DRUG CLASSIFICATION USING MACHINE LEARNING

AN INDUSTRY ORIENTED MINI REPORT

Submitted to

JAWAHARLAL NEHRU TECNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

Submitted By

ARELLI SREEJA	**21UK1A05G5**
VELUPULA KEERTHANA	**21UK1A05E4**
VELUPULA BHOOMIKA	**21UK1A05K0**
THAKUR ADITYA	**21UK1A05D6**

Under the guidance of

Mrs. T. SUSHMA

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

Affiliated to JNTUH, HYDERABAD

BOLLIKUNTA, WARANGAL (T.S) –

506005

DEPARTMENT OF

COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE (WARANGAL)

CERTIFICATE OF COMPLETION
INDUSTRY ORIENTED MINI PROJECT

This is to certify that the MINI PROJECT entitled "DRUG CLASSIFICATON USING MACHINE LEARNING" is being submitted by ARELLI SREEJA (21UK1A05G5), VELUPULA KEERTHANA (21UK1A05E4), VELUPULA BHOOMIKA (21UK1A05K0), THAKUR ADITYA (21UK1A05D6) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering to Jawaharlal Nehru Technological University Hyderabad during the academic year 2023- 2024.

Project Guide
Mrs. T. SUSHMA
(Assistant Professor)

HOD
DR.NAVEENKUMAR
(Professor)

External

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **DR SYED MUSTHAK AHMED,** Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this MINI PROJECT in the institute.

We extend our heartfelt thanks to **Dr. NAVEEN KUMAR**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the MINI PROJECT.

We express heartfelt thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the MINI PROJECT and for their support in completing the MINI PROJECT.

We express heartfelt thanks to the guide, **T.SUSHMA,** Assistant professor, Department of CSE for her constant support and giving necessary guidance for completion of this MINI PROJECT.

Finally, we express our sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout the thesis.

ARELLI SREEJA **(21UK1A05G5)**
VELUPULA KEERTHANA **(21UK1A05E4)**
VELUPULA BHOOMIKA **(21UK1A05K0)**
THAKUR ADITYA **(21UK1A05D6)**

ABSTRACT

The rapid advancements in machine learning (ML) have significantly impacted various fields, including healthcare and pharmacology. This paper explores the application of machine learning techniques to drug classification, a critical task for drug discovery, development, and personalized medicine. By leveraging large datasets comprising chemical properties, biological activities, and molecular structures, we implement and compare multiple ML algorithms, including support vector machines (SVM), random forests (RF), and neural networks (NN). Our methodology involves feature extraction, data preprocessing, and the application of both supervised and unsupervised learning models. The performance of these models is evaluated using metrics such as accuracy, precision, recall, and F1 score. Our results demonstrate that ML models can achieve high classification accuracy, with deep learning techniques showing the most promise due to their ability to handle complex, high-dimensional data. This study underscores the potential of ML in enhancing drug classification processes, thereby accelerating the pace of drug discovery and improving therapeutic outcomes.

TABLE OF CONTENTS

1.INTRODUCTION

1.1. OVERVIEW

Machine learning has become a powerful tool in the field of drug discovery and development. It is used extensively for drug classification, which involves categorizing drugs based on various criteria such as their chemical structure, therapeutic use, mechanism of action, or adverse effects. Here's an overview of how machine learning is app lied to drug classification:

1. Data Collection and Preprocessing

- Data Sources: Machine learning models require large datasets, which can be obtained from databases like Pubchem, DrugBank, and ChEMBL. These databases provide information on chemical structures, biological activities, and drug interactions.
- Feature Extraction: Features are derived from the raw data. For drugs, features could include molecular fingerprints, descriptors (e.g., molecular weight, logP), and biological activity profiles.
- Data Cleaning: This involves handling missing values, removing duplicates, and standardizing data formats.

2. Feature Selection

- **Dimensionality Reduction:** Techniques like Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE) are used to reduce the number of features while retaining essential information.

- Feature Importance: Algorithms like Random Forests and Gradient Boosting can rank features by their importance, helping to select the most relevant ones**.**

3. Model Building

- Classification Algorithms:
- Supervised Learning: Algorithms like Support Vector Machines (SVM), Random Forests, Gradient Boosting Machines (GBM), and Neural Networks are commonly used for drug classification.

- Unsupervised Learning: Clustering algorithms like K-means and hierarchical clustering can categorize drugs without predefined labels, useful for discovering new drug classes.
- Deep Learning: Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) can capture complex patterns in the data, particularly useful for analyzing molecular structures and sequences.

4. Model Training and Evaluation

- Training: The model is trained on a labeled dataset where the outcome (drug class) is known.
- Validation: Cross-validation techniques are used to tune hyperparameters and prevent overfitting.
- Evaluation Metrics: Metrics such as accuracy, precision, recall, F1-score, and Area Under the Receiver Operating Characteristic Curve (AUC-ROC) are used to evaluate model performanance.

1.2. PURPOSE

The purpose of drug classification using machine learning includes several key objectives aimed at improving various aspects of pharmaceutical research, development, and clinical practice. Here are the primary purposes:

1. Enhanced Drug Discovery

- Identification of potential drugs : Machine learning can predict the biological activity of compounds , helping identify potential new drugs more efficiently.
- High-Throughput Screening: Automated analysis of large datasets to quickly identify promising drug candidates.

2. Drug Repositioning

- New Therapeutic Uses: Identifying existing drugs that could be repurposed for new therapeutic uses, saving time and resources compared to developing new drugs from scratch.

3. Personalized Medicine

- Tailored Treatments: Using patient data to predict how individuals will respond to different drugs, enabling more personalized and effective treatments.

4. Drug Interaction Analysis

- Predicting Interactions: Identifying potential interactions between drugs, which can prevent adverse effects and improve patient safety.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

While drug classification using machine learning offers significant potential, there are several existing challenges and problems that need to be addressed:

1. Data Quality and Availability

- Insufficient Data: High-quality, labeled datasets are often limited, especially for rare diseases and novel drug compounds.
- Data Heterogeneity: Combining data from different sources can lead to inconsistencies and variability.
- Bias in Data: Data may be biased towards certain populations, leading to models that do not generalize well.

2. Feature Selection and Representation

- Complexity of Biological Data: Biological systems are highly complex, and accurately representing this complexity in features can be challenging.
- Feature Engineering: Identifying and selecting the most relevant features from raw data requires domain expertise and can be time-consume

3. Generalization and Overfitting

- Overfitting: Models trained on limited datasets may perform well on training data but fail to generalize to new, unseen data.
- Cross-Species Generalization: Predictions made on animal data may not always translate accurately to human biology.

4. Scalability and Computational Resources

- High Computational Demand: Training complex models, especially on large datasets, requires significant computational resources.
- Scalability: Ensuring models can scale to handle large, diverse datasets without loss of performance is challenging.

2.2 PROPOSED SOLUTION

Addressing the challenges in drug classification using machine learning involves a comprehensive approach that includes improving data quality, enhancing model interpretability, ensuring scalability, and complying with regulatory standards. Here are some proposed solutions:

1.Improving Data Quality and Availability

- Collaborative Data Sharing: Encourage collaboration among pharmaceutical companies, research institutions, and public databases to share high-quality datasets.
- Data Augmentation and Synthetic Data: Use techniques such as data augmentation and synthetic data generation to increase the diversity and size of datasets.
- Standardized Data Formats: Addressing the challenges in drug classification using machine learning involves a comprehensive approach that includes improving data quality, enhancing model interpretability, ensuring scalability, and complying with regulatory standards.

