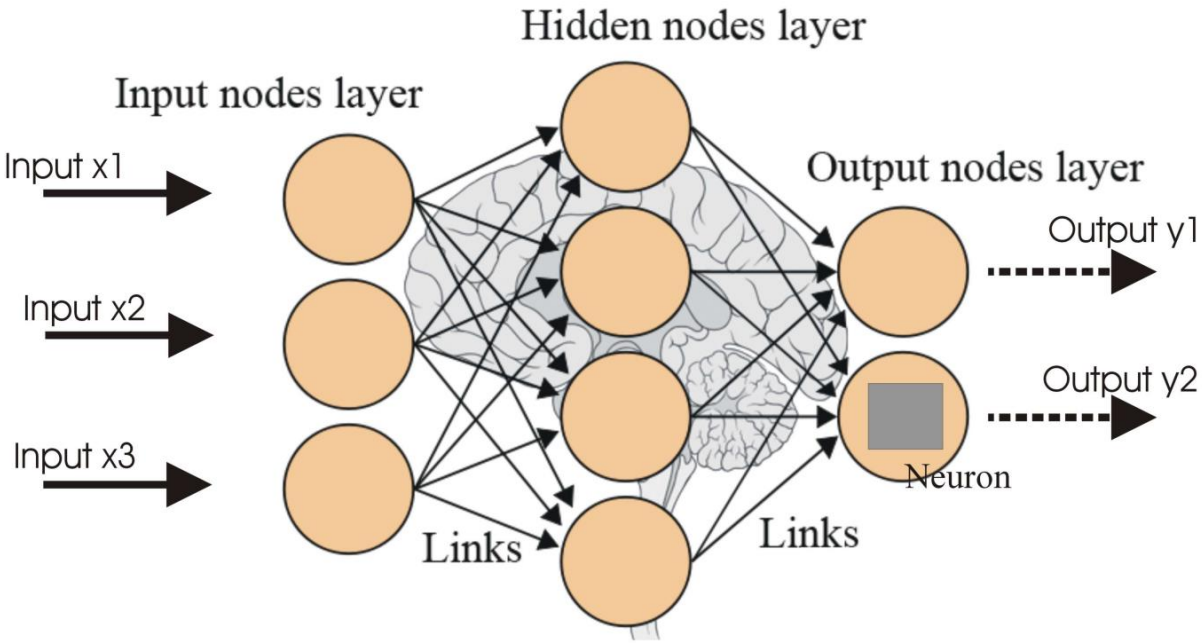


Figure 3.6 Neural network

6. **Support vector machine**- A statistical method that maps data into high-dimensional space to identify a lower dimensional hyperplane that maximizes the data separation using a nonlinear kernel. This is achieved by maximizing the margins between hyperplanes known as support vectors. Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane

that differentiate the two classes very well. Support Vector Machine is a frontier which best segregates the two classes.

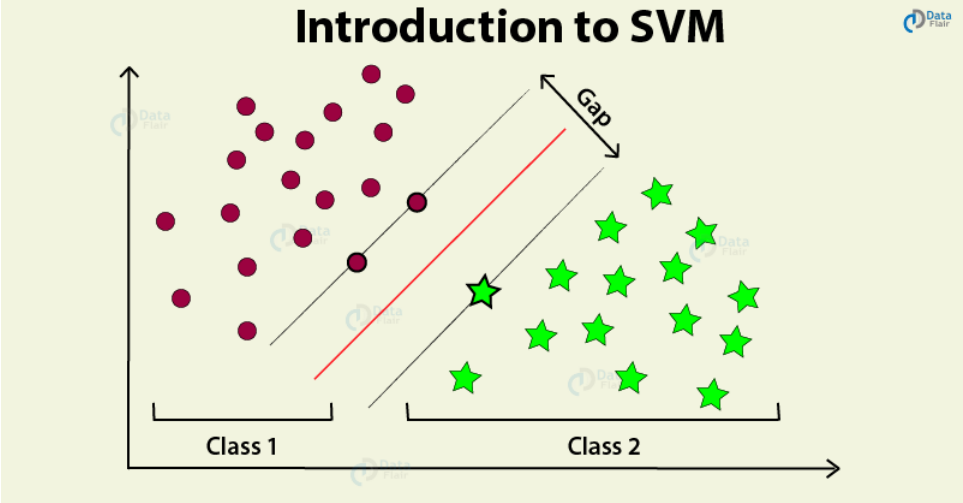


Figure 3.7 Support vector machine

3.3 Unsupervised learning

Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms, therefore, learn from test data that has not been labeled, classified or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data. A central application of unsupervised learning is in the field of density estimation in statistics, such as finding the probability density function. Though unsupervised learning encompasses other domains involving summarizing and explaining data features.

1. **k-means clustering**-A classification method that classifies data into k groups by minimizing within-group distances to the centroid. ***k*-means clustering** is a method of vector quantization, originally from signal processing, that aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells. *k*-means clustering minimizes within-cluster variances (squared Euclidean distances), but not regular Euclidean distances, which would be the more difficult Weber problem: the mean optimizes squared errors, whereas only the geometric median minimizes Euclidean distances. For instance, better Euclidean solutions can be found using k-medians and k-medoids.

