Project Report On

Classification Of Drugs Using Machine Learning

INTRODUCTION

1.1 Overview

In the field of medicine, accurate and efficient drug classification is essential for effective patient treatment. The current drug classification system relies mainly on manual methods, which can lead to errors and inconsistencies. The goal of this project is to develop a machine learning model that can accurately classify drugs based on a patient's age, gender, and other factors. The model will provide a more precise and systematic approach to drug classification, ultimately improving patient outcomes and reducing the risk of adverse events. This project aims to demonstrate the feasibility and potential benefits of using machine learning techniques in drug classification.
The project involved collecting data from various sources. This data was then pre-processed and subjected to feature selection to determine the most relevant information for the classification of drugs. A machine learning model was selected and trained using the processed data, and the performance of the model was evaluated using various metrics.

The results of the project showed that the machine learning model was able to accurately classify drugs based on a patient's age, gender, and other factors with a high level of accuracy. The model demonstrated significant improvement over traditional manual methods of drug classification, offering a more precise and systematic approach to the task.

1.2 Purpose

The purpose of this project is to develop a machine learning model for the classification of drugs based on a patient's age, gender, and other relevant factors. The project aims to demonstrate the feasibility and potential benefits of using machine learning in this context, and to provide insights and recommendations for future research and development in the field.

The report will provide a comprehensive overview of the project, including its background, methodology, results, and conclusions. The goal is to communicate the key aspects of the project and its outcomes to stakeholders and to inform future work in this area. The report will also serve as a reference for those involved in the project and for others who may be interested in this area of research

2.LITERATURE SURVEY

2.1 Existing Problem

- Deep learning models:

  1.Convolutional Neural Networks (CNN).

  2.Recurrent Neural Networks (RNN).

  3.Boltzmann machine.

  4.Autoencoders etc.

- Classification:

  1.The K-Nearest Neighbours algorithm

  2.Decision Tree

  3.Support Vector Machines

  4.Naive Bayes

- Regression:

  1.Linear Regression

  2.Lasso Regression

  3.Ridge Regression

  4.Support Vector Regression (SVR)

  5.Ensemble Regression

- Clustering:

  1.K means

  2.K means++

  3.K medoids

  4.Agglomerative clustering

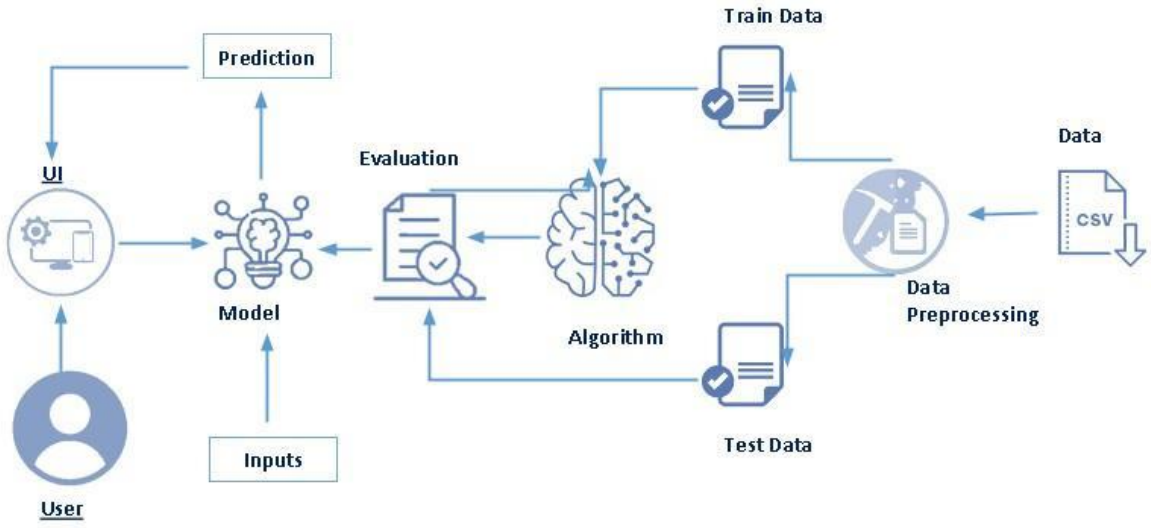
  5.DBSCAN

2.2 Proposed Solution

We used classification algorithms such as Decision tree, Random Forest, KNN, and xgboost. From the four models used, random forest and decision tree is performing well. Both models have 97% accuracy. Even confusion matrix also has same results. Training time of decision tree is faster than random forest. In such case we have to select decision tree model (time saving & cost wise profitable). But here random forest is selected and evaluated with cross validation. Additionally, we can tune the model with hyper parameter tuning techniques.

