**Data Analytic for Business Decision-Making**

**Final Project**

**DATA 1202-DATA ANALYSIS TOOLS ANALYTICS**

**Submitted to: Sk.Md.Mizanaur Rahman**



**Submitted by: -**

**Bhavneet Singh (100949210)**

**Devyang Parekh(100900886)**

**Dhiren Bhoyar (100906276)**

**Mandeep Singh (100920923)**

**Rudraksh Thakre (100949552)**

**Date: April 18, 2024**

Contents

[INTRODUCTION 3](#_Toc164355293)

[Exploratory Data Analysis 4](#_Toc164355294)

[EDA – Conducting Basic Data Analysis: 9](#_Toc164355295)

[Preparing Data and Feature Engineering 13](#_Toc164355296)

[Label Encoding 14](#_Toc164355297)

[Train-Test Split 15](#_Toc164355298)

[Model Implementation 16](#_Toc164355299)

# A pile of pills and capsules Description automatically generated**INTRODUCTION**

Drug prescription is a critical aspect of healthcare management, with the choice of medication playing a pivotal role in patient outcomes and overall well-being. This report addresses the importance of accurate drug prediction, recognizing its significance in optimizing treatment plans and enhancing therapeutic efficacy. By harnessing machine learning methodologies, we embark on a journey to develop predictive models to determine the most suitable drug for individuals based on a myriad of factors. Through analyzing a comprehensive dataset encompassing demographic, clinical, and lifestyle variables, our objective is to unearth meaningful patterns and insights that can guide clinicians in making informed decisions regarding drug prescriptions. Our efforts are geared towards advancing the field of predictive analytics in healthcare, with the ultimate goal of facilitating personalized treatment strategies and improving patient care outcomes.

In this project, our objective is to predict the most suitable drug type for individuals based on a dataset comprising diverse health-related features. By leveraging machine learning techniques, we aim **to develop precise predictive models** that can aid healthcare professionals in prescribing the most effective medications tailored to individual patient profiles.

**Objectives:**

* Dataset exploration using various types of data visualization.
* Build various ML models that can predict drug type.

**The machine learning models used in this project are:**

* K Neighbours
* Random Forest
* Linear Support Vector Machine (SVM)

# **Exploratory Data Analysis**

We are Conducting Exploratory Data Analysis (EDA) which is a critical phase in the data analysis process that involves examining and visualizing the structure, patterns, and relationships within a dataset. The primary objective of EDA is to gain insights into the data, uncover potential patterns or trends, identify outliers or anomalies, and generate hypotheses for further investigation. Through a combination of statistical techniques and data visualization methods, EDA facilitates the exploration and understanding of complex datasets, enabling data scientists and analysts to make informed decisions and formulate hypotheses for subsequent modeling or analysis tasks. EDA plays a pivotal role in the data exploration phase, serving as a preliminary step towards more advanced analytics, modeling, and interpretation of results.

**Read Data and PreCheck:** Initially, we'll load the dataset and conduct preliminary checks to ensure data integrity and cleanliness, identifying any missing values or outliers that may require handling.

**Variable Description**

* Age: Age of the patient, represented as an integer variable.
* Sex: Gender of the patient, categorized as male (M) or female (F), stored as an object (categorical) variable.
* BP: Blood pressure of the patient, categorized as LOW, NORMAL, or HIGH, stored as an object (categorical data).
* Cholesterol: Cholesterol level of the patient, categorized as NORMAL or HIGH, stored as an object (categorical data).
* Na\_to\_K: Sodium to Potassium ratio in the patient's blood, represented as a float variable.
* Drug: Type of drug prescribed to the patient, stored as an object (categorical) variable.

The dataset comprises 200 rows and 6 columns, indicating a relatively small dataset suitable for analysis and modeling tasks.

A screenshot of a computer

Description automatically generated

**Age:**

**Age Range:**

* The maximum age recorded in the dataset is 74 years.
* The minimum age recorded in the dataset is 15 years.
* A graph with a blue line

  Description automatically generatedThis indicates that the dataset covers a wide age range, spanning from adolescence to older adulthood.

**Sex Variable:**

* The dataset consists of 104 male (M) and 96 female (F) individuals.
* This indicates a relatively balanced distribution of gender in the dataset, with a slight majority of males.

A blue rectangular object with white text

Description automatically generated

**Blood Pressure Variable:**

Blood Pressure Distribution:

* 77 individuals with high blood pressure (HIGH)
* 64 individuals with low blood pressure (LOW)
* 59 individuals with normal blood pressure (NORMAL)

A blue bar graph with white text

Description automatically generated

**Cholesterol Variable:**

Cholesterol Distribution:

* 103 individuals with high cholesterol levels (HIGH)
* 97 individuals with normal cholesterol levels (NORMAL)

A blue rectangular object with white lines

Description automatically generated

**Na\_to\_K Variable:**

Na\_to\_K Range:

* The maximum, minimum, and mean values of the sodium-to-potassium ratio are computed using descriptive statistics
* The maximum sodium-to-potassium ratio recorded in the dataset is 38.247, while the minimum ratio is 6.269.
* The mean sodium to potassium ratio is 16.084485.

A graph with a blue line

Description automatically generated

**Drug Variable:**

Drug Distribution:

* 91 instances of DrugY
* 54 instances of drugX
* 23 instances of drugA
* 16 instances each of drugC and drugB

A graph with blue rectangles

Description automatically generated

# **EDA – Conducting Basic Data Analysis:**

**Age - Drug:**

**A graph showing different colored dots

Description automatically generated**

* DrugB: Taken only by individuals older than 51 years old.
* DrugA: Taken only by individuals younger than 50 years old.