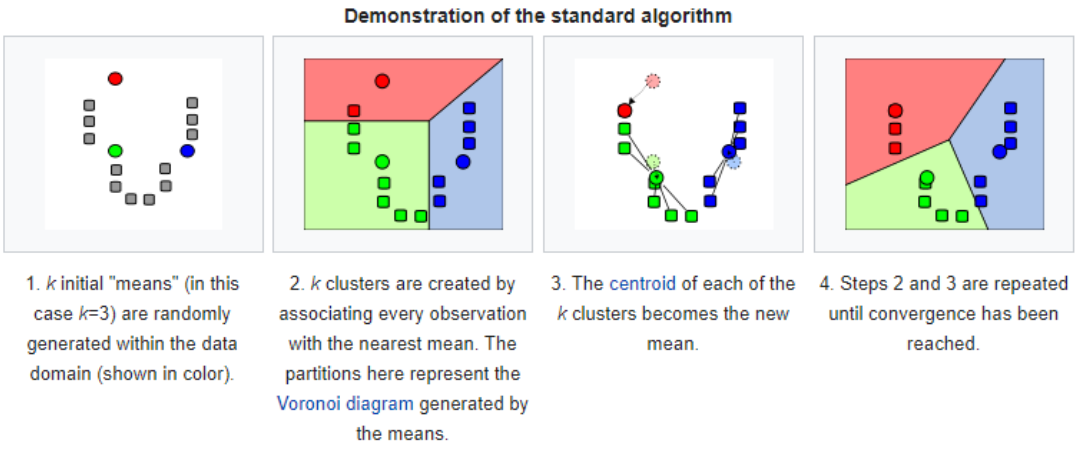


Figure 3.8 k-means clustering

2. **Hierarchical clustering**-A classification method that builds a hierarchy of clusters by agglomerative clustering e.g., merging smaller clusters or divisive clustering e.g., splitting a large cluster to smaller ones. ***Hierarchical clustering***, also known as *hierarchical cluster analysis*, is an algorithm that groups similar objects into groups called *clusters*. The endpoint is a set of clusters, where each cluster is distinct from each other cluster, and the objects within each cluster are broadly similar to each other.

Hierarchical clustering starts by treating each observation as a separate cluster. Then, it repeatedly executes the following two steps: (1) identify the two clusters that are closest together, and (2) merge the two most similar clusters. This iterative process continues until all the clusters are merged together. This is illustrated in the diagrams below.

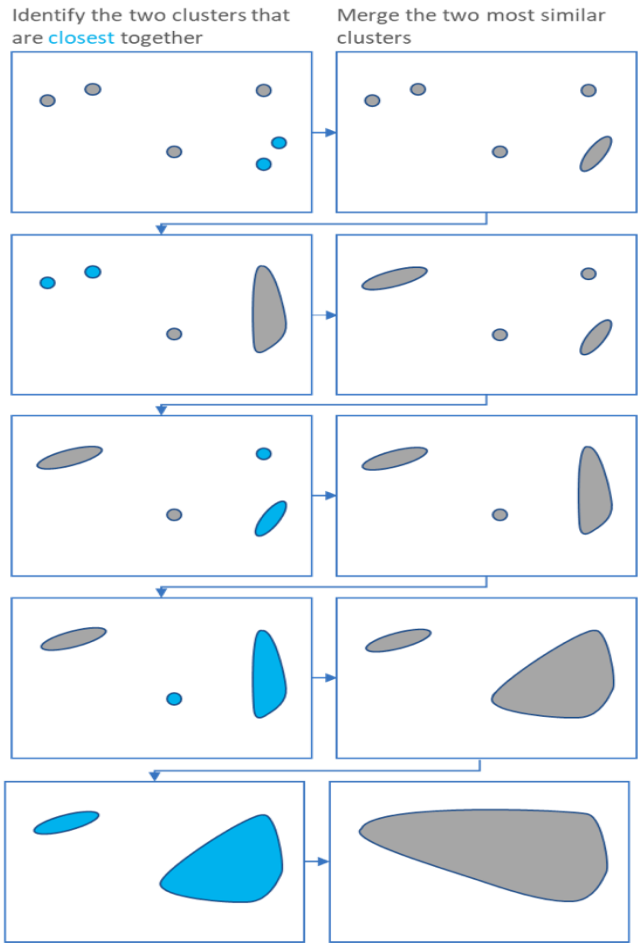


Figure 3.9 Hierarchical clustering

The main output of Hierarchical Clustering is a *dendrogram*, which shows the hierarchical relationship between the clusters:

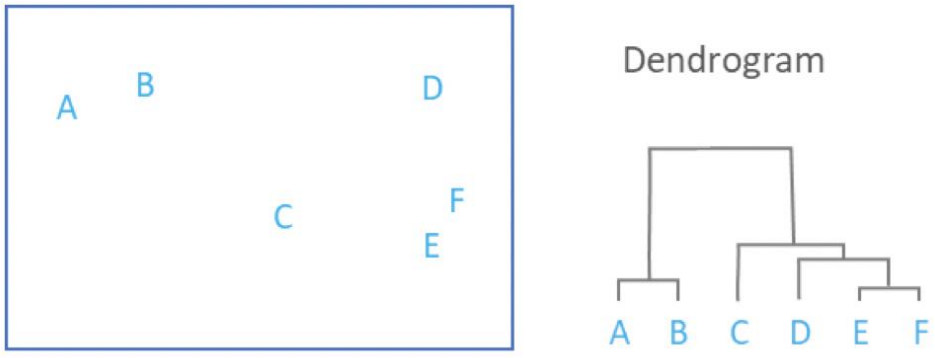


Figure 3.10 Dendrogram

3. **Principal component analysis**-A statistical method that uses orthogonal procedure to transform a set of correlated features to new independent variables called principal

components. Principal Component Analysis (PCA) is an unsupervised machine learning technique that attempts to derive a set of low-dimensional set of features from a much larger set while still preserving as much variance as possible. Perhaps the two main applications of PCA are. Variable selection. Visualizing High-Dimensional.

4. **Independent component analysis** -A statistical method that separates a multivariable output into statistical independent additive components. Independent Component Analysis (ICA) is a machine learning technique to separate independent sources from a mixed signal. Unlike principal component analysis which focuses on maximizing the variance of the data points, the independent component analysis focuses on independence, i.e., independent components.

3.4 Semi-supervised learning

Semi-supervised learning falls between unsupervised learning (without any labeled training data) and supervised learning (with completely labeled training data). Some of the training examples are missing training labels, yet many machine-learning researchers have found that unlabeled data, when used in conjunction with a small amount of labeled data, can produce a considerable improvement in learning accuracy.

