

A MINI PROJECT REPORT
ON
MEDICAL DRUG CLASSIFICATION

For the award of Degree of
BACHELOR OF ENGINEERING
IN
CSE (AI ML)

Submitted By

K. VARSHITH	245321748033
A. Ethnic Sai	245321748023
P.G. Praneeth	245321748044

Under the guidance of

DR. SHILPA CHOUDHARY



Department of CSE(AI ML)

NEIL GOGTE INSTITUTE OF TECHNOLOGY
Kachavanisingaram Village, Hyderabad, Telangana 500058.

FEBRUARY 2024



NEIL GOGTE INSTITUTE OF TECHNOLOGY
A Unit of Keshav Memorial Technical Education (KMTES)
Approved by AICTE, New Delhi & Affiliated to Osmania University,
Hyderabad

CERTIFICATE

This is to certify that the Mini project work entitled “MEDICAL DRUG CLASSIFICATION” is a bonafide work carried out by K. VARSHITH (245321748033), A. Ethnic Sai (245321748023), P.G Praneeth (245321748044) of III-year V semester Bachelor of Engineering in CSE(AIML) by Osmania University, Hyderabad during the academic year 2023-2024 is a record of bonafide work carried out by them. The results embodied in this report have not been submitted to any other University or Institution for the award of any degree

Internal Guide

Mrs. Dr. SHILPA CHOUDHARY
Assistant Professor

Head of Department

Dr. T. PREM CHANDAR
Associate Professor

External



NEIL GOGTE INSTITUTE OF TECHNOLOGY

A Unit of Keshav Memorial Technical Education (KMTES)
Approved by AICTE, New Delhi & Affiliated to Osmania University,
Hyderabad

DECLARATION

We hereby declare that the Mini Project Report entitled, “**MEDICAL DRUG CLASSIFICATION**” submitted for the B.E degree is entirely my work and all ideas and references have been duly acknowledged. It does not contain any work for the award of any other degree.

Date:

K. VARSHITH 245321748033
A. ETHNIC SAI 245321748023
P.G PRANEETH 245321748044

ACKNOWLEDGEMENT

We are happy to express our deep sense of gratitude to the principal of the college **Dr. R. Shyam Sunder**, Professor, Neil Gogte Institute of Technology, for having provided us with adequate facilities to pursue our project.

We would like to thank, **Dr. Prem Chander**, Head of the Department, CSE(AIML), Neil Gogte Institute of Technology, for having provided the freedom to use all the facilities available in the department, especially the laboratories and the library.

We would also like to thank our internal guide **Mrs.Dr. ShilpaChoudhary**, Assistant Professor for her technical guidance & constant encouragement.

We sincerely thank my seniors and all the teaching and non-teaching staff of the Department of Computer Science & Engineering for their timely suggestions, healthy criticism, and motivation during this work.

Finally, we express my immense gratitude with pleasure to the other individuals who have either directly or indirectly contributed to our need at the right time for the development and success of this work.

ABSTRACT

Medical drug classification plays a crucial role in personalized healthcare, where treatment decisions are tailored to individual patient characteristics. In this study, we propose a machine learning approach for medical drug classification based on key patient attributes including age, gender, blood pressure (BP), cholesterol levels, and the natriuretic peptide (Natok) ratio.

Utilizing a diverse dataset comprising patient records, including demographic information and clinical measurements, we employed advanced machine learning algorithms to predict the most suitable drug for a given patient. Feature engineering techniques were applied to extract relevant features from the dataset, ensuring comprehensive coverage of patient characteristics.

Various machine learning models, such as decision trees, random forests, support vector machines, and neural networks, were trained and evaluated to identify the optimal approach for drug classification. Additionally, ensemble techniques were employed to enhance prediction accuracy and robustness.

Our results demonstrate the efficacy of the proposed approach in accurately classifying medical drugs based on individual patient profiles. By considering a comprehensive set of features, including age, gender, BP, cholesterol levels, and Natok ratio, our model achieves high performance in drug classification, facilitating personalized treatment recommendations.

This study contributes to the advancement of personalized medicine by leveraging machine learning techniques to tailor medical interventions to the specific needs of patients, thereby optimizing treatment outcomes and improving overall healthcare delivery. Future research directions include refining the model with larger and more diverse datasets and incorporating additional clinical parameters for enhanced prediction accuracy.

II

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE NO.
	ACKNOWLEDGEMENT	I
	ABSTRACT	II
1.	INTRODUCTION	
1.1.	PROBLEM STATEMENT	8
1.2.	MOTIVATION	8
1.3.	SCOPE	8
1.4.	OUTLINE	9
2.	LITERATURE SURVEY	
2.1.	EXISTING SYSTEM	10
2.2.	PROPOSED SYSTEM	11
3.	SYSTEM DESIGN	
3.1.	USE-CASE DIAGRAM	12
3.2.	CLASS DIAGRAM	13
3.3.	SEQUENCE DIAGRAM	13
4.	PROPOSED METHODOLOGY	
4.1.	DATA COLLECTION	15
4.2.	DATA PREPROCESSING	18
4.3.	MODEL_SELECTION	20
5.	RESULT	
5.1.	COMPARITIVE ANALYSIS	31
5.2.	FINAL SELECTION	32

6. IMPEMETATION	
6.1. FRONT-END	34
6.2. BACK-END	35
6.3. WORKING DEMOSTRATION	36
7. CONCLUSION AND FUTURE SCOPE	
7.1. CONCLUSION	37
7.2. FUTURE SCOPE	37
APPENDIX A: TOOLS AND TECHNOLOGY	39
REFERENCES	40

CHAPTER – 1

INTRODUCTION

1.1 PROBLEM STATEMENT

The problem statement involves the development of a machine learning-based approach for medical drug classification, aiming to improve personalized medicine by considering individual patient characteristics such as age, gender, blood pressure, cholesterol levels, and the natriuretic peptide ratio. Challenges include integrating heterogeneous patient data sources, selecting informative features, developing, and evaluating predictive models, and ensuring scalability for real-world deployment. This research seeks to address these challenges to provide healthcare practitioners with a data-driven decision support tool for optimizing drug prescription and improving patient outcomes.

1.2 MOTIVATION

This problem is not challenging but also its solution is applicable to other fine – grained machine learning classification problems, helping hospitals and doctors save time and resources for consulting every patient for small and minor health issues like fever, cold and other problems regarding Blood pressure and Cholesterol etc... I found a dataset used to make prototype of my project Medical Drug Classification. Since it does not contain any real-time medical drugs.

1.3 SCOPE

The scope of the proposed model encompasses the development and validation of a machine learning-based system for medical drug classification, leveraging patient attributes including age, gender, blood pressure, cholesterol levels, and the natriuretic peptide ratio. The model will involve data preprocessing, feature selection, and the implementation of various machine learning algorithms for classification tasks.

Evaluation metrics such as accuracy, precision, recall, and F1 score will be employed to assess model performance. The model's scope also includes the exploration of ensemble methods and cross-validation techniques to enhance predictive accuracy and robustness.

1.4 OUTLINE

The Machine -learning classification model classifies the dataset into two parts, i.e., training dataset and testing dataset. The model learns from the training dataset and then using the testing dataset the model is further tested on the testing dataset. I had initiated a comparative analysis for selecting a perfect classification model that can give high accuracy and high performance. For, Front-end or User Interface, I had used basic HTML, CSS for building it. And I have used python Flask for creating a local server and gateway for the frontend and my machine learning model used for making a responsive web application that can predict the resultant drug based on the input given on HTML form

CHAPTER – 2

LITERATURE SURVEY

2.1 EXISTING SYSTEM

Machine learning (ML) represents a set of techniques that allow systems to discover required representations to features detection or classification from the raw data. The performance of works in the classification system depends on the quality of the features. As such of this study can be categorized under the field of ML, this is to make a search in this area for the studies that belong to Drug classification or prediction. Also, in the medical field there are a smaller number of systems present for predicting a drug based on different features of a human and the existing systems are not mostly responsive and accurate in predicting a drug, since they have less accuracy.

The advancement and progress in technology and related techniques have created a chance for progress in many scientific fields and various industries. Machine learning has become an important tool for drug designs and discovery with the supply of big data from large databases. During this paper I analyze Machine Learning and Deep learning techniques which help the Pharma industry throughout stages of drug discovery which incorporate target validation, prognostic biomarkers, clinical trials. Keywords: Drug discovery, Pharma industry, clinical trials.

Disadvantages of Existing System: -

- Low Accuracy and poor model selection
- No Scalability
- Not suitable for Real-time scenarios

The References mentioned below are about the existing systems: -

- <https://doi.org/10.22214/ijraset.2022.43609>

2.2 PROPOSED SYSTEM

There are various Machine learning algorithms based on classification and prediction so we are performing a comparative analysis between these models and choosing the algorithm which has the highest accuracy and best in performance and most

importantly predicting and classifying the accurate drug based on the features mentioned as input. This system is just a prototype which does not contain any real-time medical drug since there is lack of data based on it but our model is trained on the features as well so it may not be a problem. This prototype system can further improvise and can be implemented in the real-world by training it on the Real-time drug dataset assured by different doctors world-wide.

Advantages of proposed system

- High Accuracy and performance
- The final model will be very Scalable
- The model will be perfectly suitable for real-time scenarios
- It is a prototype but in future it will be implemented into Real-time applications.