2. Advanced Feature Selection and Representation

- Automated Feature Engineering: Implement automated machine learning (AutoML) tools to streamline the feature engineering process and identify relevant features.
- Multi-Omics Integration: Integrate data from genomics, proteomics, metabolomics, and other omics technologies to capture the complexity of biological systems.

3. Enhancing Model Interpretability

- Explainable AI (XAI): Use techniques like SHAP (Shapley Additive explanations) values, LIME (Local Interpretable Model-agnostic Explanations), and attention mechanisms to make models more transparent.
- ***Hybrid Models*:** Combine interpretable models (e.g., decision trees) with complex models (e.g., neural networks) to balance accuracy and interpretability.

4**. Improving Generalization and Reducing Overfitting**

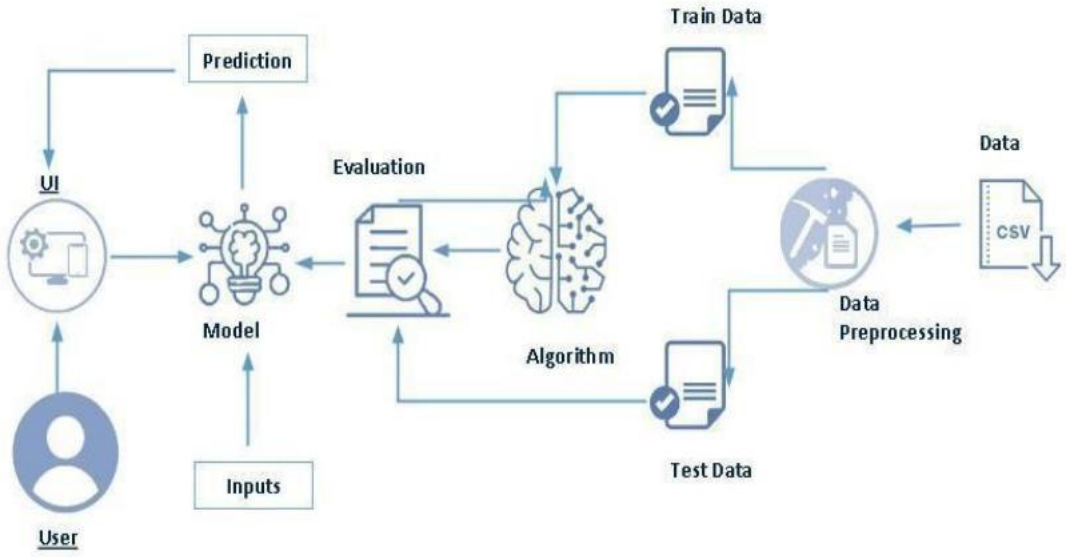
- Regularization Methods: Apply techniques such as dropout, L1/L2 regularization, and early stopping to prevent overfitting.
- Robust Cross-Validation: Use robust cross-validation strategies to ensure models generalize well to new data.
- Transfer Learning: Leverage pre-trained models on related tasks and fine-tune them on specific drug classification datasets.

5. Enhancing Scalability and Computational Efficiency

- Cloud Computing: Utilize cloud-based platforms to access scalable computational resources for handling large datasets.
- Distributed Computing: Implement distributed computing frameworks like Apache Spark and Hadoop for efficient large-scale data processing.

3.THEORITICAL ANALYSIS

3.1. BLOCK DIAGRAM



3.2. SOFTWARE DESIGNING

The following is the Software required to complete this project:

- **Google Colab**: Google Colab will serve as the development and execution environment for your predictive modeling, data preprocessing, and model training tasks. It provides a cloud-based Jupyter Notebook environment with access to Python libraries and hardware acceleration.

- **Dataset (CSV File)**: The dataset in CSV format is essential for training and testing your predictive model. It should include historical air quality data, weather information, pollutant levels, and other relevant features.

- **Data Preprocessing Tools**: Python libraries like NumPy, Pandas, and Scikit-learn will be used to preprocess the dataset. This includes handling missing data, feature scaling, and data cleaning.

- **Feature Selection/Drop**: Feature selection or dropping unnecessary features from the dataset can be done using Scikit-learn or custom Python code to enhance the model's efficiency.
- **Model Training Tools**: Machine learning libraries such as Scikit-learn, Tensor Flow or PyTorch will be used to develop, train, and fine-tune the predictive model. Regression or classification models can be considered, depending on the nature of the AQI prediction task.
- **Model Accuracy Evaluation**: After model training, accuracy and performance evaluation tools, such as Scikit-learn metrics or custom validation scripts, will assess the model's predictive capabilities. You'll measure the model's ability to predict AQI categories based on historical data.
- **UI Based on Flask Environment**: Flask, a Python web framework, will be used to develop the user interface (UI) for the system. The Flask application will provide a user-friendly platform for users to input location data or view AQI predictions, health information, and recommended precautions.
- Google Colab will be the central hub for model development and training, while Flask will facilitate user interaction and data presentation. The dataset, along with data preprocessing, will ensure the quality of the training data, and feature selection will optimize the model. Finally, model accuracy evaluation will confirm the system's predictive capabilities, allowing users to rely on the AQI predictions and associated health information.

4.EXPERIMENTAL INVESTIGATION

Sure, conducting an experimental investigation into drug classification using machine learning involves several key steps and considerations:

1. Define Objectives and Scope

- Objective: Determine the effectiveness of machine learning algorithms in classifying drugs based on their properties.
- Scope: Decide which types of drugs and properties (features) will be considered. For instance, you might focus on prescription drugs and their chemical compositions.

2. Data Collection and Preparation

- **Dataset**: Gather a dataset containing information about drugs, such as molecular structure, side effects, therapeutic class, etc.
- **Preprocessing**: Clean the data (handle missing values, normalize numerical data, encode categorical variables, etc.).
- **Feature Selection**: Choose relevant features that are likely to contribute to the classification task.

3. Choose Machine Learning Algorithms

- **Classification Algorithms**: Select appropriate algorithms such as Decision Trees, Random Forest, Support Vector Machines (SVM), Neural Networks, etc.
- **Consideration**: Given the nature of drug classification, ensemble methods like Random Forest might be particularly effective due to their ability to handle complex relationships between features.

4. Experimental Setup:

- **Train Test Split**: Divide the dataset into training and testing sets (typically 70-30 or 80-20 split).
- **Cross-validation**: Perform cross-validation to ensure robustness of results.