3.5 Reinforcement learning

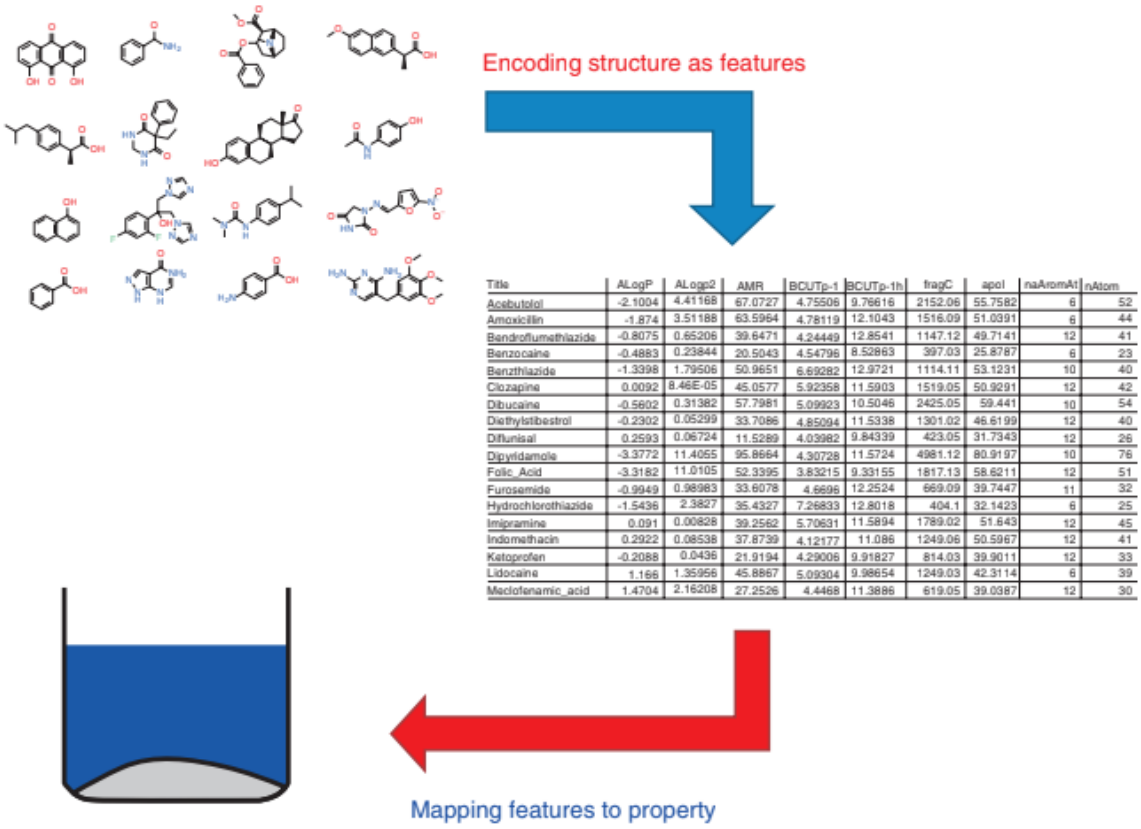
Reinforcement learning is an area of machine learning concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward. Due to its generality, the field is studied in many other disciplines, such as game theory, control theory, operations research, information theory, simulation-based optimization, multi-agent systems, swarm intelligence, statistics and genetic algorithms. In machine learning, the environment is typically represented as a Markov decision process (MDP). Many reinforcement learning algorithms use dynamic programming techniques programming techniques. Reinforcement learning algorithms do not assume knowledge of an exact mathematical model of the MDP, and are used when exact models are infeasible. Reinforcement learning algorithms are used in autonomous vehicles or in learning to play a game against a human opponent

CHAPTER 4

SYSTEM DESIGN AND IMPLEMENTATION

4.1 System Architecture

The data in the dataset is taken through pandas and it is done in jupyter notebook. Smiles are taken and we need to generate chemical fingerprint and for that we have taken features like



Title	ALogP	ALogp2	AMR	BCUTp-1	BCUTp-1h	fragC	apol	naAromAt	nAtom
Acebutolol	-2.1004	4.41168	67.0727	4.75506	9.76616	2152.06	55.7582	6	52
Amoxicillin	-1.874	3.51188	63.5964	4.78119	12.1043	1516.09	51.0391	6	44
Bendroflumethiazide	-0.8075	0.65206	39.6471	4.24449	12.8541	1147.12	49.7141	12	41
Benzocaine	-0.4883	0.23844	20.5043	4.54796	8.52863	397.03	25.8787	6	23
Benzthiazide	-1.3398	1.79506	50.9651	6.69282	12.9721	1114.11	53.1231	10	40
Clozapine	0.0092	8.46E-05	45.0577	5.92358	11.5903	1519.05	50.9291	12	42
Dibucaine	-0.5602	0.31382	57.7981	5.09923	10.5046	2425.05	59.441	10	54
Diethylstibestrol	-0.2302	0.05299	33.7086	4.85094	11.5338	1301.02	46.6199	12	40
Diflunisal	0.2593	0.06724	11.5289	4.03982	9.84339	423.05	31.7343	12	26
Dipyridamole	-3.3772	11.4055	95.8664	4.30728	11.5724	4981.12	80.9197	10	76
Folic_Acid	-3.3182	11.0105	52.3395	3.83215	9.33155	1817.13	58.6211	12	51
Furosemide	-0.9949	0.98983	33.6078	4.6696	12.2524	669.09	39.7447	11	32
Hydrochlorothiazide	-1.5436	2.3827	35.4327	7.26833	12.8018	404.1	32.1423	6	25
Imipramine	0.091	0.00828	39.2562	5.70631	11.5894	1789.02	51.643	12	45
Indomethacin	0.2922	0.08538	37.8739	4.12177	11.086	1249.06	50.5967	12	41
Ketoprofen	-0.2088	0.0436	21.9194	4.29006	9.91827	814.03	39.9011	12	33
Lidocaine	1.166	1.35956	45.8867	5.09304	9.98654	1249.03	42.3114	6	39
Meclofenamic_acid	1.4704	2.16208	27.2526	4.4468	11.3886	619.05	39.0387	12	30

Figure 4.1 System Architecture

No of carbons, no of nitrogen's, no of oxygens, no of fluorine's, no of double bonds, no of c-n bonds, no of n-c bonds, no of o-c bonds, no of c-o bonds, no of c-c bonds. After that, using sklearn preprocessing is done. Preprocessing is used to convert data into range of 0s and 1s. Now we apply kmeans for clustering of data. Before that we need to know number of clusters to be used. For that we use elbow method to know about number of clusters. After knowing the number of clusters, we perform kmeans clustering. Using principal component analysis, we perform dimensionality reduction and we set it to two and finally graph is obtained.