CHAPTER – 3

SYSTEM DESIGN

3.1 USE CASE DIAGRAM

The following below diagram(fig.3.1) explains about the structure of the project using a use case diagram.

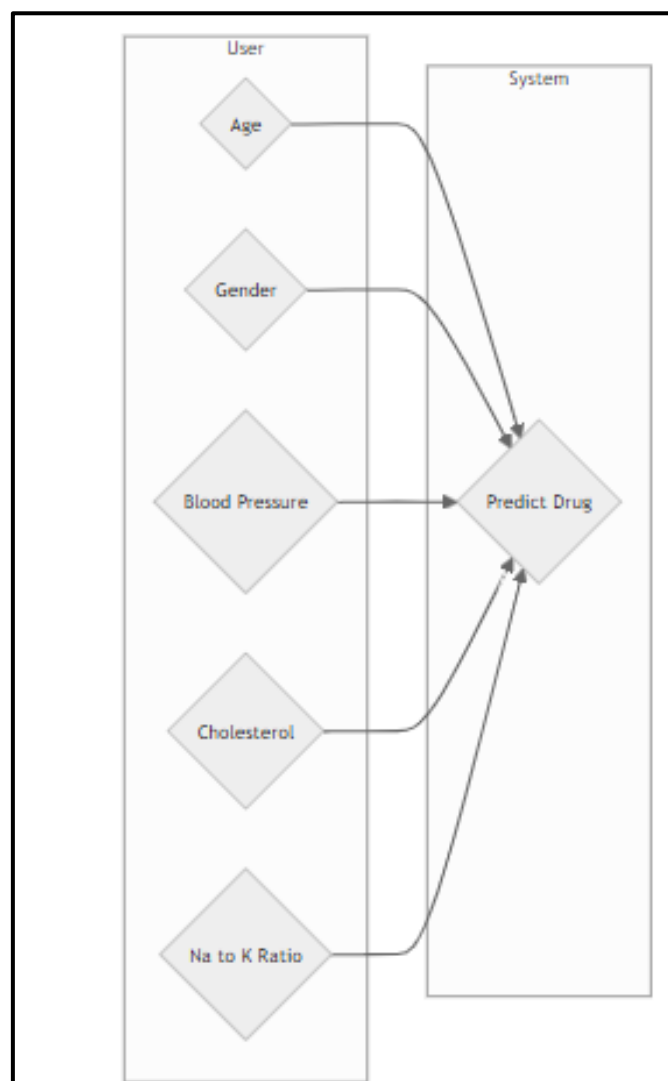


Fig.3.1 Use case diagram of user

3.2 CLASS DIAGRAM

The following below diagram(fig.3.2) represents the class diagram of drug classification and prediction of the suitable drug based on the input features.

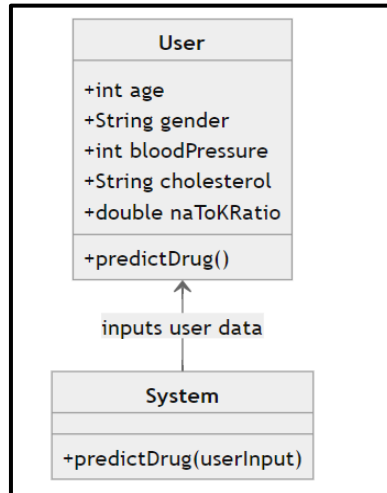


Fig.3.2 Class diagram for Drug classification

3.3 SEQUENCE DIAGRAM

The following images (Fig.3.3) and (Fig.3.4) below represents the sequence diagrams of the project Drug classification (Fig.3.4) represents the sequence diagram using the actors.

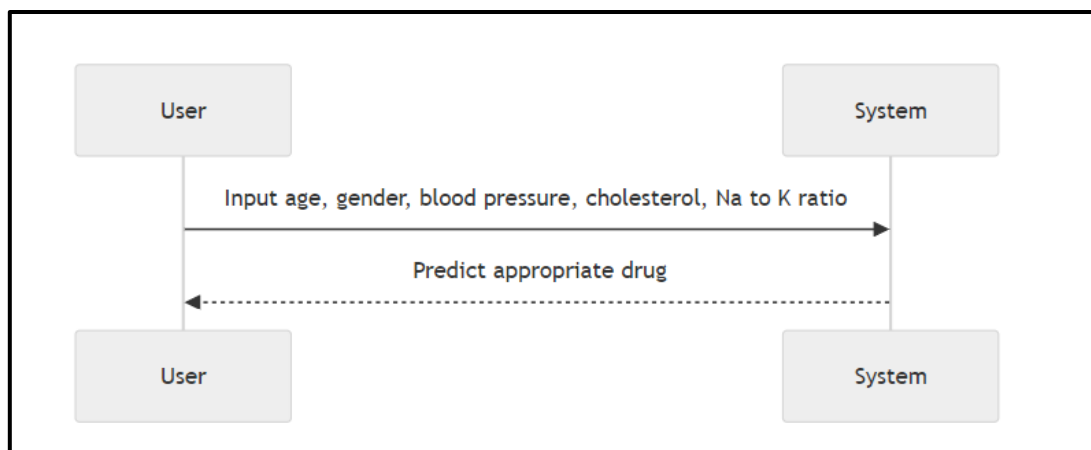


Fig.3.3a Sequence diagram of Drug classification

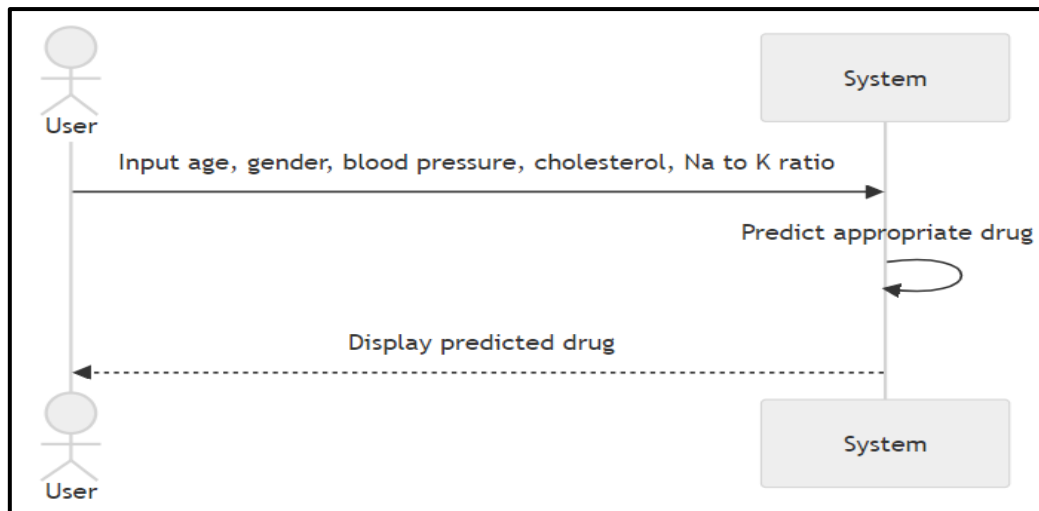


Fig.3.3b Sequence diagram of Drug classification

CHAPTER - 4

Proposed Methodology

In this chapter, we presented a detailed proposed methodology for drug classification using machine learning (ML). The approach focuses on overcoming the limitations of current methodologies and outlines a structured process involving data collection, preprocessing, feature selection, model selection, and evaluation.

Drug classification is an essential aspect of pharmaceutical research, aiming to categorize drugs based on their patient properties, symptoms, and other information. Understanding the background of drug classification is crucial to appreciate the complexities involved in developing effective methodologies for ML-based classification.

4.1 DATA COLLECTION

The first and the primary step to train the model is Data Collection. Selecting an efficient and suitable dataset is mandatory. Data Collection involves gathering comprehensive information based on the features required. So, we got a suitable dataset with a size 200 different entries which is collected from Kaggle (Drug200.csv) [1]. The Following are the features present in the dataset: -

- Age
- Sex or Gender
- Blood Pressure (BP)
- Cholesterol
- Na to K ratio
- Drug Type