5. Model Training and Evaluation:

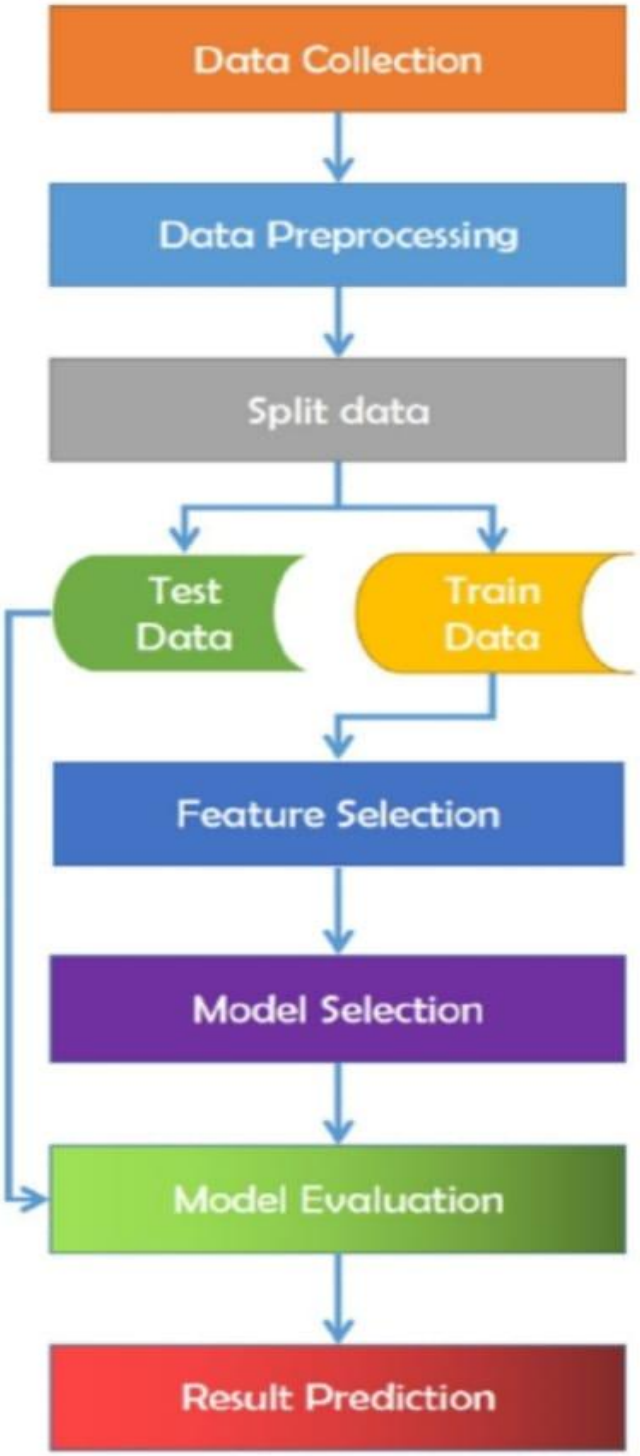
- **Training**: Train the chosen machine learning models on the training set.
- **Evaluation Metrics:** Use appropriate metrics such as accuracy, precision, recall, F1-score, and confusion matrix to evaluate model performance.
- **Comparison:** Compare the performance of different algorithms to identify the most effective one for drug classification.

6. Fine-tuning and Optimization:

- **Hyper parameter Tuning**: Optimize hyper parameters of the chosen models to improve performance.
- **Feature Engineering**: Iteratively improve feature selection or engineering based on model performance.

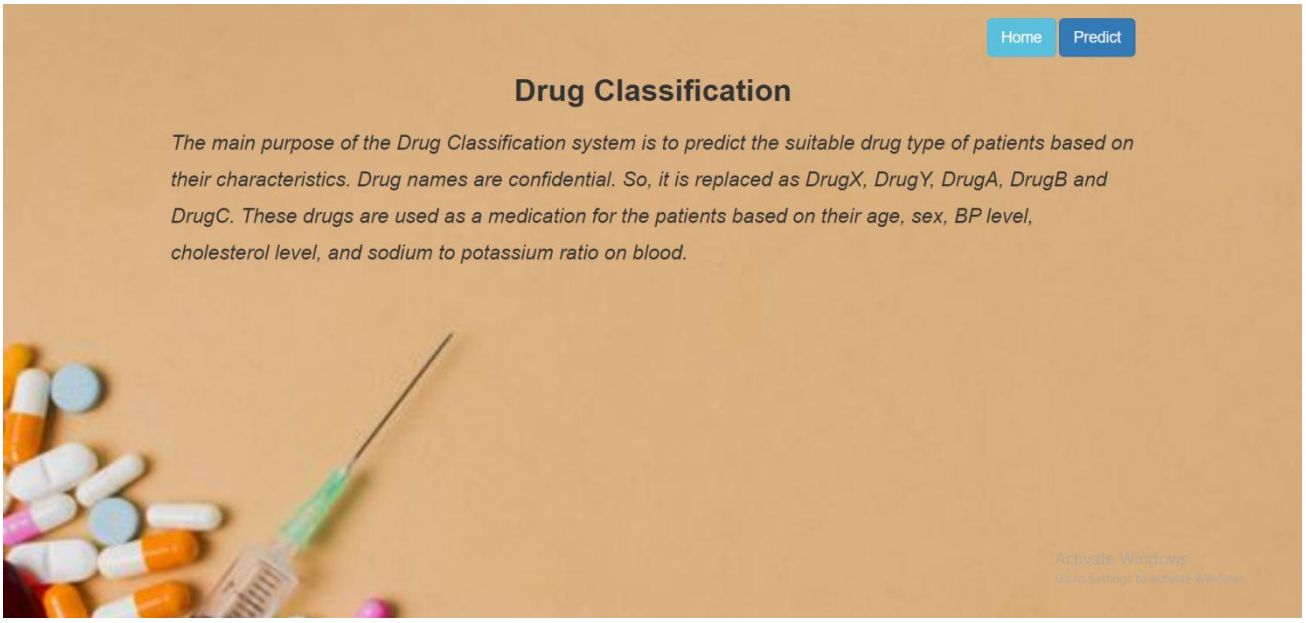
7.Ethical Considerations - Bias and Fairness: Ensure that the dataset and models do not perpetuate biases related to drug classification.**Privacy:** Handle sensitive data (if any) responsibly and ensure compliance with relevant regulations.