4.2 Procedure

1. Upload and read the data from the dataset
2. Data preprocessing
3. Feature extraction from compound databases
4. Chemical fingerprint generation
5. Find the similarity among compounds
6. Clustering

System design is transition from a user-oriented document to programmers or data base personnel. The design is a solution, how to approach to the creation of a new system. This is composed of several steps. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Designing goes through logical and physical stages of development, logical design reviews the present physical system, prepare input and output specification, details of implementation plan and prepare a logical design walkthrough.

Software Design:

In designing the software following principles are followed:

Modularity and partitioning:

Software is designed such that, each system should consist of hierarchy of modules and serve to partition into separate function.

Coupling:

Modules should have little dependence on other modules of a system.

Cohesion:

Modules should carry out in a single processing function.

Shared use:

Avoid duplication by allowing a single module is called by other that needs the function it provides.

4.3 Algorithms, Tools and Techniques Used

K-Means Clustering Algorithm:

We are given a data set of items, with certain features, and values for these features (like a vector). The task is to categorize those items into groups. To achieve this, we will use the k-Means algorithm; an unsupervised learning algorithm.

The algorithm works as follows:

1. First, we initialize k points, called means, randomly.
2. We categorize each item to its closest mean and we update the mean's coordinates, which are the averages of the items categorized in that mean so far.
3. We repeat the process for a given number of iterations and at the end, we have our clusters.

The "points" mentioned above are called means, because they hold the mean values of the items categorized in it. To initialize these means, we have a lot of options. An intuitive method is to initialize the means at random items in the data set. Another method is to initialize the means at random values between the boundaries of the data set (if for a feature *x* the items have values in [0,3], we will initialize the means with values for *x* at [0,3]). The above algorithm in pseudocode:

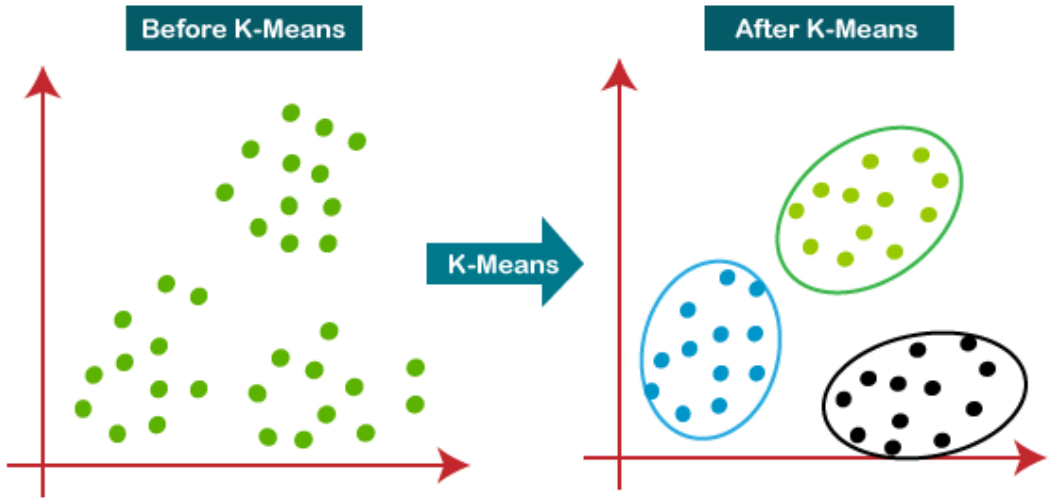


Figure 4.2 Kmeans Clustering

Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids. (It can be other from the input dataset).

Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.

Step-4: Calculate the variance and place a new centroid of each cluster.

Step-5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.

Step-7: The model is ready.

The performance of the K-means clustering algorithm depends upon highly efficient clusters that it forms. But choosing the optimal number of clusters is a big task. There are some different ways to find the optimal number of clusters, but here we are discussing the most appropriate method to find the number of clusters or value of K. The method is given below:

Elbow method

The Elbow method is one of the most popular ways to find the optimal number of clusters. This method uses the concept of WCSS value. **WCSS** stands for **Within Cluster Sum of Squares**, which defines the total variations within a cluster. The formula to calculate the value of WCSS (for 3 clusters) is given below:

$$WCSS= \sum_{Pi \text{ in Cluster1}} distance(P_i\ C_1)^2 + \sum_{Pi \text{ in Cluster2}} distance(P_i\ C_2)^2 + \sum_{Pi \text{ in CLuster3}} distance(P_i\ C_3)^2$$

In the above formula of WCSS,

$\sum_{Pi \text{ in Cluster1}} distance\ (P_i\ C_1)^2$: It is the sum of the square of the distances between each data point and its centroid within a cluster1 and the same for the other two terms.

To measure the distance between data points and centroid, we can use any method such as Euclidean distance or Manhattan distance.

To find the optimal value of clusters, the elbow method follows the below steps:

1. It executes the K-means clustering on a given dataset for different K values (ranges from 1-10).
2. For each value of K, calculates the WCSS value.
3. Plots a curve between calculated WCSS values and the number of clusters K.
4. The sharp point of bend or a point of the plot looks like an arm, then that point is considered as the best value of K.

Since the graph shows the sharp bend, which looks like an elbow, hence it is known as the elbow method. The graph for the elbow method looks like the below image:

Example image

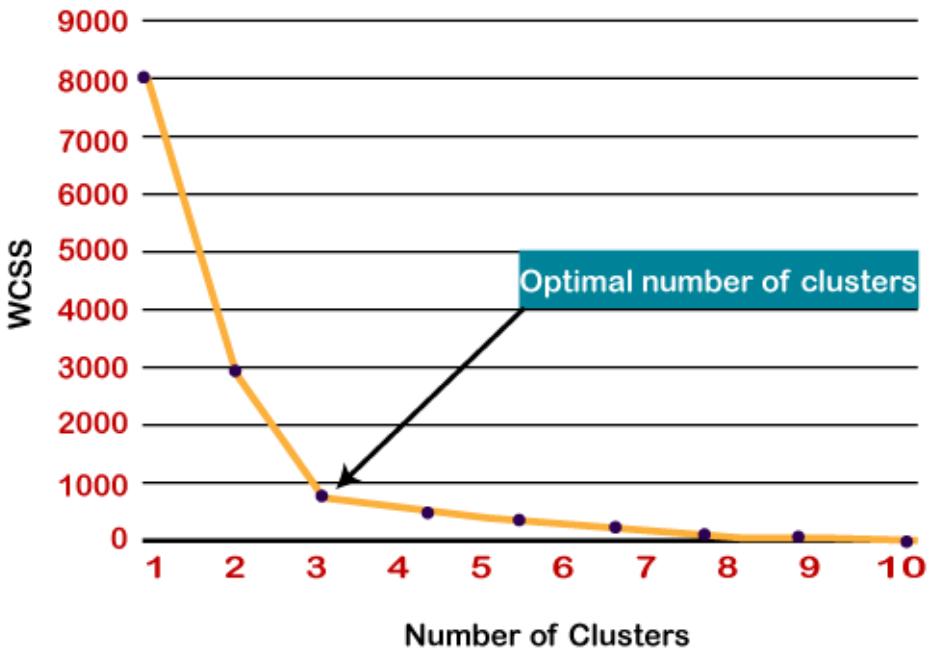


Figure 4.3 Elbow Method

Density-Based Clustering Algorithm

Density-Based **Clustering** refers to unsupervised learning methods that identify distinctive groups/clusters in the data, based on the idea that a cluster in data space is a contiguous region of high point density, separated from other such clusters by contiguous regions of low point density.

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a base algorithm for density-based clustering. It can discover clusters of different shapes and sizes from a large amount of data, which is containing noise and outliers.

The DBSCAN algorithm uses two parameters:

1. **minPts:** The minimum number of points (a threshold) clustered together for a region to be considered dense.
2. **eps (ε):** A distance measure that will be used to locate the points in the neighborhood of any point.

These parameters can be understood if we explore two concepts called Density Reachability and Density Connectivity.

Reachability in terms of density establishes a point to be reachable from another if it lies within a particular distance (eps) from it.

Connectivity, on the other hand, involves a transitivity-based chaining-approach to determine whether points are located in a particular cluster. For example, p and q points could be connected if p->r->s->t->q, where a->b means b is in the neighborhood of a.

There are three types of points after the DBSCAN clustering is complete:

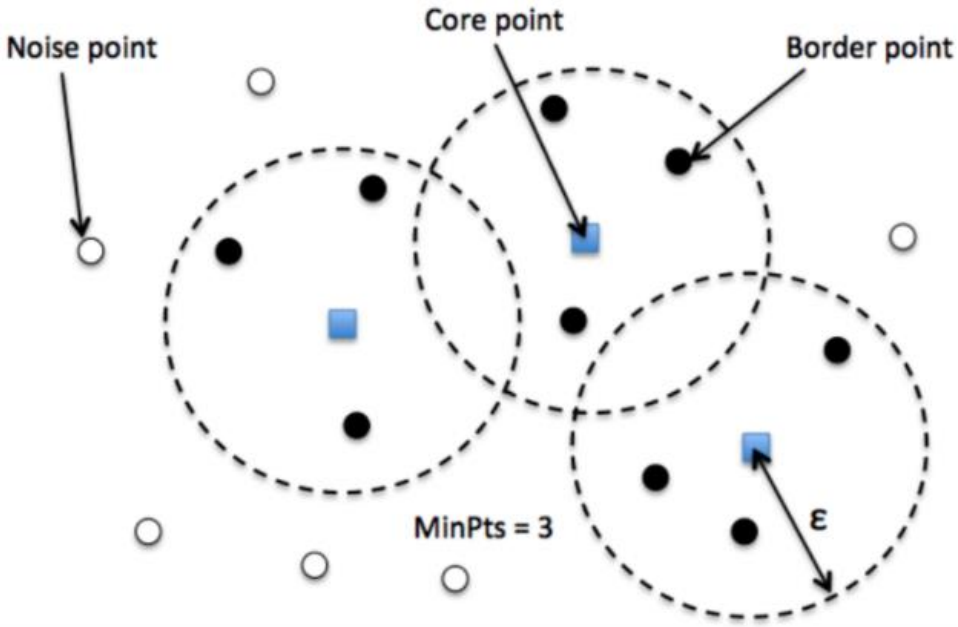


Figure 4.4 Density-Based Clustering

1. **Core** — This is a point that has at least *m* points within distance *n* from itself.
2. **Border** — This is a point that has at least one Core point at a distance *n*.
3. **Noise** — This is a point that is neither a Core nor a Border. And it has less than *m* points within distance *n* from itself.
4. **Algorithmic steps for DBSCAN clustering**
5. The algorithm proceeds by arbitrarily picking up a point in the dataset (until all points have been visited).
6. If there are at least 'minPoint' points within a radius of 'ε' to the point then we consider all these points to be part of the same cluster.
7. The clusters are then expanded by recursively repeating the neighborhood calculation for each neighboring point

Parameter Estimation

Every data mining task has the problem of parameters. Every parameter influences the algorithm in specific ways. For DBSCAN, the parameters **ε** and **minPts** are needed.

- **minPts**: As a rule of thumb, a minimum *minPts* can be derived from the number of dimensions *D* in the data set, as $\boldsymbol{minPts \geq D + 1}$. The low value $\boldsymbol{minPts = 1}$ does not make sense, as then every point on its own will already be a cluster. With $\boldsymbol{minPts \leq 2}$, the result will be the same as of hierarchical clustering with the single link metric, with the dendrogram cut at height ε. Therefore, *minPts* must be chosen at least 3. However, larger values are usually better for data sets with noise and will yield more significant clusters. As a rule of thumb, $\boldsymbol{minPts = 2 \cdot dim}$ can be used, but it may be necessary to choose larger values for very large data, for noisy data or for data that contains many duplicates.
- **ε**: The value for ε can then be chosen by using a k-distance graph, plotting the distance to the $\boldsymbol{k = minPts - 1}$ nearest neighbor ordered from the largest to the smallest value. Good values of ε are where this plot shows an "elbow": if ε is chosen much too small, a large part of the data will not be clustered; whereas for a too high value of ε, clusters will merge and the majority of objects will be in the same cluster. In general, small values of ε are preferable, and as a rule of thumb, only a small fraction of points should be within this distance of each other.
- **Distance function**: The choice of distance function is tightly linked to the choice of ε, and has a major impact on the outcomes. In general, it will be necessary to first identify a reasonable measure of similarity for the data set, before the parameter ε can be chosen. There is no estimation for this parameter, but the distance functions need to be chosen appropriately for the data set.