Table 4.1: Co-Relation of Numerical Variables

	Age	Na_to_K ratio
Age	1.00000	-0.063119
Na_to_K ratio	-0.063119	1.00000

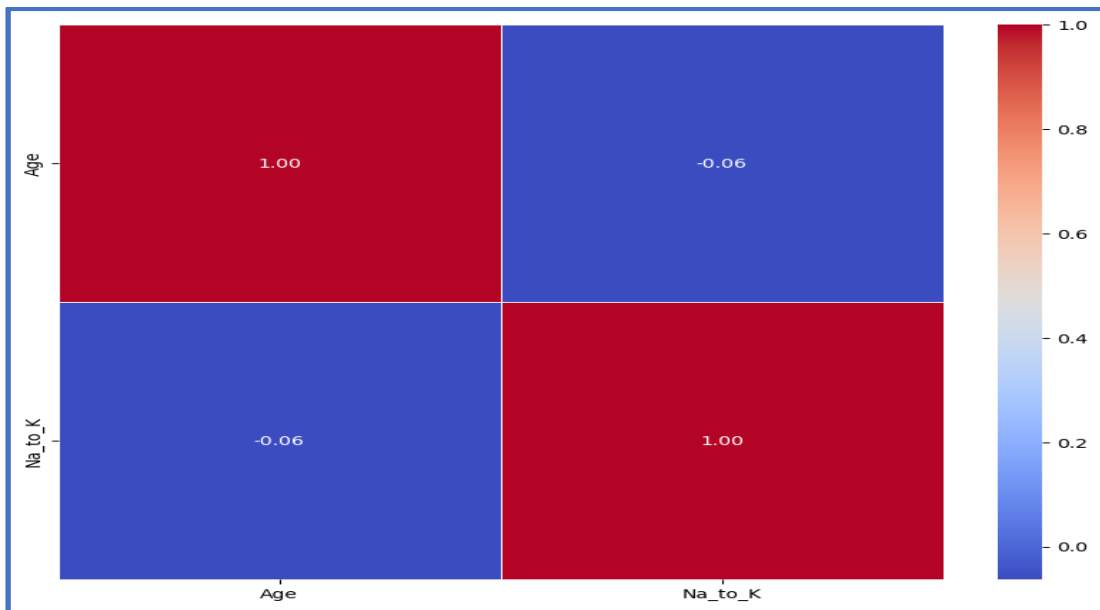


Fig. 4.1: Heatmap of Co-Relation

The following below figures is distribution graph of the features: -

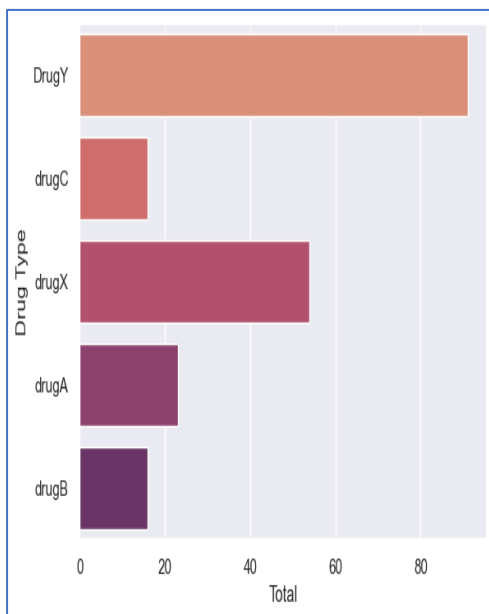


Fig.4.2a: Drug Type Distribution

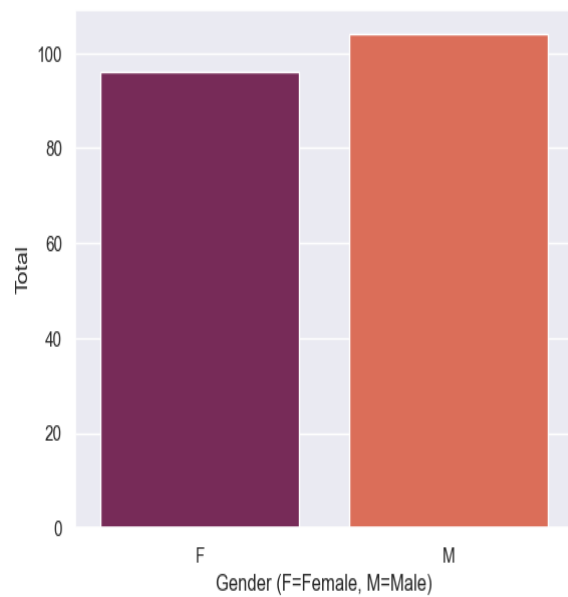


Fig. 4.2b: Gender Distribution

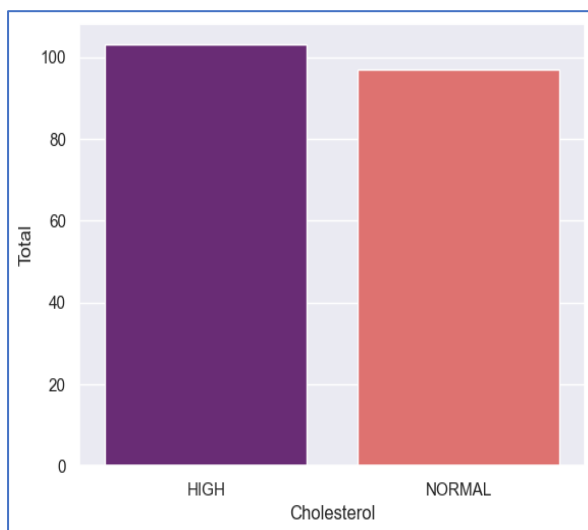


Fig. 4.2c: *Cholesterol Distribution*

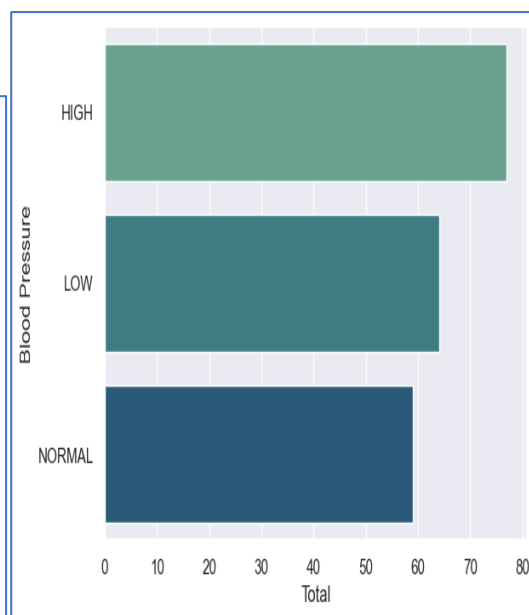


Fig. 4.2d: *Blood Pressure Distribution*

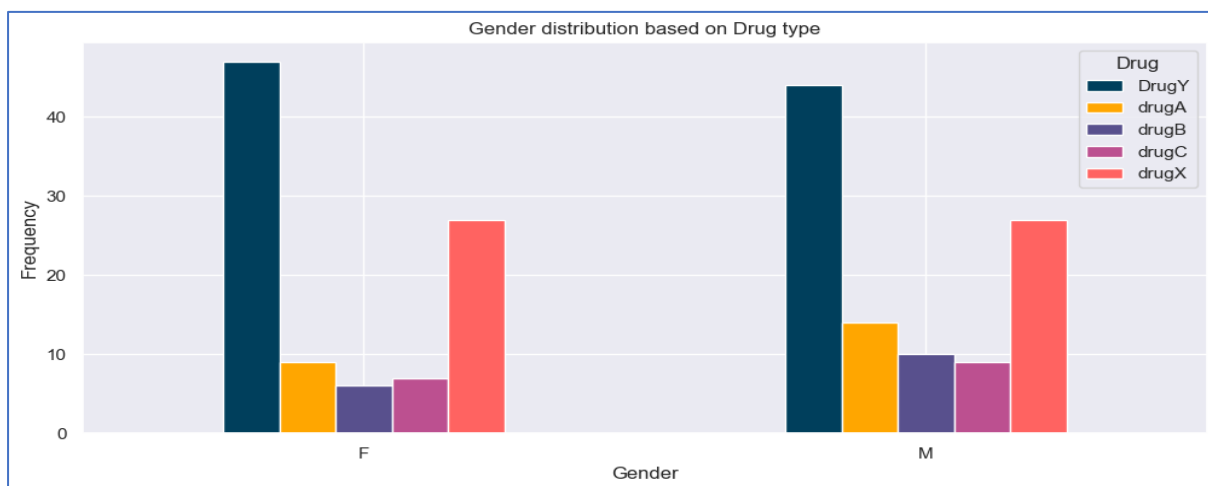


Fig. 4.2e: *Gender Distribution based on Drug Type*

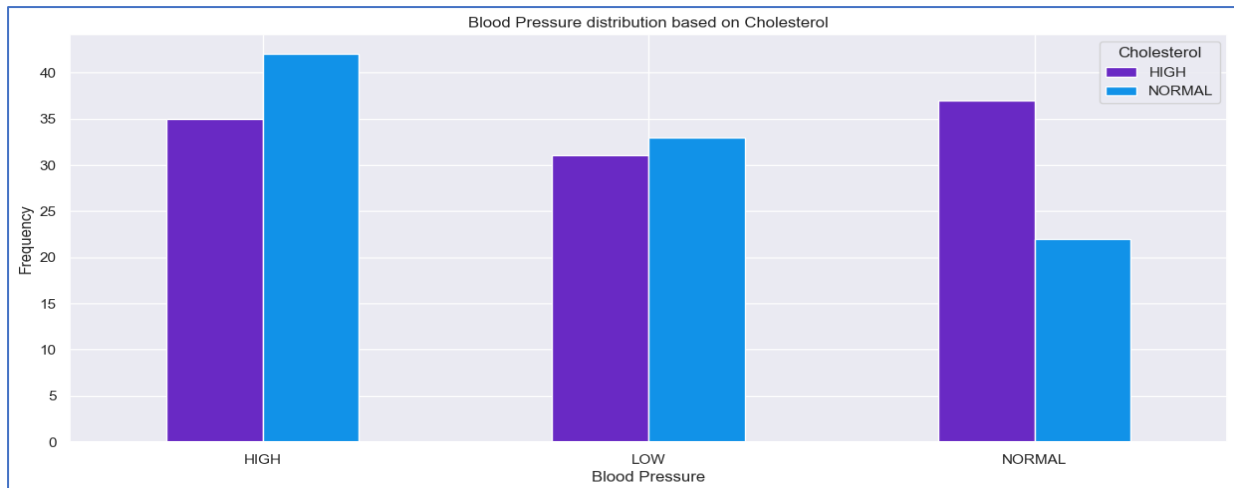


Fig. 4.2f: *Blood Pressure distribution based on Cholesterol*

4.2 DATA PREPROCESSING

Data pre-processing is a crucial step in training the model. Data pre-processing is generally purifying the data by handling the missing data and outliers appropriately & also, to normalize and scale features, encode the categorical data etc.