3. THEORETICAL ANALYSIS

3.1 Block Diagram:



3.2 Hardware/Software Designing

1.Software Requirements

1.Downloading of Anaconda Navigator

2.Downloading of python packages like

a. NumPy Package

b. Pandas

c. librosa

d. Tensor Flow

e. Matplotlib

f. scikit-learn

g. Flask

h. pyhton_speech_features

I. mfcc

j. from python_speech_features import mfcc

k. import sklearn.model_selection

l. from sklearn.model_selection import train_test_split

m. import scipy.io.wavfile as wav

n. import os

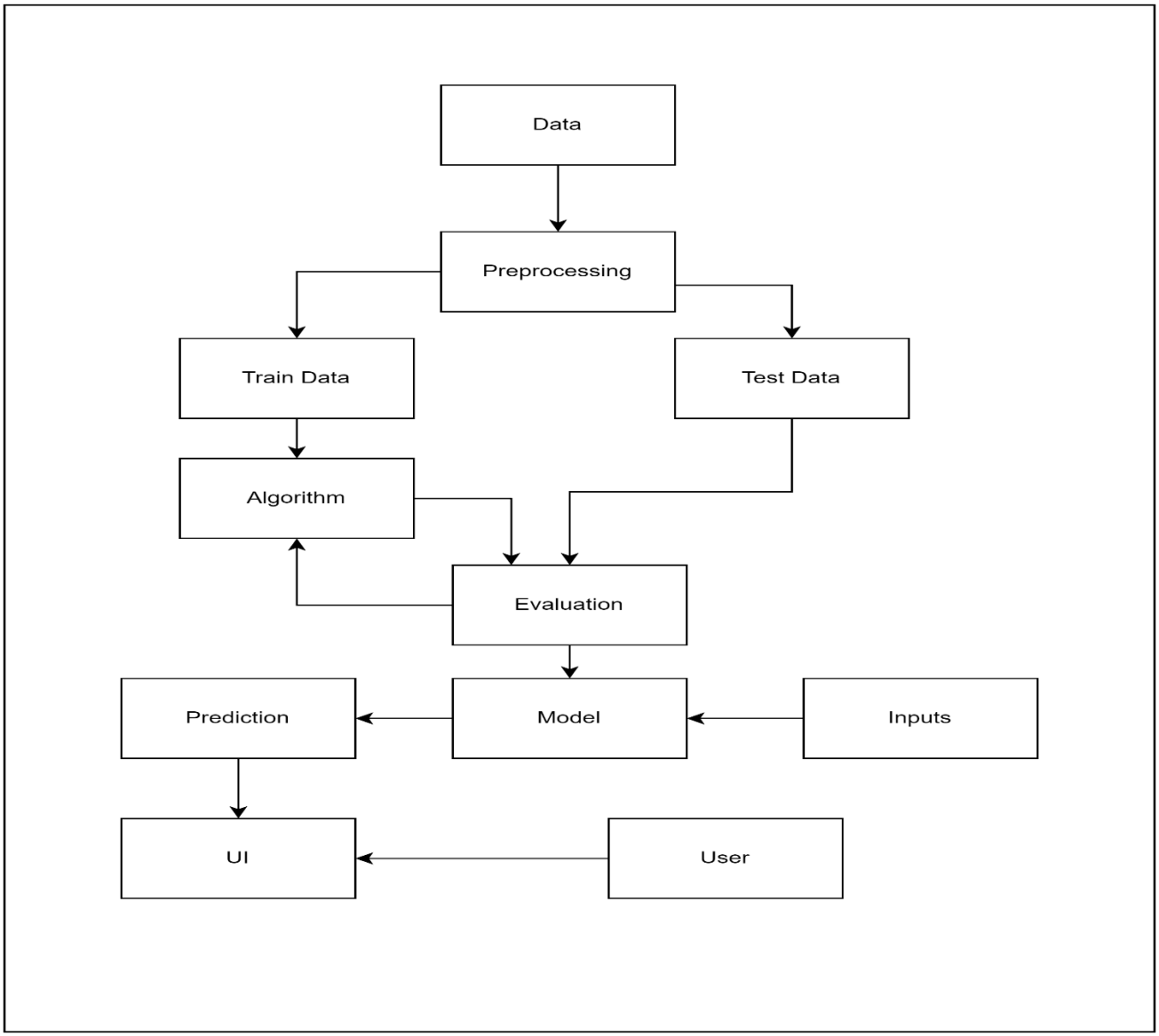
o. import pickle

p. import operator

4.EXPERIMENTAL INVESTIGATION

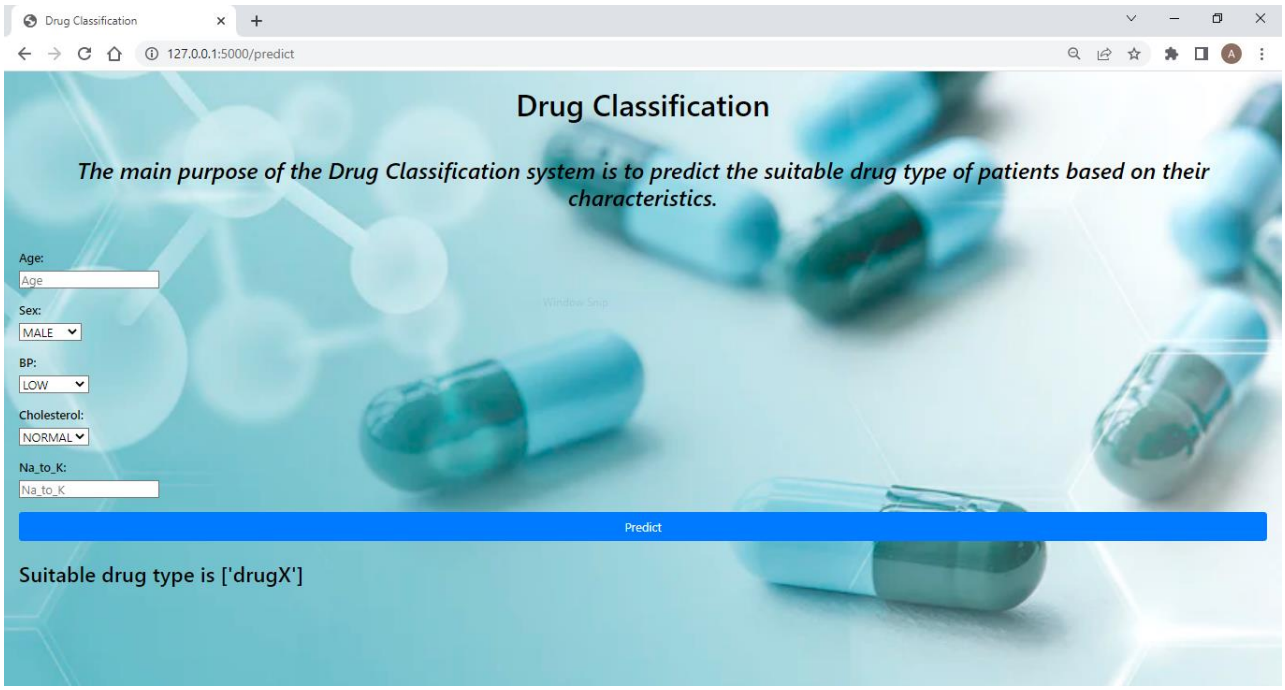
A dataset containing five attributes that were decisive in choosing the drug was collected. The attributes were Age, Sex, BP, Cholesterol and Na_to_K. There was 6 different types of drugs which were named DrugA, DrygB, DrugC, DrugX, DrugY and DrugZ to keep their details confidential. The dataset contained 200 records.