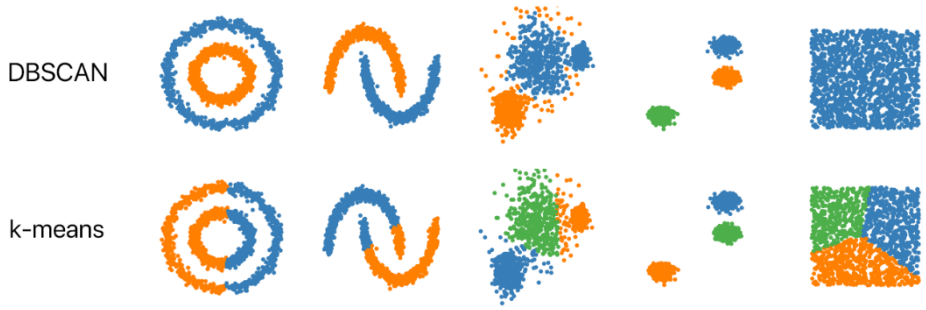


Figure 4.5 Difference between k-means and dbscan

Principal Component Analysis

Principal Component Analysis, or PCA, is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set.
Reducing the number of variables of a data set naturally comes at the expense of accuracy, but the trick in dimensionality reduction is to trade a little accuracy for simplicity. Because smaller data sets are easier to explore and visualize and make analyzing data much easier and faster for machine learning algorithms without extraneous variables to process.
So, to sum up, the idea of PCA is simple — reduce the number of variables of a data set, while preserving as much information as possible.

STEP 1: STANDARDIZATION

Mathematically, this can be done by subtracting the mean and dividing by the standard deviation for each value of each variable.

$$z = \frac{value - mean}{standard\ deviation}$$

Once the standardization is done, all the variables will be transformed to the same scale.

STEP 2: COVARIANCE MATRIX COMPUTATION

The aim of this step is to understand how the variables of the input data set are varying from the mean with respect to each other, or in other words, to see if there is any relationship between them. Because sometimes, variables are highly correlated in such a way that they contain redundant information. So, in order to identify these correlations, we compute the covariance matrix.

STEP 3: COMPUTE THE EIGENVECTORS AND EIGENVALUES OF THE COVARIANCE MATRIX TO IDENTIFY THE PRINCIPAL COMPONENTS

Principal components are new variables that are constructed as linear combinations or mixtures of the initial variables. These combinations are done in such a way that the new variables (i.e., principal components) are uncorrelated and most of the information within the initial variables is squeezed or compressed into the first components. So, the idea is 10-dimensional data gives you 10 principal components, but PCA tries to put maximum possible information in the first component, then maximum remaining information in the second and so on until having something like shown in the scree plot below.

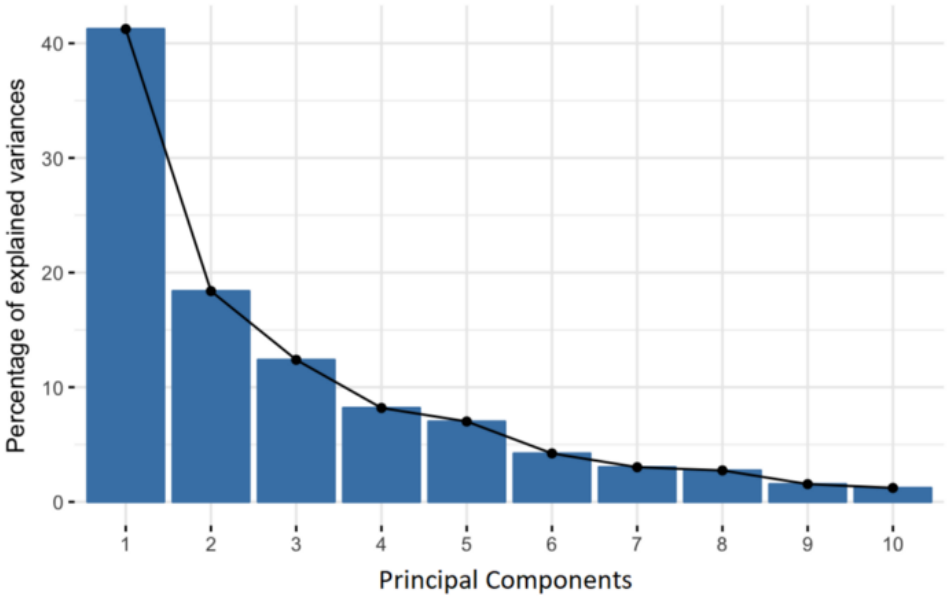


Figure 4.6 Principal Component Analysis

CHAPTER 5

EXPERIMENTAL RESULTS AND DISCUSSION

5.1 Training Snippet

The below snippet is used for clustering using the method kmeans clustering

```
from sklearn.cluster import KMeans
Error =[]
for i in range(1, 10):
   kmeans = KMeans(n_clusters = i).fit(data)
   kmeans.fit(data)
   Error.append(kmeans.inertia_)
import matplotlib.pyplot as plt
plt.plot(range(1, 10), Error)
plt.title('Elbow method')
plt.xlabel('No of clusters')
plt.ylabel('Error')
plt.show()
kmeans2 = KMeans(n_clusters = 4).fit(data)
y_kmeans=kmeans2.fit_predict(data)
```

The below is code snippet for clustering compounds using dbscan

```
from sklearn.cluster import DBSCAN
model = DBSCAN(eps=0.1)
y_dbscan = model.fit_predict(data)
```