The drug200.csv is a perfect dataset since it has no missing data and outliers but during the training of model Data Binning process is performed to simplify data, handling non-linear relationships, to detect for outliers and to avoid over-fitting of data. and, to handle the categorical variables using hot-encoding.

The following below is the code snippet of Handling the categorical variables that includes in Data preprocessing: -

```
]: X = pd.get_dummies(X,['Age','BP','Cholesterol'])
   y = pd.get_dummies(y,['Drug'])
```

Fig. 4.2a: *Code Snippet for Handling the categorical variables*

The following below is the code snippet of Data Binning process: -

```
#Data Binning
#for age
bin_age = [0, 19, 29, 39, 49, 59, 69, 80]
category_age = ['<20s', '20s', '30s', '40s', '50s', '60s', '>60s']
df_drug['Age_binned'] = pd.cut(df_drug['Age'], bins=bin_age, labels=category_age)
df_drug = df_drug.drop(['Age'], axis = 1)
```

Fig. 4.2b: Code Snippet for Data Binning process

To overcome a challenge called over-fitting we use Synthetic Minority Over-sampling Technique (SMOTE) which is a Data preprocessing technique to avoid over-fitting of the data. Since in the dataset Drug_Y has the maximum occurrence to normalize it we use this SMOTE Technique.

The following below is the code snippet for initializing the SMOTE Technique:

-

```
: #To avoid overfitting we use SMOTE Techinque
# beacause the number of DRUG_Y is morethan other drugs it may overfit

from imblearn.over_sampling import SMOTE
X_train, y_train = SMOTE().fit_resample(X_train, y_train)
```

Fig.4.2c: Code Snippet for SMOTE Technique

The following below is the Drug Distribution after the using the SMOTE Technique: -

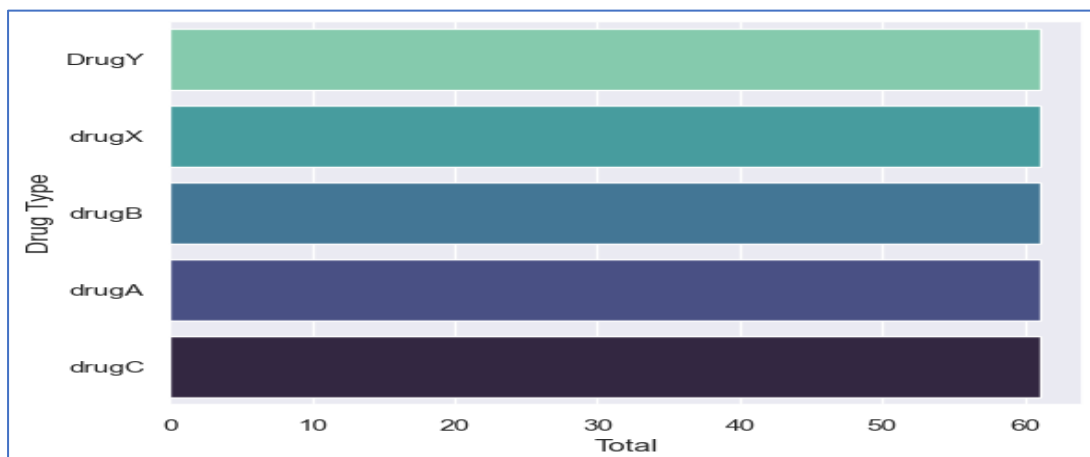


Fig.

4.2d: Drug Distribution after using the SMOTE Technique

4.3 MODEL SELECTION

It is important to select the model on basis of accuracy it produces in our dataset. To know the accuracies of the models or algorithms we have perform a comparative analysis between the models.

4.3.1 LOGISTIC REGRESSION

Logistic Regression is a type of Machine Learning Classification model. Basically, Logistic Regression is used for predicting categorical data.

The following below is the code snippet of Logistic Regression: -

```
: #Intializing the model
: #using the Logistic Regression

from sklearn.linear_model import LogisticRegression
LRclassifier = LogisticRegression(solver='liblinear', max_iter=5000)
LRclassifier.fit(X_train, y_train)

y_pred = LRclassifier.predict(X_test)

print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))

from sklearn.metrics import accuracy_score
LRAcc = accuracy_score(y_pred,y_test)
print('Logistic Regression accuracy is: {:.2f}%'.format(LRAcc*100))
```

Fig.

4.3.1a: Code snippet for Logistic Regression

The Following below is the Classification Report,Accuracy score and metrics graph of the model using Logistic Regression: -

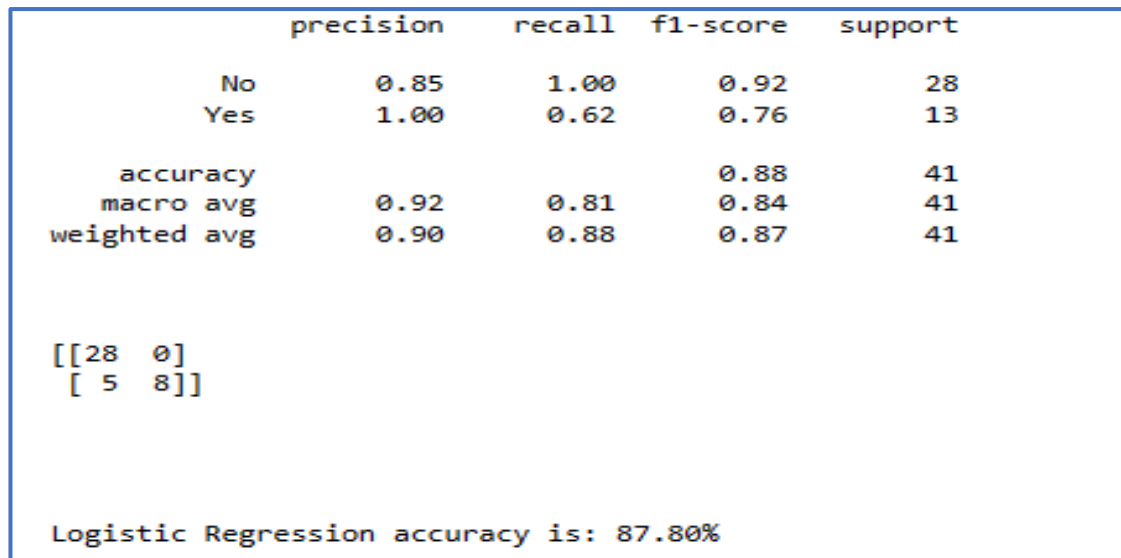


Fig.

4.3.1b: Classification report, confusion matrix and accuracy using logistic regression

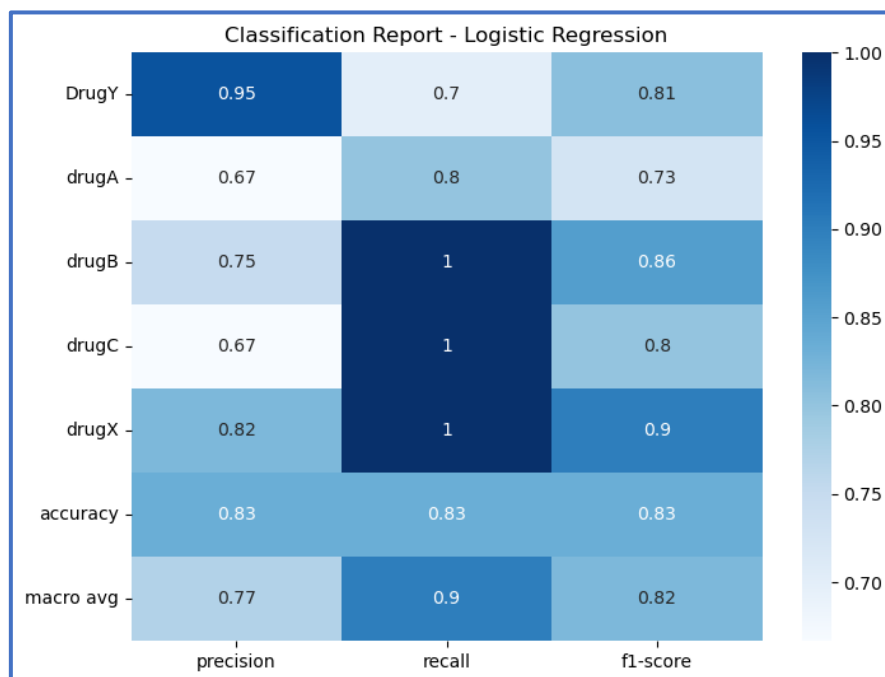


Fig. 4.3.1c: Metric graph using logistic regression

4.3.2 K NEAREST NEIGHBORS

K Nearest Neighbors is a type of Machine Learning Classification model.