5.FLOWCHART


Data
Preprocessing
Train Data
Test Data
Algorithm
Evaluation
Prediction
Model
Inputs
UI
User

6.RESULT

```
In [2]: runfile('D:/Drug classification/flask/app.py', wdir='D:/Drug classification/flask')
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```



7.ADVANTAGES AND DISADVANTAGES

ADVANTAGES

There are several advantages of using machine learning for drug classification:

- Improved accuracy: Machine learning algorithms can handle large amounts of data and learn from it to make accurate predictions. This can result in a more accurate classification of drugs than could be achieved by manual methods.

- Increased efficiency: Machine learning models can classify drugs much faster and more efficiently than manual methods, especially when dealing with large datasets.

- Automation: Machine learning models can automate the classification process, reducing the need for manual input and freeing up time for other tasks.

- Discovery of new insights: Machine learning can help uncover patterns and relationships in the data that might not be apparent through manual methods. This can lead to the discovery of new insights into the relationships between drugs and their features, which can inform future research.

- Better decision making: Machine learning models can provide healthcare professionals and researchers with more accurate and reliable information about drugs, which can help them make better-informed decisions about treatment and research.

- Reduced cost: By automating the drug classification process, machine learning can help reduce the cost of manual labor and improve efficiency, leading to cost savings in the long run.

In conclusion, machine learning can bring many benefits to the field of drug classification, and is becoming an increasingly important tool for researchers and healthcare professionals

DISADVANTAGES

- Despite the many advantages of using machine learning for drug classification, there are also some potential disadvantages to consider:
- Bias: Machine learning models can learn and replicate biases in the training data, leading to biased predictions. This can be particularly problematic in sensitive areas such as healthcare, where biased predictions can have serious consequences.
- Lack of interpretability: Many machine learning models, particularly deep learning models, can be difficult to interpret, making it challenging to understand why they are making specific predictions. This can make it difficult to identify and correct errors in the model.
- Overfitting: Machine learning models can sometimes overfit to the training data, leading to poor performance on new, unseen data. This can be addressed through techniques such as cross-validation, but it still remains a challenge for many practitioners.
- Data quality: The quality of the input data can greatly impact the performance of machine learning models. If the data is noisy, incomplete, or biased, this can result in poor model performance.
- Complexity: Machine learning models can be complex, requiring a good understanding of both the domain and the machine learning algorithms to implement and interpret them. This can make them difficult to use for non-experts.
- High computational cost: Some machine learning models, particularly deep learning models, can require a large amount of computational resources, which can make them impractical for use in resource-constrained environments.

8.APPLICATIONS

There are several potential applications of machine learning in drug classification, including:

- Drug discovery: Machine learning can be used to help identify new drugs and their potential therapeutic uses, by analyzing large amounts of chemical and biological data.
- Drug repositioning: Machine learning can be used to identify new uses for existing drugs, by analyzing their mechanism of action, side effects, and other relevant information.
- Drug toxicity prediction: Machine learning can be used to predict the toxicity of drugs, helping to identify potential side effects before they reach clinical trials.
- Personalized medicine: Machine learning can be used to personalize treatment plans for patients, by using information such as genetic data, medical history, and response to previous treatments to predict which drugs will be most effective.
- Clinical decision support: Machine learning can be used to support healthcare professionals in making treatment decisions, by providing them with more accurate and up-to-date information about drugs and their potential side effects.

- Drug abuse detection: Machine learning can be used to detect and prevent drug abuse, by analyzing patterns of drug use and identifying at-risk individuals.

9. CONCLUSION

In conclusion, machine learning is a rapidly growing field with many potential applications in drug classification. By leveraging the ability of machine learning algorithms to handle large amounts of data and make accurate predictions, researchers and healthcare professionals can gain a deeper understanding of drugs and their mechanisms of action. This can lead to the discovery of new drugs, the repositioning of existing drugs, and the personalized treatment of patients.

However, it is important to be aware of the potential disadvantages of machine learning, such as bias, lack of interpretability, and high computational cost. These challenges can be addressed through careful data preparation, model selection, and evaluation, and by working with domain experts to ensure that the models are developed and used appropriately.

Overall, machine learning has the potential to revolutionize the field of drug classification, bringing many benefits and enabling new discoveries that could improve the lives of millions of people.

10. FUTURE SCOPE

The future of machine learning in drug classification is very promising, with many opportunities for continued development and innovation

Deep learning models are likely to become even more sophisticated in the future, enabling more accurate predictions and better results for drug classification. Machine learning is likely to be integrated with other technologies, such as genomic data and electronic health records, to provide a more comprehensive understanding of drugs and their effects.
With the growing awareness of the importance of personalized medicine, machine learning is likely to play a larger role in the development of treatments that are tailored to individual patients. Machine learning can help to improve drug safety by predicting the toxicity of drugs before they reach clinical trials and by identifying potential side effects.
As the field of machine learning continues to evolve, new and innovative applications of machine learning in drug classification are likely to emerge, bringing even more benefits to the field.

11. BIBLIOGRAPHY

1. https://www.kaggle.com/datasets/prathamtripathi/drug-classification

12. APPENDIX:

Jupyter Notebook

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix
import warnings
import pickle
from scipy import stats
warnings.filterwarnings('ignore')
plt.style.use('fivethirtyeight')
```