5.2 Output

The below is the output of Elbow method

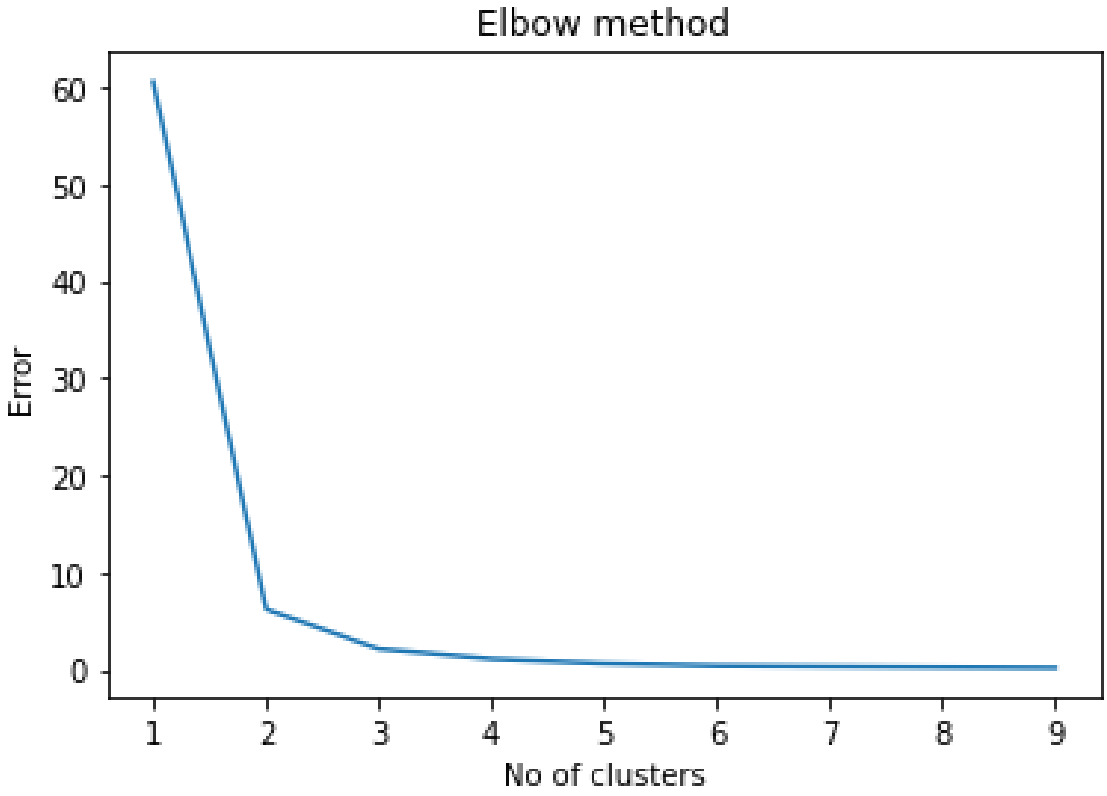


Figure 5.1 Elbow method

kmeans Clustering graph

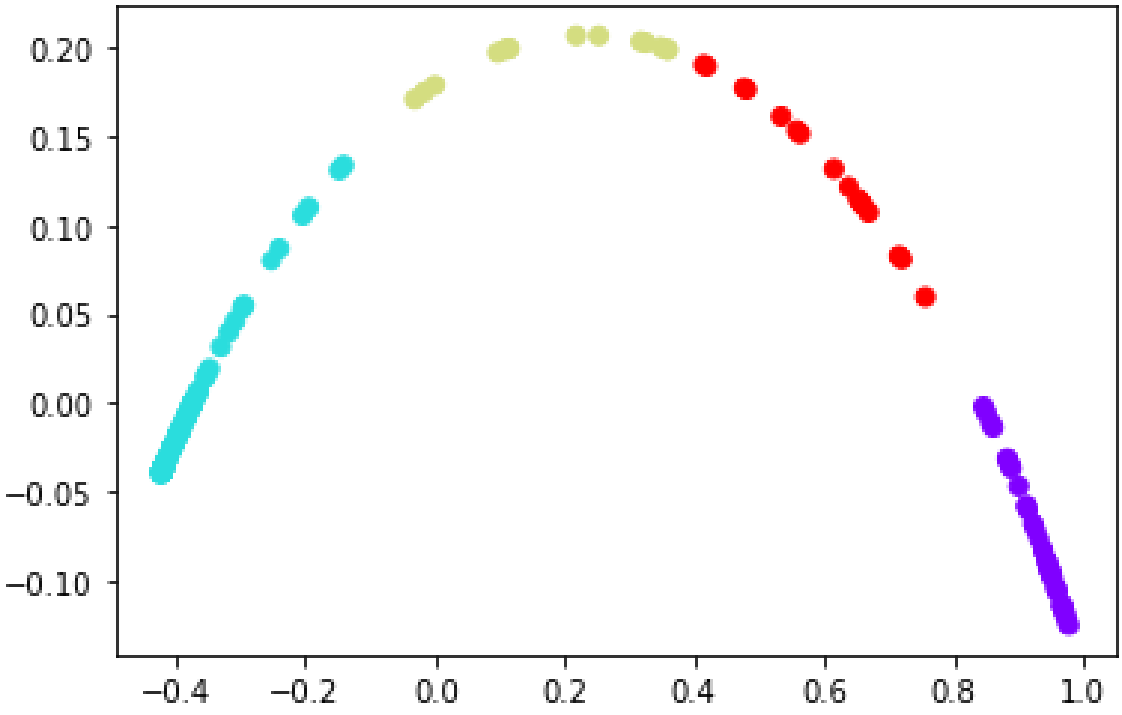


Figure 5.2 Kmeans Clustering

dbscan clustering graph

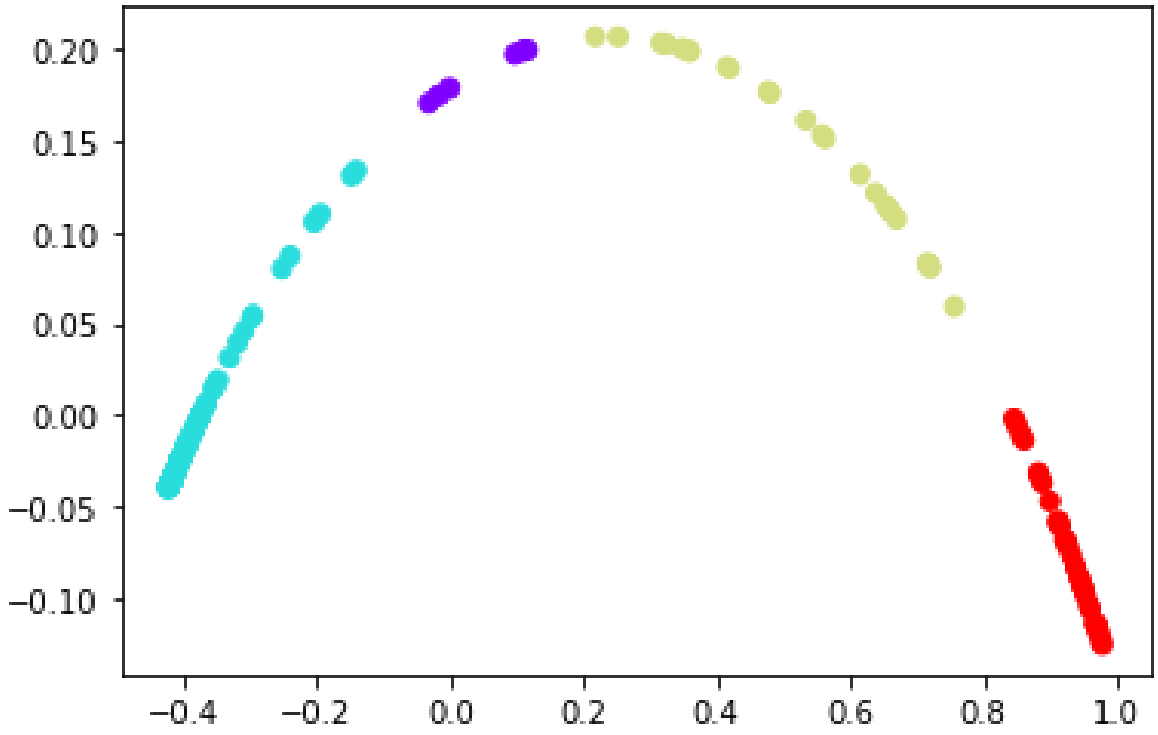


Figure 5.3 Density based Clustering

CHAPTER 6

CONCLUSION

Drug discovery is a long process and takes years to discover a new drug. So in order to reduce time we need machine learning algorithms to speed up the process of drug discovery. We also find many drugs but we don't know the reaction of each drug. If we know the reaction of drug then we could save some time in discovering new medicine.so we have done compound similarity prediction and until now we have read many papers regarding classification, supervised and unsupervised clustering techniques and many algorithms. We used k-means clustering for compound similarity prediction and also, we used density-based clustering algorithm. Finally, we clustered the compounds based on their similarity. We have successfully implemented compound similarity prediction and finally we can know the reaction of the new drug.

References

1. Machine learning in chemoinformatics and drug discovery by Yu-Chen Lo, Stefano E.Rensi, Wen Torng and Russ B.Altman.
2. Applications of machine learning in drug discovery and development by Jessica Vamathevan.
3. A comparative study of smiles-based compound similarity functions for drug-target interaction prediction by Hakime Ozturk,Arzucan Ozgur.
4. A study on cheminformatics and its applications on modern drug discovery by B.Firdaus Begam and Dr.J.Satheesh Kumar.
5. Machine learning for Drug-Target Interaction Prediction by Ruolan Chen,Shuting Jin,Juan Liu.
6. Machine learning methods in chemoinformatics by John B.O.Mitchell.

Appendix A: Packages, Tools used & Working Process

Python Programming language

Python is an interpreted high-level programming language for general-purpose programming. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.
Python is available in two versions, Python 2 and Python 3. This project uses the latest version of Python, i.e., Python 3. Pythons source code is available under GNU-GPL
Python uses a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.
Python's design offers some support for functional programming in the Lisp tradition. It has filter(), map(), and reduce() functions; list comprehensions, dictionaries, and sets; and generator expressions. The standard library has two modules (itertools and functools) that implement functional tools borrowed from Haskell and Standard ML.

Libraries

Pandas
Pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. This helps in handling large amounts of data with help of data structures like Series, Data Frames etc. It has inbuilt methods for reading and writing data in different formats like CSV, xlsx, HTML etc. Different machine learning algorithms have the compatibility for pandas data structures.