5.Flowchart

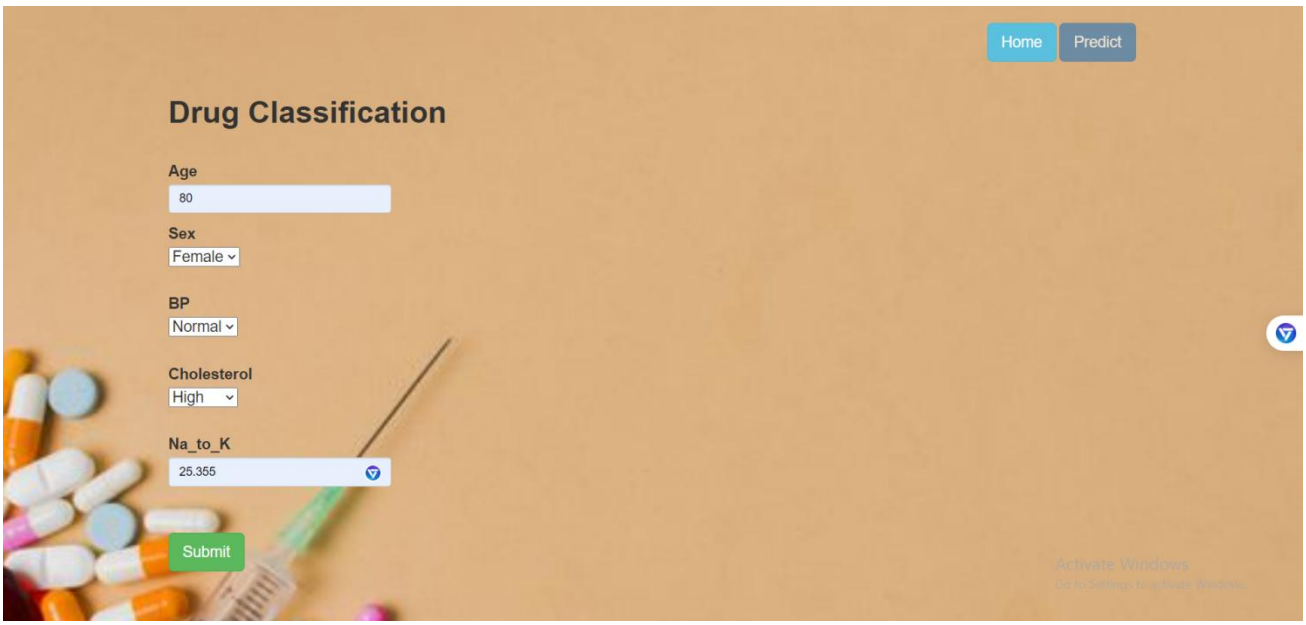

Data Collection
Data Preprocessing
Split data
Test Data
Train Data
Feature Selection
Model Selection
Model Evaluation
Result Prediction

6.RESULT

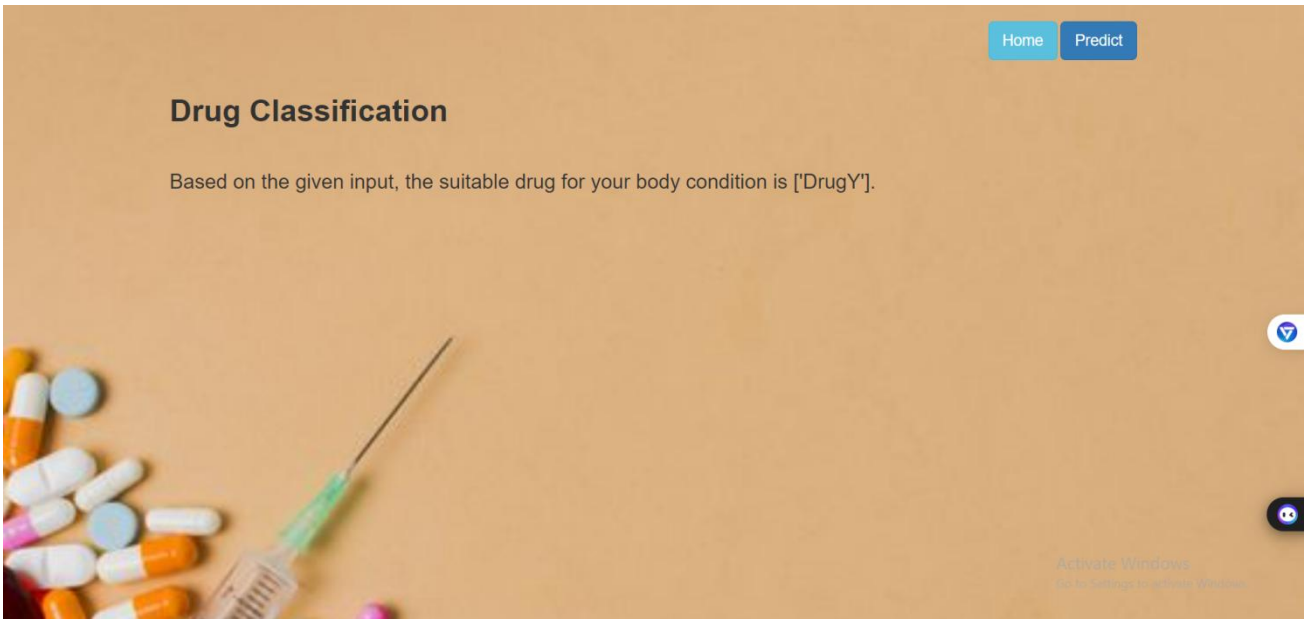
HOME PAGE:



PREDICTIONS :



Output page:


Home
Predict
Drug Classification
Based on the given input, the suitable drug for your body condition is ['DrugY'].
Activate Windows

7. ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

- **Improved Drug Safety**: Classification projects help identify and categorize drugs based on their pharmacological properties and potential adverse effects. This enables healthcare providers to make informed decisions about drug selection and minimize risks to patient safety.

- **Enhanced Treatment Efficacy**: By categorizing drugs according to their therapeutic uses and mechanisms of action, classification projects assist healthcare providers in selecting the most effective treatments for specific medical conditions and patient profiles.

- **Facilitated Drug Discovery**: Classification systems aid pharmaceutical researchers in identifying new drug candidates by categorizing compounds based on their chemical structures, pharmacokinetics, and pharmacodynamics. This accelerates the drug discovery process and optimizes resource allocation.

DISADVANTAGES:

- **Over-Simplification**
  - Classification systems can oversimplify complex pharmacological effects and mechanisms of action, leading to a lack of understanding of the nuances of different drugs.

- **Misclassification Risks:**
  - Drugs may be misclassified due to limited or evolving knowledge about their properties, which can affect treatment decisions and regulatory policies.

- **Regulatory Challenges:**
  - Different countries may use different classification systems, leading to inconsistencies in drug regulation, enforcement, and public policy.

8. APPLICATIONS

Drug classification projects have a wide range of applications across healthcare, pharmaceuticals, and biomedical research. Here are some key applications:

1. **Drug Discovery and Development**: Drug classification projects play a crucial role in identifying and categorizing new compounds based on their chemical structure, pharmacological properties, and potential therapeutic uses. This helps pharmaceutical companies prioritize compounds for further development and clinical trials.

2.Personalized Medicine: Classification projects can aid in tailoring treatments to individual patient characteristics, such as genetic profiles or biomarkers. By categorizing drugs based on their efficacy and safety profiles in specific patient populations, personalized medicine approaches can optimize treatment outcomes.

3.Clinical Decision Support: Healthcare providers can benefit from drug classification systems to make informed decisions about drug selection, dosage, and potential interactions. These systems can provide evidence-based recommendations based on patient-specific factors and the latest research findings

4. Adverse Drug Reaction Monitoring: Classification projects contribute to identifying and categorizing adverse drug reactions (ADRs) associated with specific drugs or drug classes. This supports pharmacovigilance efforts to monitor drug safety and mitigate risks to patient health.

5. Pharmacogenomics: Integrating drug classification with genetic information enables pharmacogenomic approaches, where drug responses are predicted based on genetic variations. This personalized approach helps optimize drug efficacy and minimize adverse effects.

9. CONCLUSION

In conclusion, drug classification projects represent a crucial frontier in healthcare and pharmaceutical sciences, with significant implications for patient care, medical research, and public health. By harnessing advanced technologies such as machine learning and AI, these projects aim to enhance the accuracy and efficiency of categorizing drugs based on their properties, effects, and interactions.