In [2]:
```
# Reading the csv data
df = pd.read_csv(r'D:\DRUG_CLASSIFICATION\data\drug200.csv')
```

In [3]:
```
df.head()
```

Out[3]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	DrugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	DrugY

In [4]:
```
# Shape of csv data
df.shape
```

Out[4]:
```
(200, 6)
```

In [5]:
```
# Finding null values
df.isnull().sum()
```

Out[5]:
```
Age            0
Sex            0
BP             0
Cholesterol    0
Na_to_K        0
Drug           0
dtype: int64
```

In [6]:
```
# Checking the information of features
df.info()
```

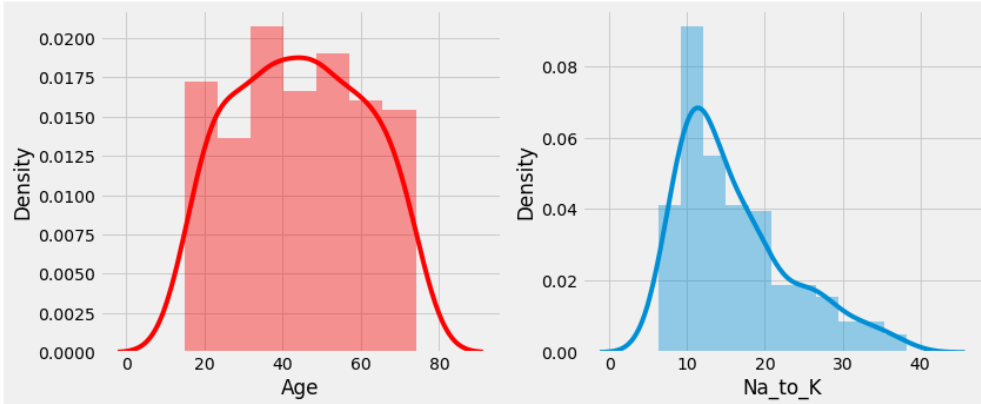
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Age          200 non-null    int64
 1   Sex          200 non-null    object
 2   BP           200 non-null    object
 3   Cholesterol  200 non-null    object
 4   Na_to_K      200 non-null    float64
 5   Drug         200 non-null    object
dtypes: float64(1), int64(1), object(4)
memory usage: 9.5+ KB
```

In [7]:
```
df.describe(include='all')
```

Out[7]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
count	200.000000	200	200	200	200.000000	200
unique	NaN	2	3	2	NaN	5
top	NaN	M	HIGH	HIGH	NaN	DrugY
freq	NaN	104	77	103	NaN	91
mean	44.315000	NaN	NaN	NaN	16.084485	NaN
std	16.544315	NaN	NaN	NaN	7.223956	NaN
min	15.000000	NaN	NaN	NaN	6.269000	NaN
25%	31.000000	NaN	NaN	NaN	10.445500	NaN
50%	45.000000	NaN	NaN	NaN	13.936500	NaN
75%	58.000000	NaN	NaN	NaN	19.380000	NaN
max	74.000000	NaN	NaN	NaN	38.247000	NaN

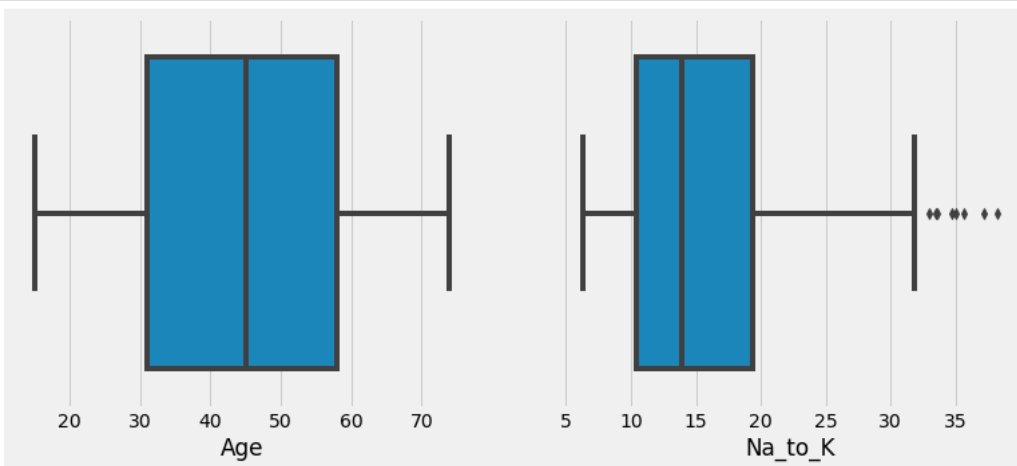
In [8]:
```
# Checking the distribution (normal or skewed)