The K Nearest Neighbors algorithm, also known as KNN, is a non – parametric, supervised learning classifier, which uses proximity to make classifications or prediction about the grouping of an individual data point.

We also used K Nearest Neighbors model in comparative analysis. The following below is the code snippet of K Nearest Neighbors model we used in our project:

```
#Intializing the model
#using the KNN Classifier

from sklearn.neighbors import KNeighborsClassifier
KNclassifier = KNeighborsClassifier(n_neighbors=20)
KNclassifier.fit(X_train, y_train)

y_pred = KNclassifier.predict(X_test)

print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))

from sklearn.metrics import accuracy_score
KNAcc = accuracy_score(y_pred,y_test)
print('K Neighbours accuracy is: {:.2f}%'.format(KNAcc*100))
```

Fig. 4.3.2a: Code snippet of KNN classifier

The Following below is the Classification Report,Accuracy score and metrics graph of the model using Logistic Regression: -

	precision	recall	f1-score	support
DrugY	0.86	0.60	0.71	30
drugA	0.50	0.80	0.62	5
drugB	0.33	0.33	0.33	3
drugC	0.57	1.00	0.73	4
drugX	0.76	0.89	0.82	18
accuracy			0.72	60
macro avg	0.60	0.72	0.64	60
weighted avg	0.75	0.72	0.72	60


```
[[18  3  1  3  5]
 [ 0  4  1  0  0]
 [ 1  1  1  0  0]
 [ 0  0  0  4  0]
 [ 2  0  0  0 16]]
```

K Neighbours accuracy is: 71.67%

Fig. 4.3.2b: Classification report, confusion matrix and accuracy using KNN

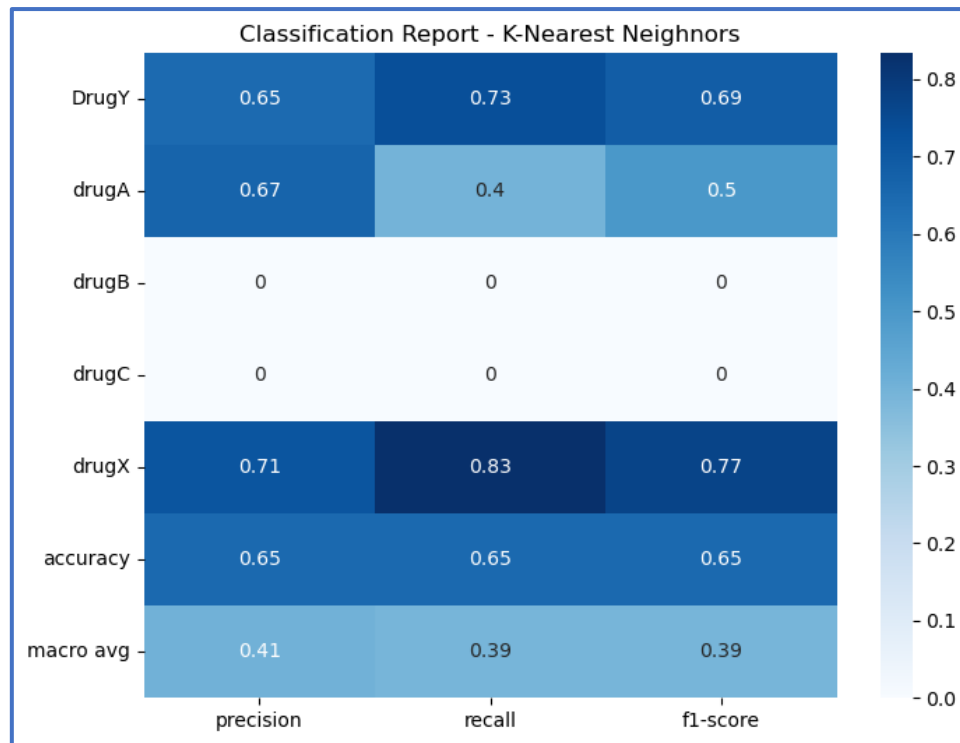


Fig. 4.3.2c: Metric graph using KNN

4.3.3 SUPPORT VECTOR MACHINE

A Support Vector Machine (SVM) is a type of supervised learning algorithm used in machine learning to solve classification and regression tasks. SVM's are particularly good at solving binary classification problems, which require classifying the elements of a data set into two groups. SVM's are effective in high dimensional spaces and can still be effective when the number of dimensions is greater than the number of samples.

We had also considered Support vector machine in comparative analysis. The following below is the code snippet of Support Vector Machine: -

```
8]: #Initializing the model
    #using the Support Vector Machine

    from sklearn.svm import SVC
    SVCclassifier = SVC(kernel='linear', max_iter=251)
    SVCclassifier.fit(X_train, y_train)

    y_pred = SVCclassifier.predict(X_test)

    print(classification_report(y_test, y_pred))
    print(confusion_matrix(y_test, y_pred))

    from sklearn.metrics import accuracy_score
    SVCAcc = accuracy_score(y_pred, y_test)
    print('SVC accuracy is: {:.2f}%'.format(SVCAcc*100))
```

Fig. 4.3.3a: Code snippet of using SVM

The Following below is the Classification Report, Accuracy score and metrics graph of the model using Logistic Regression: -

	precision	recall	f1-score	support
DrugY	0.85	0.73	0.79	30
drugA	0.67	0.80	0.73	5
drugB	0.00	0.00	0.00	3
drugC	0.67	1.00	0.80	4
drugX	0.82	1.00	0.90	18
accuracy			0.80	60
macro avg	0.60	0.71	0.64	60
weighted avg	0.77	0.80	0.78	60


```

[[22  2  0  2  4]
 [ 1  4  0  0  0]
 [ 3  0  0  0  0]
 [ 0  0  0  4  0]
 [ 0  0  0  0 18]]

```

SVC accuracy is: 80.00%

Fig. 4.3.3b: Classification report, confusion matrix and accuracy using SVM

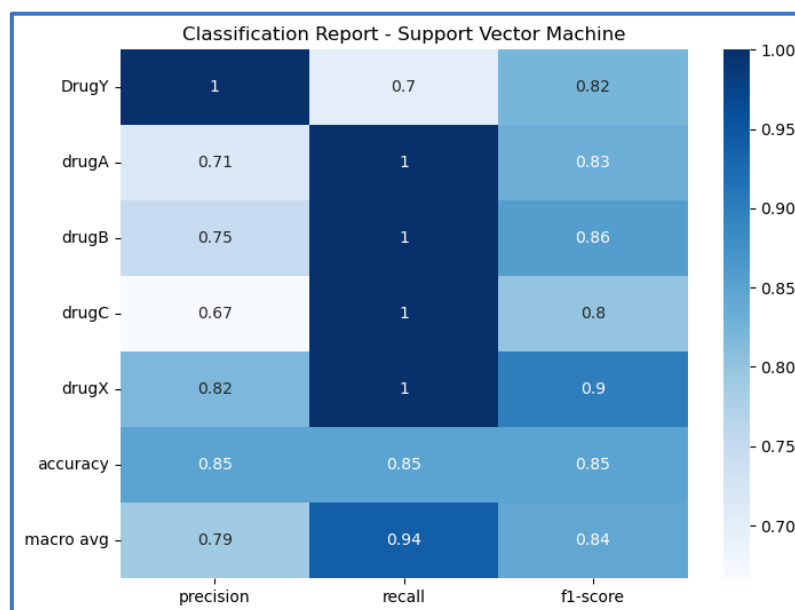


Fig. 4.3.3c: Metric graph using SVM

4.3.4 NAIVE BAYES

A Naive Bayes classifier is a supervised machine learning algorithm that is based on Bayes' theorem. It is a collection of algorithms that share the principle that every pair of features being classified is independent of each other. Naïve Bayes classifier is used for classification.