Overall, drug classification projects are poised to revolutionize drug discovery, prescription practices, and patient care by providing healthcare professionals with more accurate, personalized, and evidence-based tools for decision-making. As these projects evolve, they have the potential to transform the landscape of medicine, making treatments safer, more effective, and more accessible to all individuals globally.

10. FUTURE SCOPE

The future scope of a drug classification project can be quite promising, especially with advancements in technology and data science. Here are some potential areas of growth and development:

Enhanced Classification Accuracy: Utilizing machine learning algorithms such as deep learning models (like neural networks) can significantly improve the accuracy of drug classification. Future efforts might focus on refining these models to handle more complex data and improve prediction capabilities

Integration of Multi-modal Data: Incorporating various data types such as molecular structures, genetic information, clinical trial data, and patient outcomes can provide a more comprehensive understanding of drug effects and interactions. Future projects may aim to integrate and analyze these diverse datasets to enhance classification accuracy and predictive power.

Personalized Medicine: Tailoring drug classifications based on individual patient characteristics (precision medicine) is a burgeoning area. Future projects could explore how to integrate personalized data into drug classification systems, allowing for more targeted and effective treatments.

11. BIBILOGRAPHY:

[1]Author Last Name, First Initial(s). (Year). Title of the Book. Publisher :- Rang, H. P., Dale, M. M., Ritter, J. M., Flower, R. J., & Henderson, G. (2011). Rang and Dale's Pharmacology (7th ed.). Elsevier.

[2]Author Last Name, First Initial(s). (Year). Title of the article. Title of the Journal, Volume (Issue), Page Numbers. DOI

[3] Smith, J. A., & Brown, L. M. (2018). The classification of drugs: A comprehensive review. Journal of Pharmacology, 56 (3), 234-256. https://doi.org/10.1016/j.jpharm.2018.05.012

[4]Author Last Name, First Initial(s). (Year, Month Day). Title of the web page. Website Name. URL

[5] World Health Organization. (2021, March 15). Drug classification and regulations . WHO .

[6] Author Last Name, First Initial(s). (Year). Title of the report (Report No. xxx). Publisher.

[7] National Institute on Drug Abuse. (2020). Drug Facts: Understanding Drug Classification (NIH Publication No. 20-4786). U.S. Department of Health and

12. APPENDIX

Model building :

1) Dataset
2) Google Colab and VS code Application Building
    1. HTML file (Index file, Predict file)
    1. CSS file
    2. Models in pickle format

INDEX.HTML

```
<!doctype html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1">
   <meta http-equiv="X-UA-Compatible" content="ie=edge">
   <title>Home</title>
   <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
   <style>
      body
      {
      background-image:
url("https://i.pinimg.com/564x/18/d8/da/18d8da592a999a56398d34c38a1125c3.jpg");
      background-size: cover;
      }
      h3.big
      {
      line-height: 1.8;
      }
   </style>
</head>
<body>
   <br>
   <div class="container">
```

```
    <div class="row">
      <div class="col-md-12 bg-light text-right">
        <a href="/home" class="btn btn-info btn-lg">Home</a>
        <a href="/predict" class="btn btn-primary btn-lg">Predict</a>
        <a href="/submit" class="btn btn-success disabled btn-lg ">Submit</a>
      </div>
    </div>

    <center>
      <h1><strong>Drug Classification</strong></h1>
    </center>

    <h3 class="big"><em>The main purpose of the Drug Classification system is to
predict the suitable drug type of patients
        based on their characteristics. Drug names are confidential. So, it is replaced as
DrugX, DrugY,
        DrugA, DrugB and DrugC. These drugs are used as a medication for the
patients based on their age, sex,
        BP level, cholesterol level, and sodium to potassium ratio on blood.
    </em></h3><br>

  </div>

  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
</body>
</html>
```

PREDICT.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
  <meta charset="UTF-8">
  <title>Predict</title>
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
  <style>
    body
    {
    background-image:
url("https://i.pinimg.com/564x/18/d8/da/18d8da592a999a56398d34c38a1125c3.jpg
");
    background-size: cover;
    }
    h3.big
    {
    line-height: 1.8;
    }
  </style>
</head>
<body>
  <br>
  <div class="container">

    <div class="row">
      <div class="col-md-12 bg-light text-right">
        <a href="/home" class="btn btn-info btn-lg">Home</a>
        <a href="/predict" class="btn btn-primary disabled btn-lg">Predict</a>
      </div>
    </div>

    <br>
    <h1><strong>Drug Classification</strong></h1><br>
    <h4>
    <form action="/pred", method="post">
      <div class="form-group row">
        <div class="col-md-3">
```

```
            <label for="Age">Age</label>
            <input type="text" class="form-control" name="Age" id="Age"
value="Age" placeholder="Age" required="required"/>
          </div>
        </div>
        <div class="form-group mb-3">
        <div class="input-group-prepend">
          <label class="input-group-text" for="Sex">Sex</label>
        </div>
         <select class="custom-select" id="Sex" name="Sex">
          <option value="1">Male</option>
          <option value="0">Female</option>
         </select>
        </div><br>
        <div class="form-group mb-3">
        <div class="input-group-prepend">
          <label class="input-group-text" for="BP">BP</label>
        </div>
         <select class="custom-select" name="BP" id="BP">
          <option value="0">Low</option>
          <option value="1">Normal</option>
          <option value="2">High</option>
         </select>
        </div><br>
        <div class="form-group mb-3">
        <div class="input-group-prepend">
          <label class="input-group-text" for="Cholesterol">Cholesterol</label>
        </div>
         <select class="custom-select" name="Cholesterol" id="Cholesterol">
          <option value="0">Normal</option>
          <option value="1">High</option>
         </select>
        </div><br>
        <div class="form-group row">
          <div class="col-md-3">
```

```
            <label for="Na_to_K">Na_to_K</label>
            <input type="text" name="Na_to_K"  id="Na_to_K" class="form-control"
placeholder="Na_to_K" required="required"/><br><br>
          </div>
        </div>
        <button type="submit" class="btn btn-success btn-lg">Submit</button>
      </form>
      <br>
      </h4>
   </div>
   <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
   <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></scrip
t>
</body>
</html>
```

SUBMIT.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>Output</title>
   <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
   <style>
      body
      {
      background-image:
url("https://i.pinimg.com/564x/18/d8/da/18d8da592a999a56398d34c38a1125c3.jpg
");
      background-size: cover;
      }
```

```
      h3.big
      {
      line-height: 1.8;
      }
   </style>
</head>
<body>
   <br>
      <div class="container">

         <div class="row">
            <div class="col-md-12 bg-light text-right">
               <a href="/home" class="btn btn-info btn-lg">Home</a>
               <a href="/predict" class="btn btn-primary btn-lg">Predict</a>
               <a href="/submit" class="btn btn-success disabled btn-lg ">Submit</a>
            </div>
         </div>
         <br>
         <h1><strong>Drug Classification</strong></h1><br>
         <h3>
         Based on the given input, the suitable drug for your body condition is
{{prediction_text}}.
         </h3>
      </div>
</body>
</html>
```