plt.figure(figsize=(12,5))
plt.subplot(121)
sns.distplot(df['Age'],color='r')
plt.subplot(122)
sns.distplot(df['Na_to_K'])
plt.show()
```



In [9]:
```
# From the above plot age column is normally distributed. Na_to_k is right skewed (mean>mode). To overcome skewness transformatic

print(stats.mode(df['Na_to_K']))
print(np.mean(df['Na_to_K']))
```

```
ModeResult(mode=array([12.006]), count=array([2]))
16.08448499999999
```

In [10]:
```
# Finding outliers

plt.figure(figsize=(12,5))
plt.subplot(121)
sns.boxplot(df['Age'])
plt.subplot(122)
sns.boxplot(df['Na_to_K'])
plt.show()
```



In [11]:
```
# Na_to_K has 8 outliers. In this project we are not going to handle outliers. Most of the classification algorithms are not sens

q1 = np.quantile(df['Na_to_K'],0.25)
q3 = np.quantile(df['Na_to_K'],0.75)

IQR = q3-q1

upper_bound = q3+(1.5*IQR)

print('Upper Bound :',upper_bound)

print('Skewed data :',len(df[df['Na_to_K']>upper_bound]))
```

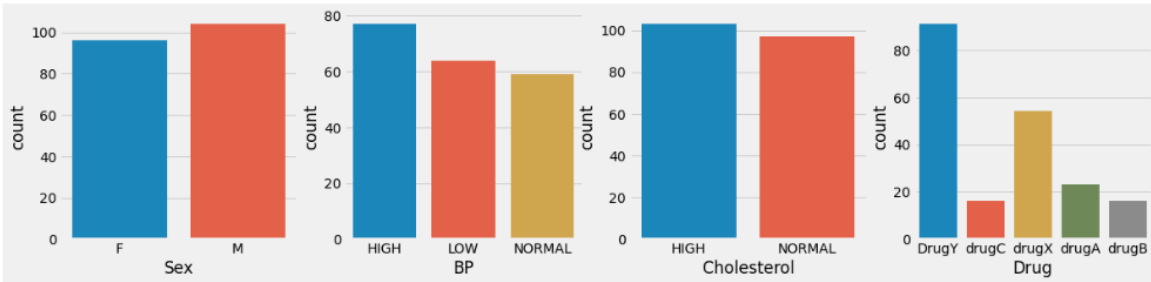
```
Upper Bound : 32.78175
Skewed data : 8
```

In [12]:
```
# Creating a data frame with categorical features for following visualization

df_cat = df.select_dtypes(include='object')
df_cat.head()
```

Out[12]:

	Sex	BP	Cholesterol	Drug
0	F	HIGH	HIGH	DrugY
1	M	LOW	HIGH	drugC
2	M	LOW	HIGH	drugC
3	F	NORMAL	HIGH	drugX
4	F	LOW	HIGH	DrugY

In [13]:
```
# Visualizing the count of categorical variable.