We had also considered Naive Bayes classifier in comparative analysis. The following below is the code snippet we used: -


```

#Intializing the model
#using the Naive Bayes Classifier

from sklearn.naive_bayes import CategoricalNB
NBclassifier1 = CategoricalNB()
NBclassifier1.fit(X_train, y_train)

y_pred = NBclassifier1.predict(X_test)

print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))

from sklearn.metrics import accuracy_score
NBacc1 = accuracy_score(y_pred,y_test)
print('Naive Bayes accuracy is: {:.2f}%'.format(NBacc1*100))

```

Fig. 4.3.4a: Code snippet of using Naïve Bayes Classifier

The Following below is the Classification Report, Accuracy score and metrics graph of the model using Logistic Regression: -

	precision	recall	f1-score	support
DrugY	0.91	0.67	0.77	30
drugA	0.62	1.00	0.77	5
drugB	0.75	1.00	0.86	3
drugC	0.50	0.50	0.50	4
drugX	0.73	0.89	0.80	18
accuracy			0.77	60
macro avg	0.70	0.81	0.74	60
weighted avg	0.80	0.77	0.76	60


```

[[20  3  1  2  4]
 [ 0  5  0  0  0]
 [ 0  0  3  0  0]
 [ 0  0  0  2  2]
 [ 2  0  0  0 16]]

```

Naive Bayes accuracy is: 76.67%

Fig. 4.3.4b: Classification report, confusion matrix and accuracy using NB

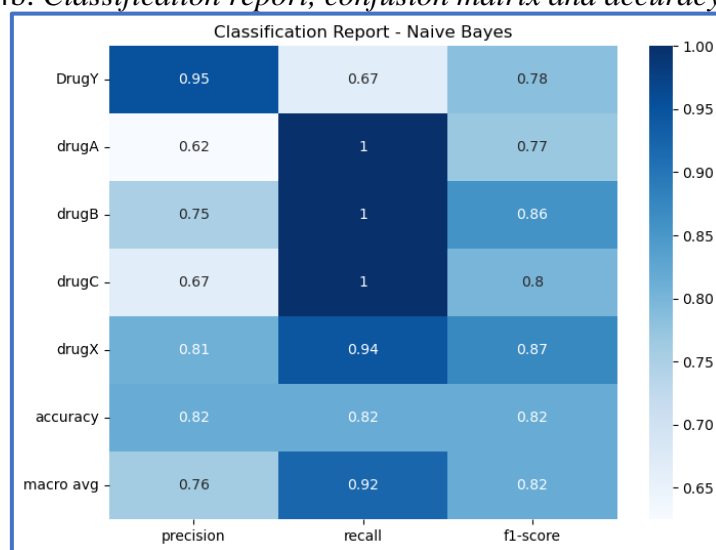


Fig. 4.3.4c: Metric graph using Naïve Bayes

4.3.5 RANDOM FOREST

Random forest is commonly used Machine Learning Supervised algorithm that is used for solving both Regression and Classification. Basically, Random Forest combines the results of multiple decision trees to produce a single result. It uses Ensemble learning method training the model.

We had considered Random Forest classifier into the comparative analysis. The following below is the code snippet of Random Forest classifier we used: -

```
#Intializing the model
#using the RandomForestClassifier

from sklearn.ensemble import RandomForestClassifier

RFclassifier = RandomForestClassifier(max_leaf_nodes=30)
RFclassifier.fit(X_train, y_train)

y_pred = RFclassifier.predict(X_test)

print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))

from sklearn.metrics import accuracy_score
RFAcc = accuracy_score(y_pred,y_test)
print('Random Forest accuracy is: {:.2f}%'.format(RFAcc*100))
```

Fig. 4.3.5a: Code snippet of using Random Forest classifier

The Following below is the Classification Report, Accuracy score and metrics graph of the model using Logistic Regression: -

	precision	recall	f1-score	support
DrugY	1.00	0.63	0.78	30
drugA	0.62	1.00	0.77	5
drugB	0.60	1.00	0.75	3
drugC	0.67	1.00	0.80	4
drugX	0.82	1.00	0.90	18
accuracy			0.82	60
macro avg	0.74	0.93	0.80	60
weighted avg	0.87	0.82	0.81	60


```
[[19  3  2  2  4]
 [ 0  5  0  0  0]
 [ 0  0  3  0  0]
 [ 0  0  0  4  0]
 [ 0  0  0  0 18]]
```

Random Forest accuracy is: 81.67%

Fig. 4.3.5b: Classification report, confusion matrix and accuracy using Random Forest

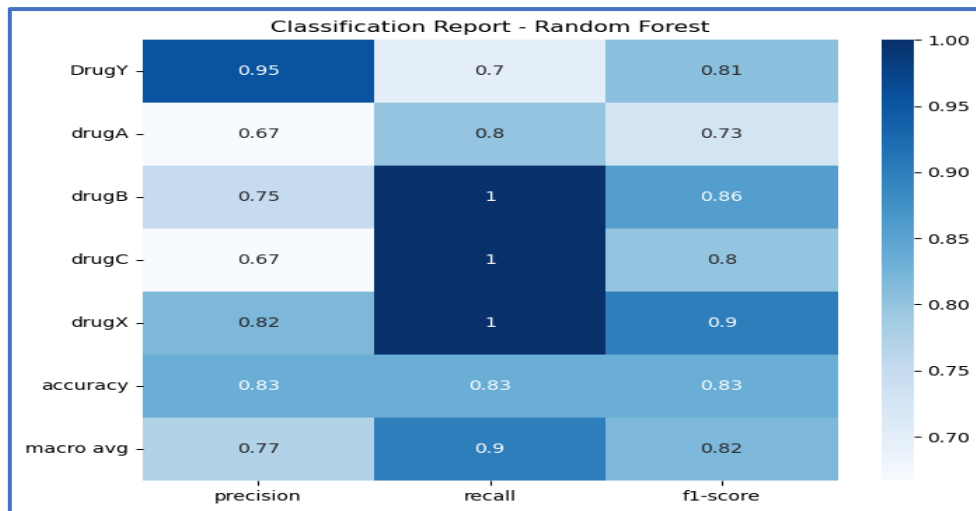


Fig. 4.3.5c: Metric graph using Random Forest

4.3.6 ARTIFICIAL NEURAL NETWORK

Artificial neural networks (ANNs) are a type of machine learning process that uses interconnected nodes, or neurons, to process data. ANNs are also known as simulated neural networks (SNNs). ANNs are inspired by the human brain, where each circular node represents an artificial neuron and an arrow represents a connection from the output of one artificial neuron to the input of another.

We had considered ANN into the comparative analysis. The following code snippet below about ANN we used in the model:

```
model_2 = Sequential([
    normalize,
    Flatten(input_shape=(9,)),
    Dense(32, activation='tanh'),
    Dropout(0.5),
    Dense(5, activation='sigmoid'),
])

model_2.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

from tensorflow.keras.callbacks import EarlyStopping
early_stop = EarlyStopping(monitor='val_loss', mode='min', patience=10, restore_best_weights=True)

model_2.fit(x=X_train,
            y=y_train,
            epochs=1000,
            batch_size=10,
            validation_data=(X_test, y_test),
            callbacks=[early_stop])
```

Fig. 4.3.6a: Code snippet of using ANN

The following below is the Classification Report, Confusion Matrix, and Accuracy score: -

	precision	recall	f1-score	support
0	1.00	0.97	0.98	29
1	1.00	0.75	0.86	8
2	0.57	1.00	0.73	4
3	1.00	1.00	1.00	6
4	1.00	1.00	1.00	19
accuracy			0.95	66
macro avg	0.91	0.94	0.91	66
weighted avg	0.97	0.95	0.96	66

Accuracy: 95.45454382896423 %

Fig. 4.3.6b: *Classification report, confusion matrix and accuracy using ANN*

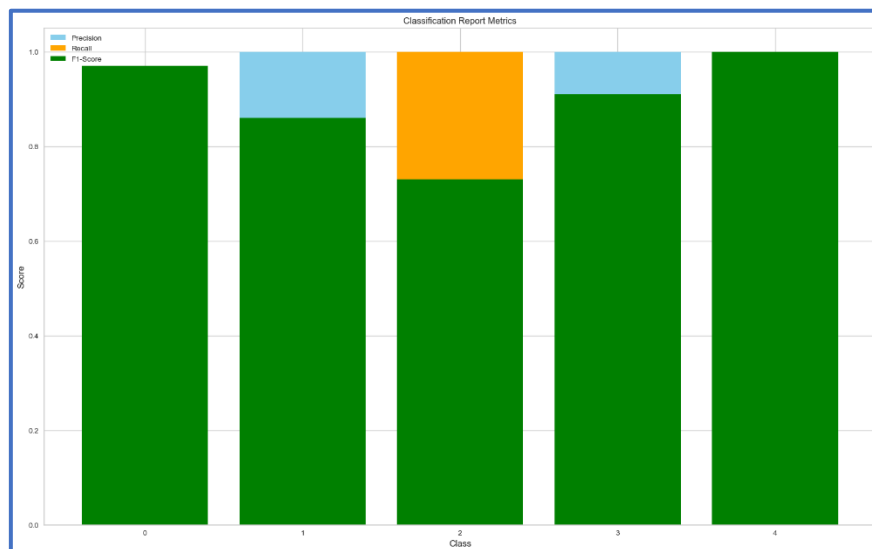


Fig. 4.3.6c: *Metric graph using ANN*

4.3.7 DECISION TREE

Decision tree is a machine learning model which is applicable for both regression and classification problems. Decision tree is a supervised learning algorithm that structures decisions based on input data. It is a flowchart-like tree structure. The following are the parts of a decision tree: -

- Root node: The node that contains the complete dataset
- Decision nodes: Nodes that result from splitting root nodes
- Leaf nodes: Nodes where further splitting is not possible

The following below is the code snippet representing the decision tree model: -

```

from sklearn.tree import DecisionTreeClassifier
DTclassifier = DecisionTreeClassifier(max_leaf_nodes=20)
DTclassifier.fit(X_train, y_train)

y_pred = DTclassifier.predict(X_test)

print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))

from sklearn.metrics import accuracy_score
DTAcc = accuracy_score(y_pred, y_test)
print('Decision Tree accuracy is: {:.2f}%'.format(DTAcc*100))