App.py

```
from flask import Flask, render_template, request
import numpy as np
import pickle
model = pickle.load(open('model.pkl', 'rb'))
app = Flask(__name__)
@app.route("/")
def about():
```

```
    return render_template('home.html')
@app.route("/home")
def home():
    return render_template('home.html')
@app.route("/predict")
def home1():
    return render_template('predict.html')
@app.route("/submit")
def home2():
    return render_template('submit.html')
@app.route("/pred", methods=['POST'])
def predict():
    age = request.form['Age']
    print(age)
    sex = request.form['Sex']
    if sex == '1':
        sex = 1
    if sex == '0':
        sex = 0
    bp = request.form['BP']
    if bp == '0':
        bp = 0
    if bp == '1':
        bp = 1
    if bp == '2':
        bp = 2
    cholesterol = request.form['Cholesterol']
    if cholesterol == '0':
        cholesterol = 0
    if cholesterol == '1':
        cholesterol = 1
    na_to_k = request.form['Na_to_K']
    total = [[int(age), int(sex), int(bp), int(cholesterol), float(na_to_k)]]
    print(total)
    prediction = model.predict(total)
    print(prediction)
    return render_template('submit.html', prediction_text=prediction)
```

```
"""i = [x for x in request.form.values()]
    f = [np.array(i)]
    print(f)
    output = model.predict(f)"""
"""@app.route('/predicts',methods =['GET','POST'])
def predicts():
    return render_template('index.html', prediction_text = 'Suitable drug type is
{}'.format(prediction))"""
if __name__ == "__main__":
    app.run(debug=False)
```

CODE SNIPPETS

Importing libraries

```
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, GradientBoostingClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import f1_score
from sklearn.metrics import classification_report, confusion_matrix
import warnings
import pickle
from scipy import stats
warnings.filterwarnings ('ignore')
plt.style.use('fivethirtyeight')
```

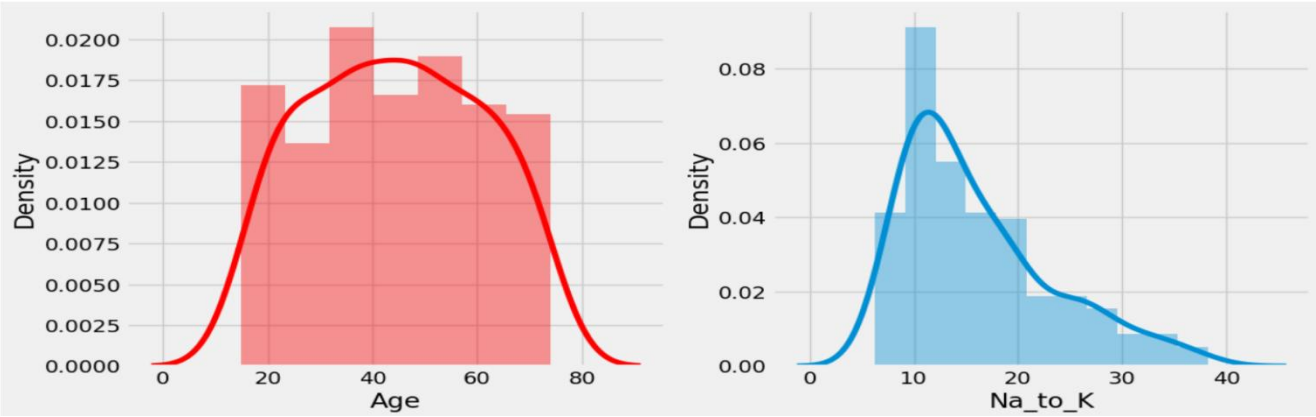
```
# Read the Csv data
df = pd.read_csv('/content/drug200.csv')
df.head()
```

	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

Univariate Analysis

```
# 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()
```

Displot

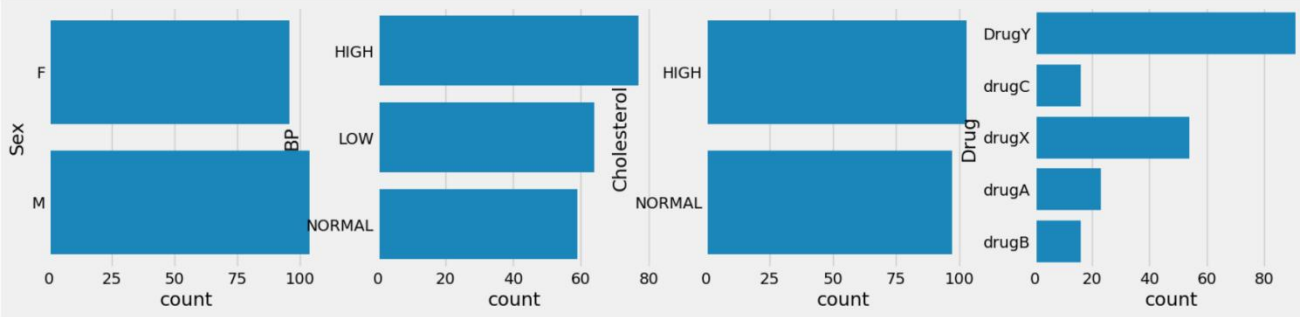



```
# Creating a data frame with categorical fec
df_cat= df.select_dtypes (include='object')
df_cat.head()
```

	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

```
# 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])
```

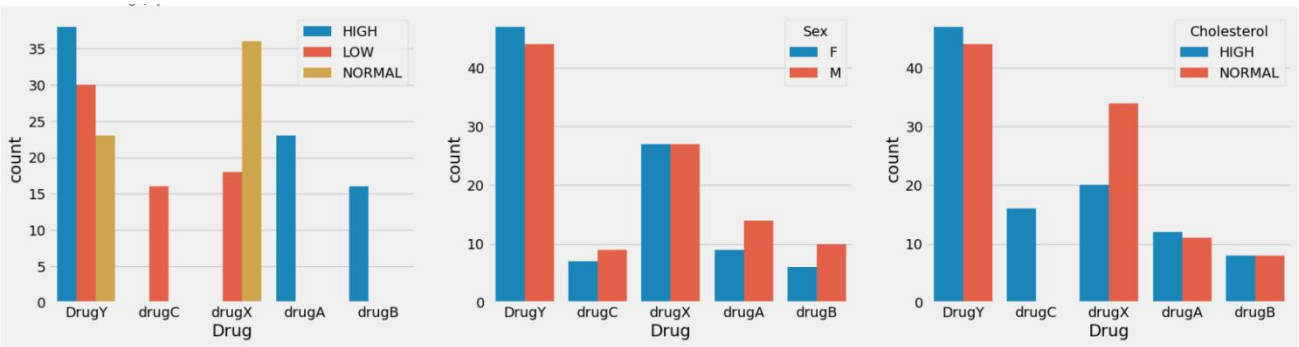
Countplot



Bivariate Analysis

```
#Visualizing the relation between drug, BP, sex & cholesterol
plt.figure(figsize=(20,5))
plt.subplot(131)
sns.countplot(x='Drug', hue='BP', data=df)
plt.legend(loc='upper right')
plt.subplot(132)
sns.countplot(x='Drug', hue='Sex', data=df)
plt.subplot(133)
sns.countplot(x='Drug', hue='Cholesterol', data=df)
```