Sklearn
Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

1. **Components of scikit-learn:**
2. Scikit-learn comes loaded with a lot of features. Here are a few of them to help you understand the spread:
3. **Supervised learning algorithms:** Think of any supervised machine learning algorithm you might have heard about and there is a very high chance that it is part of scikit-learn. Starting from Generalized linear models (e.g Linear Regression), Support Vector Machines (SVM), Decision Trees to Bayesian methods – all of them are part of scikit-learn toolbox. The spread of machine learning algorithms is one of the big reasons for the high usage of scikit-learn. I started using scikit to solve supervised learning problems
4. **Cross-validation:** There are various methods to check the accuracy of supervised models on unseen data using sklearn.
5. **Unsupervised learning algorithms:** Again, there is a large spread of machine learning algorithms in the offering – starting from clustering, factor analysis, principal component analysis to unsupervised neural networks.
6. **Various toy datasets:** This came in handy while learning scikit-learn. I had learned SAS using various academic datasets (e.g. IRIS dataset, Boston House prices dataset). Having them handy while learning a new library helped a lot.
7. **Feature extraction:** Scikit-learn for extracting features from images and text (e.g. Bag of words)

Matplotlib

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002.One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc. Matplotlib comes with a wide variety of plots. Plots helps to understand trends, patterns, and to make correlations. They’re typically instruments for reasoning about quantitative information.

Tools Used

Anaconda Distribution

The open-source Anaconda Distribution is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X. With over 11 million users worldwide, it is the industry standard for developing, testing, and training on a single machine, enabling individual data scientists to:

Quickly download 1,500+ Python/R data science packages.

Manage libraries, dependencies, and environments with Conda.

Develop and train machine learning and deep learning models with scikit-learn, TensorFlow, and Theano.

Analyze data with scalability and performance with Dask, NumPy, pandas, and Numba.

Visualize results with Matplotlib, Bokeh, Datashader, and Holoviews.

Jupyter Notebook

Jupyter notebook is handy tool for any machine learning programmer. It segregates the code into cells and executes. It helps in auto filling the functions and prompts the syntaxes for the function. It also gives the complete documentation on site for a function that is being used. It supports version control and shows the graphs, charts without the need of any new application. This project uses Jupyter Notebook as an IDE.