```

Fig 4.3.7a Code snippet representing Decision Tree

The following below image (Fig 4.3.7.2) represents the classification report, confusion matrix and accuracy score of the model Decision Tree: -

	precision	recall	f1-score	support
DrugY	0.95	0.63	0.76	30
drugA	0.50	0.80	0.62	5
drugB	0.75	1.00	0.86	3
drugC	0.67	1.00	0.80	4
drugX	0.82	1.00	0.90	18
accuracy			0.80	60
macro avg	0.74	0.89	0.79	60
weighted avg	0.84	0.80	0.80	60
[[19 4 1 2 4]				
[1 4 0 0 0]				
[0 0 3 0 0]				
[0 0 0 4 0]				
[0 0 0 0 18]]				
Decision Tree accuracy is: 80.00%				

Fig 4.3.7b Classification report, confusion matrix, Accuracy score for Decision Tree

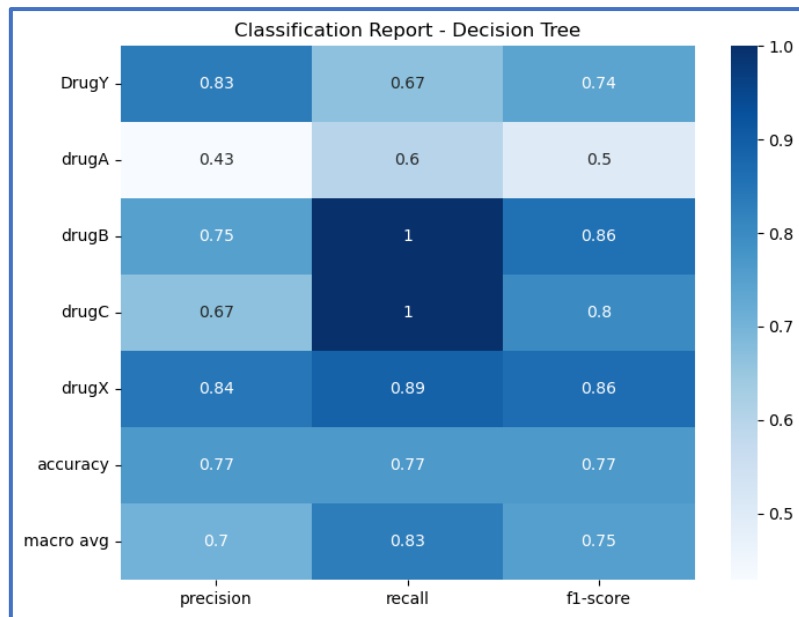


Fig. 4.3.7c: Metric graph using Decision Tree

CHAPTER -5

RESULT

5.1 COMPARITIVE ANALYSIS

To create a machine learning model or system model selection plays a major role. It includes selecting a perfect that is suitable with the problem statement, type of data dealing with and at last the dataset. So, to select the model that gives higher accuracy and as well as better performance. As we trained various models on the dataset, we will compare them based on accuracy, performance, scalability, and relevance.

The table below represents the name of the model and its respective accuracies in high to low order: -

S.no	Model Name	Accuracy (%)
1	Artificial Neural Network (ANN)	95.45 %
2	Logistic Regression	87.80%
3	Random Forest (RF)	81.67%
4	Support Vector Machine (SVM)	80.00%
5	Decision Tree	80.00%
6	Naïve Bayes (NB)	76.67%
7	K-Nearest Neighbors (KNN)	71.67%

NOTE: - After considering all the different hyper parameters in training the model the above table is summarized and it is the maximum accuracies that are considered after changing the parameters.

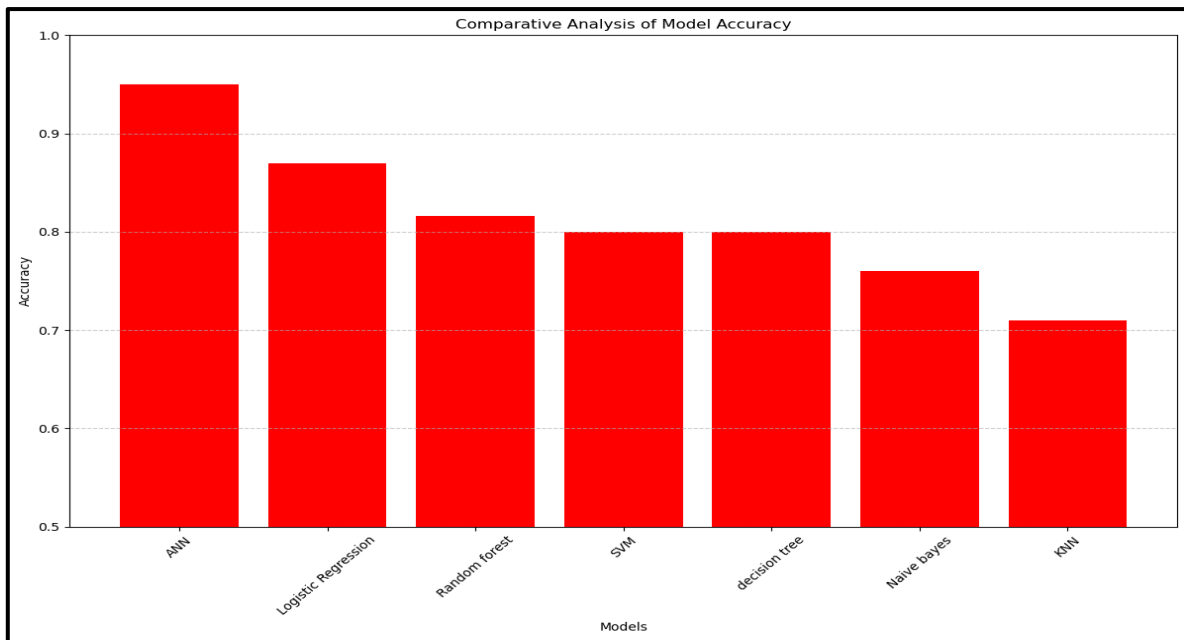


Fig.5.1 Comparative Analysis Graph

The above table and graph (Fig.5.1.1) summarize the comparative analysis where ANN and Logistic Regression models are having the high accuracy and better performance than the other models. So, they are been finalised after the analysis, comparison, and discussion but at last there will be again comparison in predicting the drug based on classification so that we can understand how better it will work in real – time application and at the time implementation.

5.2 FINAL SELECTION

Model is selected under the conditions of high accuracy and better performance. This comparative analysis had helped in selecting the high accuracy best model.

Yes, ANN has the high accuracy and better performance compared to other models, but it is working very well on the dataset at the time of training and testing. ANN is considered best in classification compared to other models but in prediction it is not working accurately. This will be a huge problem in future. So, we decided to consider the second-best model Logistic regression model where it has high accuracy which very near to ANN's accuracy and the best part is that it is working very well in classification and in predicting the accurate drug based on the input. After a huge analysis, comparison, and discussion we had decided to consider and use the Logistic Regression model for the project. This is conclusion in selecting the perfect model that works on both cases with high accuracy and performance and other metrics and Logistic Regression model is easily syncing with the gateway and server

connection(app.py) and with user interface and producing the accurate result that can be easily displayed on the user interface. So, the finalized model that will be used is Logistic Regression model for the project based on the above analysis and discussion.

CHAPTER -6

IMPLEMENTATION

6.1 FRONT-END or USER INTERFACE

It is common that mostly every user interface will be developed using HTML and CSS. We have also used the same pattern as creating the Front-End using HTML and CSS. The below image (Fig.6.1.1) represents our project's Front-End: -

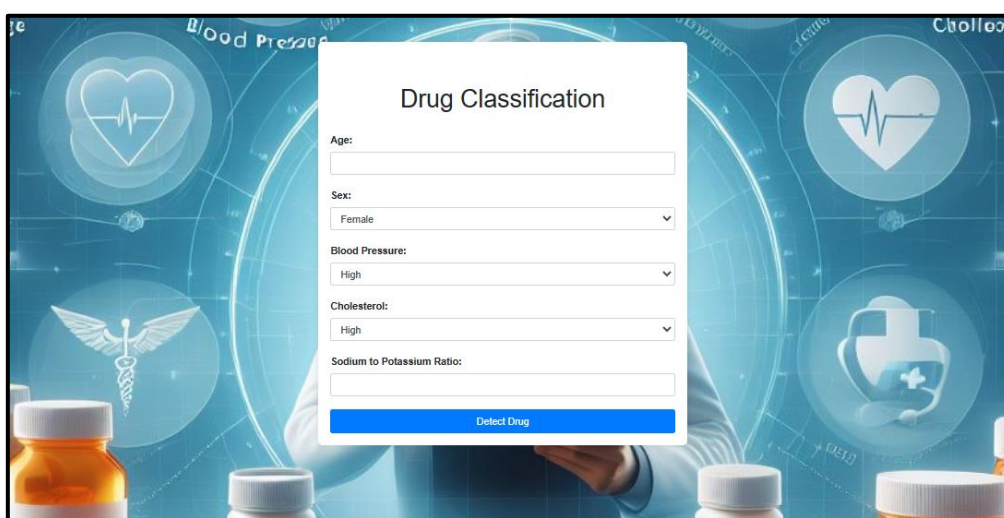
The image shows a web application interface for 'Drug Classification'. The background is a blue-toned medical theme with icons for a heart, a caduceus, and a pill bottle. A white form is centered on the screen. The form has the title 'Drug Classification' at the top. Below the title are several input fields: 'Age:' with a text box, 'Sex:' with a dropdown menu showing 'Female', 'Blood Pressure:' with a dropdown menu showing 'High', 'Cholesterol:' with a dropdown menu showing 'High', and 'Sodium to Potassium Ratio:' with a text box. At the bottom of the form is a blue button labeled 'Detect Drug'.