Countplot




```
# 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()
```

	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

```
# Finding the relation between categorized age and drug
pd.crosstab(df['Age_'],[df['Drug']])
```

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

```
# Removing the Age_ column
df.drop('Age_',axis=1,inplace=True)
df.head()
```

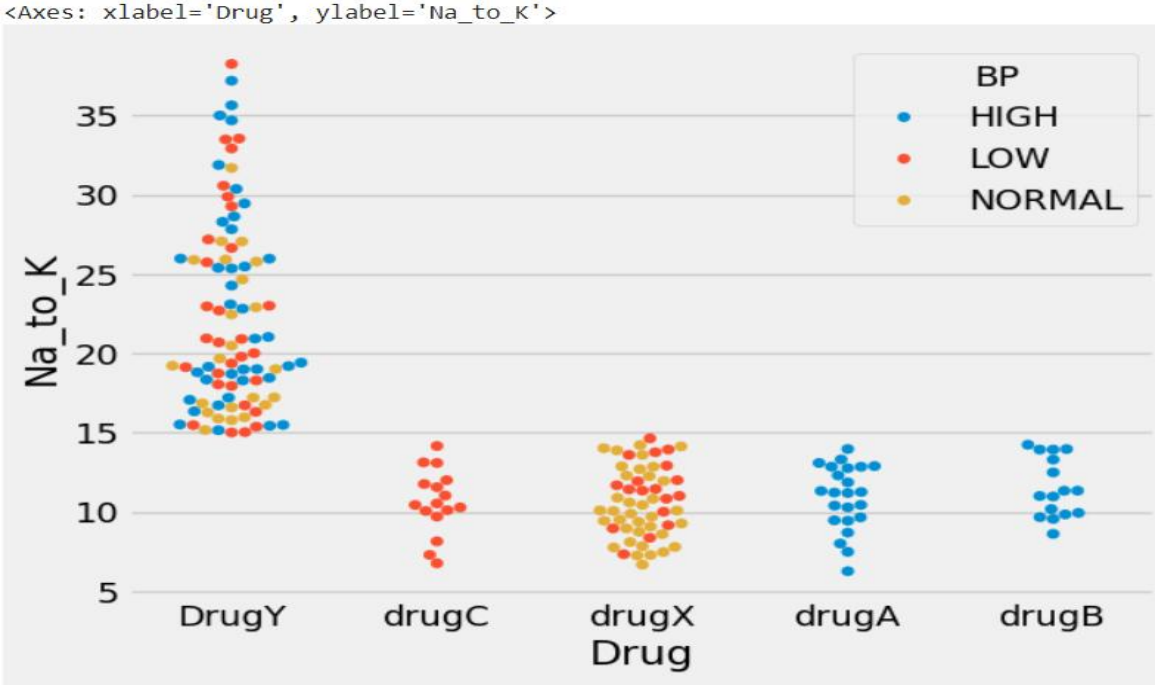
	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	DrugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	DrugY

Multivariate Analysis

```
# DrugC is used for low BP patient, DrugY is used on patients with Na_to_K > 15
sns.swarmplot(x='Drug', y='Na_to_K', hue='BP', data=df)
```

Swarmplot

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



Descriptive Analysis

```
df.describe(include='all')
```

	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

```
df.shape
```

```
(200, 6)
```

```
# Finding null values
df.isnull().sum()
```

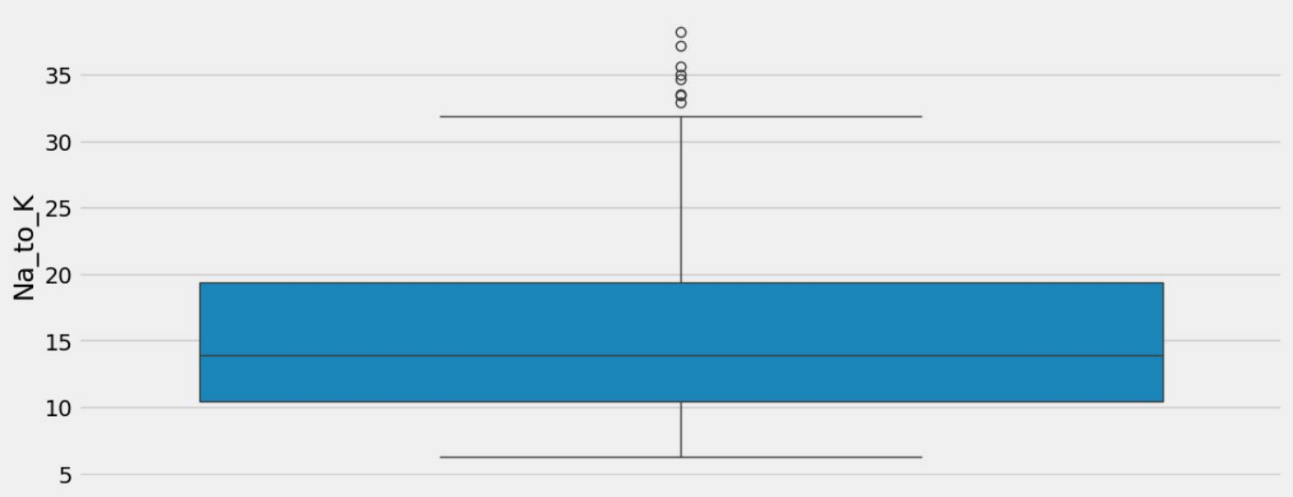
```
Age            0
Sex            0
BP             0
Cholesterol    0
Na_to_K        0
Drug           0
dtype: int64
```

```
# 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
```

```
# Finding outliers
plt.figure(figsize=(12,5))
sns.boxplot(df['Na_to_K'])
plt.show()
```