plt.figure(figsize=(18,4))
for i,j in enumerate(df_cat):
    plt.subplot(1,4,i+1)
    sns.countplot(df[j])
```



In [14]:
```
df.head()
```

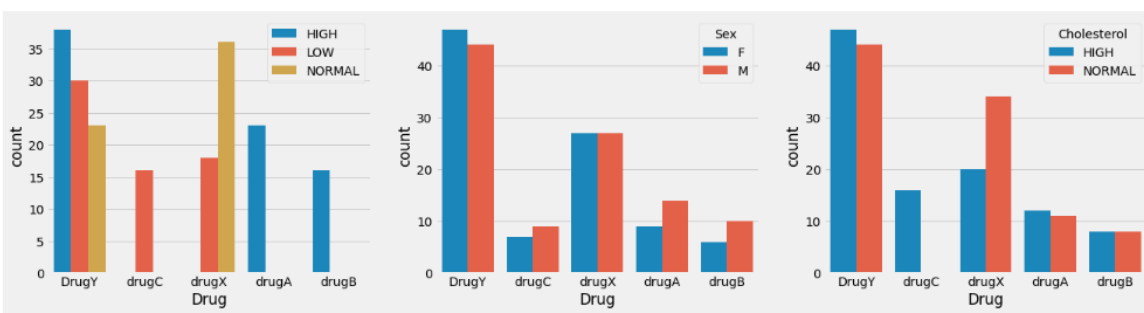
Out[14]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	DrugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	DrugY

In [15]:
```
# Visualizing the relation between drug, BP, sex & cholesterol

plt.figure(figsize=(20,5))
plt.subplot(131)
sns.countplot(df['Drug'],hue=df['BP'])
plt.legend(loc='upper right')
plt.subplot(132)
sns.countplot(df['Drug'],hue=df['Sex'])
plt.subplot(133)
sns.countplot(df['Drug'],hue=df['Cholesterol'])
```

Out[15]: <AxesSubplot:xlabel='Drug', ylabel='count'>

Out[15]: <AxesSubplot:xlabel='Drug', ylabel='count'>



In [16]:
```
# Creating a new column Age_. This column shows the categorized age.

df['Age_'] = ['15-30' if x<=30 else '30-50' if x>30 and x<=50 else '50-75' for x in df['Age']]
df.head()
```

Out[16]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug	Age_
0	23	F	HIGH	HIGH	25.355	DrugY	15-30
1	47	M	LOW	HIGH	13.093	drugC	30-50
2	47	M	LOW	HIGH	10.114	drugC	30-50
3	28	F	NORMAL	HIGH	7.798	drugX	15-30
4	61	F	LOW	HIGH	18.043	DrugY	50-75

In [17]:

```
# Finding the relation between categorized age and drug

pd.crosstab(df['Age_'],[df['Drug']])
```

Out[17]:

Drug	DrugY	drugA	drugB	drugC	drugX
Age_					
15-30	24	6	0	5	13
30-50	33	17	0	7	22
50-75	34	0	16	4	19

In [18]:

```
# Removing the Age_ column

df.drop('Age_',axis=1,inplace=True)
df.head()
```

Out[18]:

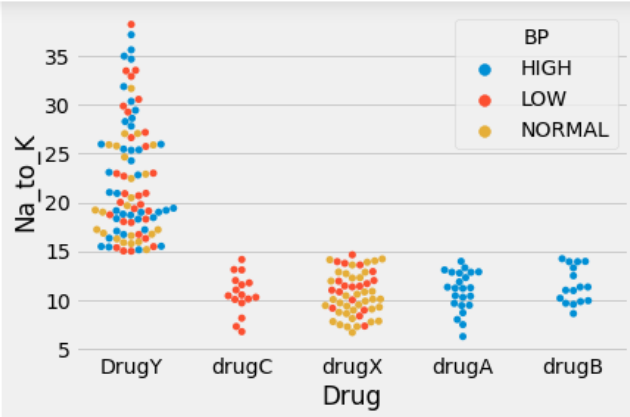
	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	DrugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	DrugY

In [19]:

```
sns.swarmplot(df['Drug'],df['Na_to_K'],hue=df['BP'])

# DrugC is used for low BP patient, DrugY is used on patients having Na_to_K > 15.
```

Out[19]:

```
<AxesSubplot:xlabel='Drug', ylabel='Na_to_K'>
```



In [20]:
```
# Replacing low, normal & high with 0, 1 & 2...

df['BP'] = [0 if x=='LOW' else 1 if x=='NORMAL' else 2 for x in df['BP']]
```

In [21]:
```
# Replacing normal and high cholesterol with 0 & 1

df['Cholesterol'] = [0 if x=='NORMAL' else 1 for x in df['Cholesterol']]
```

In [22]:
```
# Replacing female and male with 0 & 1

df['Sex'] = [0 if x=='F' else 1 for x in df['Sex']]
```

In [23]:
```
df.head()
```

Out[23]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	0	2	1	25.355	DrugY
1	47	1	0	1	13.093	drugC
2	47	1	0	1	10.114	drugC
3	28	0	1	1	7.798	drugX
4	61	0	0	1	18.043	DrugY

In [24]:
```
df['Drug'].value_counts()
```

Out[24]:
```
DrugY    91
drugX    54
drugA    23
drugC    16
drugB    16
Name: Drug, dtype: int64
```

In [25]:
```
x = df.drop('Drug',axis=1)
x.head()
```

Out[25]:

	Age	Sex	BP	Cholesterol	Na_to_K
0	23	0	2	1	25.355
1	47	1	0	1	13.093
2	47	1	0	1	10.114
3	28	0	1	1	7.798
4	61	0	0	1	18.043

In [26]:
```
y = df['Drug']
y.head()
```

Out[26]:
```
0    DrugY
1    drugC
2    drugC
3    drugX
4    DrugY
Name: Drug, dtype: object
```

In [27]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=10)
```

In [28]:
```
print('Shape of x_train {}'.format(x_train.shape))
print('Shape of y_train {}'.format(y_train.shape))
print('Shape of x_test {}'.format(x_test.shape))
print('Shape of y_test {}'.format(y_test.shape))
```

```
Shape of x_train (140, 5)
```

In [28]:
```
print('Shape of x_train {}'.format(x_train.shape))
print('Shape of y_train {}'.format(y_train.shape))
print('Shape of x_test {}'.format(x_test.shape))
print('Shape of y_test {}'.format(y_test.shape))
```

```
Shape of x_train (140, 5)
Shape of y_train (140,)
Shape of x_test (60, 5)
Shape of y_test (60,)
```

In [29]:
```
rf = RandomForestClassifier()
rf.fit(x_train,y_train)
```

Out[29]:
```
RandomForestClassifier()
```

In [30]:
```
ypred = rf.predict(x_test)
```

In [31]:
```
confusion_matrix(y_test,ypred)
```

Out[31]:
```
array([[25,  0,  0,  0,  0],
       [ 0,  7,  0,  0,  0],
       [ 0,  2,  4,  0,  0],
       [ 0,  0,  0,  7,  0],
       [ 0,  0,  0,  0, 15]], dtype=int64)
```

In [32]:
```
print(classification_report(y_test,ypred))
```

In [32]:
```
print(classification_report(y_test,ypred))
```

```
              precision    recall  f1-score   support

       DrugY       1.00      1.00      1.00        25
       drugA       0.78      1.00      0.88         7
       drugB       1.00      0.67      0.80         6
       drugC       1.00      1.00      1.00         7
       drugX       1.00      1.00      1.00        15

    accuracy                           0.97        60
   macro avg       0.96      0.93      0.93        60
weighted avg       0.97      0.97      0.97        60
```

In [33]:
```
pickle.dump(rf,open('model.pkl','wb'))
```

Python Source Code