Fig.6.1 Front-End of the project

Basically, the front-end consists of a background image that is related to the theme and apart from this it consists of a HTML form that takes the input from the user and is sent to the model through the server and gets the result based on the input and displayed on the web page. And this form consisting of the button that detects the drug based on the input given, this button will trigger the server to take the input from the form and give the result that should be displayed on the page

6.2 BACK-END

The Back-end of the project is built using the python library called Flask. Since the model is trained in python. It will become easy to integrate the model to the server. That is the main reason of using FLASK to build the server and gateway for taking input from user and giving output to the user.

The below images (Fig 6.2.1 and Fig.6.2.2) represents code of the Back-End of the project: -

```

from flask import Flask, render_template, request
import pandas as pd
import pickle

app = Flask(__name__)

# Load the trained model
model = pickle.load(open('drug.pkl', 'rb'))

# Define the feature names used during training
feature_names = ['Sex_F', 'Sex_M', 'BP_HIGH', 'BP_LOW', 'BP_NORMAL', 'Cholesterol_HIGH', 'Cholesterol_NORMAL', 'Age_binned_

def detect_drug(age, sex, bp, cholesterol, na_to_k):
    # Create a DataFrame with input data
    input_data = {
        'Sex': [sex],
        'BP': [bp],
        'Cholesterol': [cholesterol],
        'Age_binned': [age], # Assuming age is binned
        'Na_to_K_binned': [na_to_k] # Assuming Na_to_K is binned
    }
    input_df = pd.DataFrame(input_data)

```

Fig.6.2a Code of Back-End server

```

    input_df = pd.get_dummies(input_df)

    # Make sure input features match the columns used during training
    # Use the same order of columns as used during training
    input_df = input_df.reindex(columns=feature_names, fill_value=0)

    # Make predictions
    predicted_drug = model.predict(input_df)

    return predicted_drug[0]

@app.route('/', methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        # Get form data
        age = request.form['age']
        sex = request.form['sex']
        bp = request.form['bp']
        cholesterol = request.form['cholesterol']
        na_to_k = request.form['na_to_k']

        # Call detect_drug function
        predicted_drug = detect_drug(age, sex, bp, cholesterol, na_to_k)

        return render_template('index.html', prediction=predicted_drug)

    return render_template('index.html', prediction=None)

if __name__ == '__main__':
    app.run(debug=True)

```

Fig.6.2b Code of Back- End Server

6.3 WORKING DEMONSTRATION OF THE PROJECT

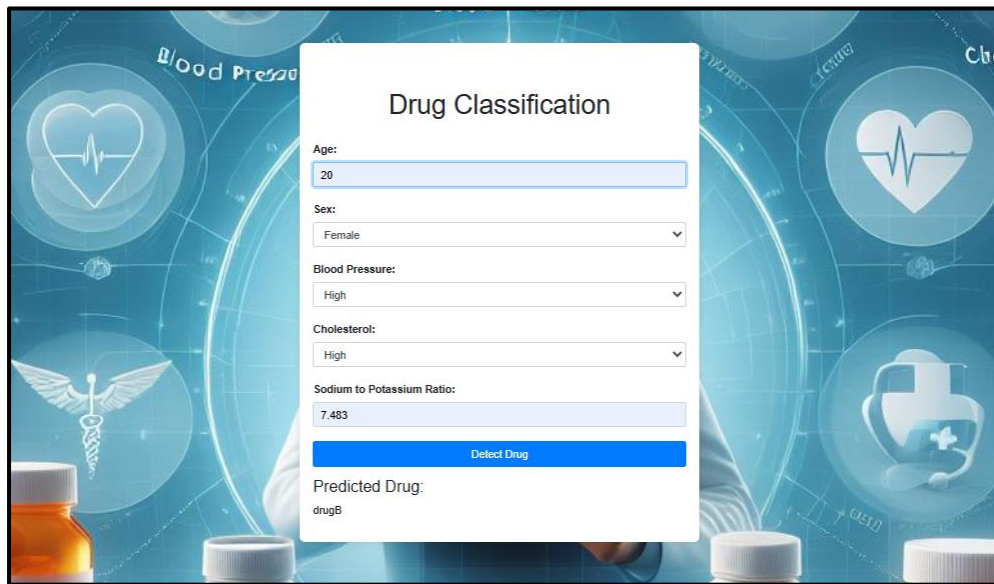


Fig.6.3a Working Demonstration of the project

The above figure (Fig.6.3a) represents the working stage of the model which predicts and displayed the drug based on the input given by the user. Finally, the system is providing the accurate result.

	Sex	BP	Cholesterol	Age_binned	Na_to_K_binned	DrugType
0	Male	Low	High	20s	<10	drugX
1	Female	Normal	High	20s	10-20	drugX
2	Male	Low	High	40s	20-30	DrugY
3	Male	High	Normal	20s	>30s	DrugY
4	Male	Normal	High	20s	20-30	DrugY

Fig.6.3b Classification Analysis made by the model

he image (Fig.6.3b) that consists table which represents the classification analysis made by the model. It concisely showing the Classification analysis based on this classification the model should predict and classify the drug on the different input given by the user which is not included in the dataset. In other words, it is representing the output of the model that is desired.

CHAPTER – 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The main reason behind developing “**Medical Drug Classification**” system is to build awareness among the people that they will no longer consult the doctor for every small and medium health hazard or disease.

In conclusion, the development of a machine learning-based approach for medical drug classification holds significant promise in advancing personalized medicine and improving patient care outcomes. Through the integration of patient attributes such as age, gender, blood pressure, cholesterol levels, and the natriuretic peptide ratio, our model demonstrates the potential to tailor drug prescriptions to individual patient profiles with greater accuracy and efficacy. By leveraging techniques such as logistic regression, we have shown how predictive modelling can assist healthcare practitioners in making informed decisions regarding drug selection, thereby optimizing treatment strategies, and minimizing adverse effects. The results obtained underscore the importance of harnessing computational methods to augment traditional approaches in drug classification, paving the way for a more precise and personalized healthcare delivery system. As we continue to refine and expand upon these methodologies, future advancements in drug classification hold the promise of revolutionizing patient care and improving overall public health outcomes.

7.2 FUTURE SCOPE

The future scope of this project encompasses several avenues for further exploration and enhancement:

- **Incorporation of Additional Patient Data:** -Expand the model to incorporate a broader range of patient data, including genetic information, lifestyle factors, comorbidities, and medication history. Integrating these additional variables could improve the accuracy and specificity of drug classification predictions, leading to more personalized treatment recommendations.
- **Integration of Advanced Machine Learning Techniques:** -Explore more advanced machine learning algorithms and techniques, such as deep learning, ensemble methods, and reinforcement learning, to further enhance the predictive performance of the model. These approaches may uncover complex patterns and relationships within

the data that traditional methods might overlook.

- **Real-time Decision Support System:** - Develop a real-time decision support system that can provide healthcare practitioners with timely and actionable drug classification recommendations at the point of care. This system could streamline clinical workflows, improve treatment decision-making, and ultimately enhance patient outcomes.
- **Validation in Clinical Settings:** -Conduct extensive validation studies in clinical settings to assess the real-world performance and utility of the drug classification model. Collaborate with healthcare institutions and practitioners to gather feedback, refine the model, and ensure its practicality and effectiveness in diverse patient populations.
- **Integration with Electronic Health Records (EHRs):** - Integrate the drug classification model with electronic health record (EHR) systems to enable seamless access to patient data and facilitate automated decision-making processes. This integration could improve efficiency, reduce errors, and promote the adoption of personalized medicine practices in healthcare settings.

By pursuing these future directions, this project has the potential to advance the field of personalized medicine, improve healthcare delivery, and ultimately, enhance the quality of patient care on a global scale.

APPENDIX A: TOOLS AND TECHNOLOGIES

- **PYTHON V3:** The Python language comes with many libraries and frameworks that make coding easy. This also saves a significant amount of time.
- **JUPYTER NOTEBOOK:** The Jupyter Notebook is an open-source web application that you can use to create and share documents that contain live code, equations, visualizations, and text.
- **NUMPY:** NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices.
- **WINDOWS 11:** Windows 11 was used as the operating system.
- **FLASK:** Flask is a microframework, which means it is a small and lightweight framework that does not require tools or libraries. It is based on the WSGI toolkit and Jinja2 template engine. Flask is considered a microframework because it does not require tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

REFERENCES

- [1] <https://www.kaggle.com/datasets/prathamtripathi/drug-classification>
- [2] <https://www.kaggle.com/code/caesarmario/drug-classification-w-various-ml-models>
- [3] <https://www.kaggle.com/code/mohamedzayton/drug-classification-rf-nn>
- [4] <https://doi.org/10.22214/ijraset.2022.43609>
- [5] <https://www.kaggle.com/code/gorkemgunay/drug-classification-with-different-algorithms>