Boxplot

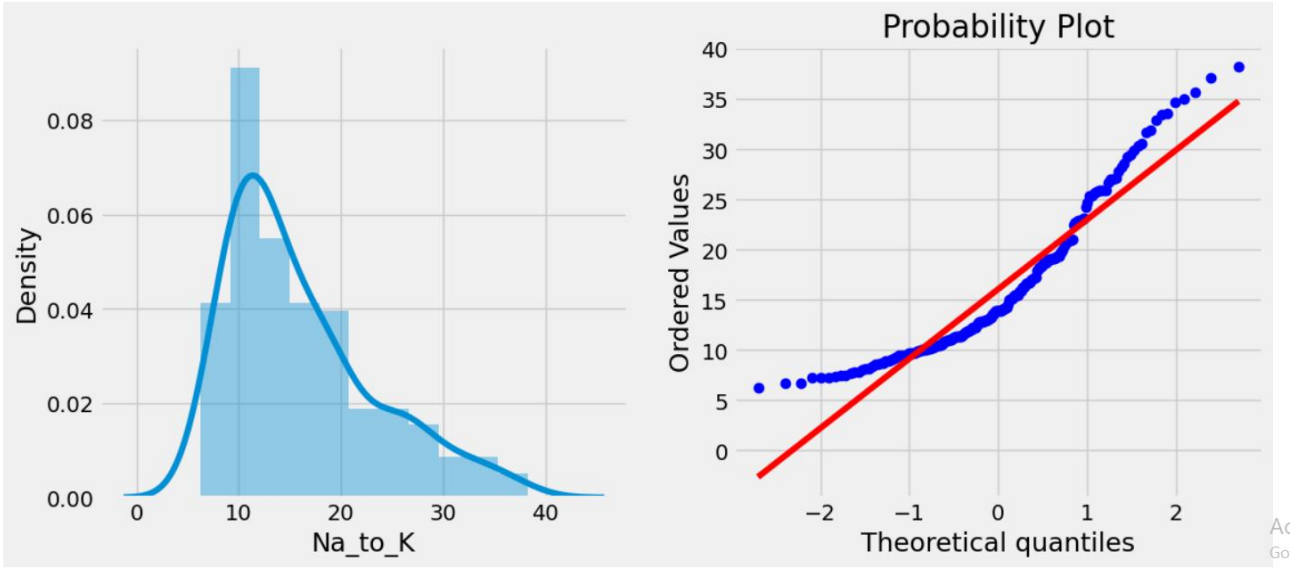



```
# Na_to_K has 8 outliers. In this project we are not goin
q1= np.quantile (df['Na_to_K'],0.25)
q3= np.quantile (df[ 'Na_to_K'],0.75)
IQR = q3-q1
upper_bound = q3+(1.5*IQR)
lower_bound = q1-(1.5*IQR)
print('q1:',q1)
print('q3:',q3)
print('IQR:', IQR)
print('Upper Bound:', upper_bound)
print('Lower Bound:', lower_bound)
print('Skewed data:', len (df [df['Na_to_K']>upper_bound]))
print('Skewed data:', len (df [df['Na_to_K']<lower_bound]))
```

```
q1: 10.4455
q3: 19.38
IQR: 8.9345
Upper Bound: 32.78175
Lower Bound: -2.9562500000000007
Skewed data: 8
Skewed data: 0
```

```
# To handle outliers transformation techniques are used.
def transformationPlot(feature):
    plt.figure(figsize=(12,5))
    plt.subplot(1,2,1)
    sns.distplot(feature)
    plt.subplot(1,2,2)
    stats.probplot(feature, plot=plt)
```

```
transformationPlot(df['Na_to_K'])
```




```
df['Na_to_K']=np.log(df['Na_to_K'])
```

```
# 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']]

# Replacing normal and high cholesterol with 0 & 1
df['Cholesterol'] = [0 if x=='NORMAL' else 1 for x in df['Cholesterol']]

# Replacing female and male with 0 & 1
df['Sex'] = [0 if x=='F' else 1 for x in df['Sex']]

df.head()
```

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	0	2	1	3.232976	DrugY
1	47	1	0	1	2.572078	drugC
2	47	1	0	1	2.313921	drugC
3	28	0	1	1	2.053867	drugX
4	61	0	0	1	2.892758	DrugY

Splitting data into train and test

```
x = df.drop('Drug', axis=1)
y = df['Drug']
```

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=10)
```

```
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,)
```

Model Building

```
def decisionTree(x_train, x_test, y_train, y_test):
    dt=DecisionTreeClassifier()
    dt.fit(x_train,y_train)
    yPred = dt.predict(x_test)
    print('***DecisionTreeClassifier***')
    print('Confusion matrix')
    print(confusion_matrix(y_test,yPred))
    print('Classification report')
    print(classification_report(y_test,yPred))
```

```
def randomForest(x_train, x_test, y_train, y_test):
    rf = RandomForestClassifier()
    rf.fit(x_train,y_train)
    yPred = rf.predict(x_test)
    print('***RandomForestClassifier***')
    print('Confusion matrix')
    print(confusion_matrix(y_test,yPred))
    print('Classification report')
    print(classification_report(y_test,yPred))
```

```
def KNN(x_train, x_test, y_train, y_test):
    knn = KNeighborsClassifier()
    knn.fit(x_train,y_train)
    yPred = knn.predict(x_test)
    print('***KNeighborsClassifier***')
    print('Confusion matrix')
    print(confusion_matrix(y_test,yPred))
    print('Classification report')
    print(classification_report(y_test,yPred))
```

```
def xgboost (x_train, x_test, y_train, y_test):
    xg = GradientBoostingClassifier()
    xg.fit(x_train,y_train)
    yPred = xg.predict(x_test)
    print('***Gradient BoostingClassifier***')
    print('Confusion matrix')
    print(confusion_matrix(y_test,yPred))
    print('Classification report')
    print(classification_report (y_test, yPred))
```

```
def compareModel(x_train, x_test, y_train, y_test):
    decisionTree(x_train, x_test, y_train, y_test)
    print('-'*100)
    randomForest(x_train, x_test, y_train, y_test)
    print('-'*100)
    KNN(x_train, x_test, y_train, y_test)
    print('-'*100)
    xgboost(x_train, x_test, y_train, y_test)
```

```
***DecisionTreeClassifier***
Confusion matrix
[[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]]
Classification report
              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
```

```
***RandomForestClassifier***
Confusion matrix
[[25  0  0  0  0]
 [ 0  7  0  0  0]
 [ 0  2  4  0  0]
 [ 0  0  0  6  1]
 [ 0  0  0  0 15]]
Classification report
              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      0.86      0.92         7
       drugX       0.94      1.00      0.97        15

    accuracy                           0.95        60
   macro avg       0.94      0.90      0.91        60
weighted avg       0.96      0.95      0.95        60

***KNeighborsClassifier***
Confusion matrix
[[18  2  1  0  4]
 [ 6  0  0  0  1]
 [ 3  0  2  0  1]
 [ 5  0  0  0  2]
 [10  1  1  1  2]]
Classification report
              precision    recall  f1-score   support

       DrugY       0.43      0.72      0.54        25
       drugA       0.00      0.00      0.00         7
       drugB       0.50      0.33      0.40         6
       drugC       0.00      0.00      0.00         7
       drugX       0.20      0.13      0.16        15

    accuracy                           0.37        60
   macro avg       0.23      0.24      0.22        60
weighted avg       0.28      0.37      0.30        60
```

```
***Gradient BoostingClassifier***
Confusion matrix
[[25  0  0  0  0]
 [ 0  7  0  0  0]
 [ 0  2  3  0  1]
 [ 0  0  0  6  1]
 [ 0  0  0  0 15]]
Classification report
              precision    recall  f1-score   support

       DrugY       1.00      1.00      1.00        25
       drugA       0.78      1.00      0.88         7
       drugB       1.00      0.50      0.67         6
       drugC       1.00      0.86      0.92         7
       drugX       0.88      1.00      0.94        15

    accuracy                           0.93        60
   macro avg       0.93      0.87      0.88        60
weighted avg       0.94      0.93      0.93        60
```

```
# Decision tree and Random forest performs well
from sklearn.model_selection import cross_val_score
```

```
# Random forest model is selected
rf = RandomForestClassifier()
rf.fit(x_train,y_train)
yPred=rf.predict(x_test)
```

```
f1_score (yPred, y_test, average='weighted')
```

```
0.9516222084367246
```

```
cv = cross_val_score(rf,x,y,cv=5)
np.mean(cv)
```

```
0.985
```

```
pickle.dump(rf,open('model.pkl','wb'))
```