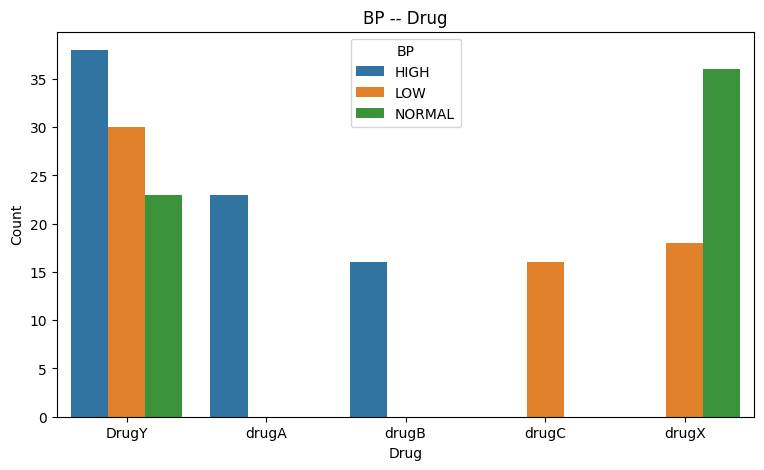
**Sex - Drug:**

**A graph of a drug

Description automatically generated with medium confidence**

* Male people get drugA, drugB and drugC more than male people.
* Female people get DrugY more than female people.
* drugX seems equal for male and female people.
* According to this graph, Sex feature is not an important feature for classification.

**BP - Drug:**

****

* drugA and drugB are got only by people who have HIGH blood pressure.
* drugC is got by people who have LOW blood pressure.
* drugX is got by people who have HIGH blood pressure.
* BP is an important feature for classification.
* DrugY:
* HIGH BP: Count is 38.
* NORMAL BP: Count is 23.
* LOW BP: Count is 30.
* drugA:

HIGH BP: The count is 23.

* drugB:

HIGH BP: The count is 16.

* drugC:

LOW BP: The count is 16.

* drugX:

LOW BP: The count is 18.

NORMAL BP: Count is 36.

**A graph showing different colored dots

Description automatically generatedNa\_to\_K - Drug:**

People who have Na\_to\_K ratio is bigger than 15, get DrugY.

We can create a new feature from here.

**Cholesterol - Drug:**

**A graph of a drug

Description automatically generated**

* drugC is got by people who have HIGH cholesterol.
* Cholesterol is an important feature to classify drugC

**Na\_to\_K - BP - Drug:**

**A graph of different colored dots

Description automatically generated with medium confidence**

* If people have HIGH blood pressure and Na\_to\_K ratio is lower than 15 , they get drugA and drugB only.
* If people have LOW blood pressure and Na\_to\_K ratio is lower than 15 , they get drugC only.

# **Preparing Data and Feature Engineering**

We discuss the process of preparing the dataset for modeling and feature engineering. We’ll focus on creating a new feature called Na\_to\_K\_Bigger\_Than\_15, which plays a crucial role in classifying the drug category.

**New Feature: Na\_to\_K\_Bigger\_Than\_15**

The Na\_to\_K\_Bigger\_Than\_15 feature is designed to identify whether the sodium-to-potassium ratio (Na\_to\_K) exceeds a threshold of 15. Here’s how it works:

* If Na\_to\_K is greater than or equal to 15, the feature value is set to 1 (indicating true).
* Otherwise, the feature value is set to 0 (indicating false).

**Impact of Na\_to\_K\_Bigger\_Than\_15**

The newly engineered feature, Na\_to\_K\_Bigger\_Than\_15, has significant implications for drug classification. Specifically:

* When Na\_to\_K exceeds 15, it is always associated with DrugY.
* Other drugs (such as drugA, drugB, drugC, and drugX) do not consistently exhibit this pattern.

**Visualization**

To visualize the impact of this feature, we create a bar plot showing the distribution of Na\_to\_K\_Bigger\_Than\_15 across different drugs:

A graph of a drug

Description automatically generated

**Conclusion**

The Na\_to\_K\_Bigger\_Than\_15 feature provides valuable information for drug classification, particularly in identifying DrugY. Its inclusion in predictive modeling can enhance accuracy and contribute to better patient outcomes.

# **Label Encoding**

In the data preprocessing phase, we perform label encoding to convert categorical variables into numerical representations. This transformation ensures that machine learning algorithms can work effectively with the data. Let’s discuss the label encoding process for the following features:

**Sex:**

* Original values: ‘F’ (female) and ‘M’ (male)
* Encoded values: 0 (female) and 1 (male)

**BP (Blood Pressure):**

* Original values: ‘LOW’, ‘NORMAL’, and ‘HIGH’
* Encoded values: 0, 1, and 2

**Cholesterol:**

* Original values: ‘NORMAL’ and ‘HIGH’
* Encoded values: 0 and 1

**Na\_to\_K (Sodium-to-Potassium Ratio):**

* The continuous numerical feature remains unchanged.

**Na\_to\_K\_Bigger\_Than\_15:**

* This new binary feature indicates whether the sodium-to-potassium ratio is greater than or equal to 15.
* Encoded values: 0 (false) and 1 (true)

**Drug:**

* Original values: ‘DrugY’, ‘drugA’, ‘drugB’, ‘drugC’, and ‘drugX’
* Encoded values: 0, 1, 2, 3, and 4

The label encoding process allows us to represent categorical information in a format suitable for machine learning models. It ensures that the algorithms can learn from these features effectively.

A screenshot of a computer

Description automatically generated

# **Train-Test Split**

In the process of building a machine learning model, it is essential to divide the dataset into training and testing subsets. The purpose of this split is to evaluate the model’s performance on unseen data. Let’s discuss the train-test split:

**Data Split:**

* We divided the dataset into two parts:
  + Training Data: Consists of 80% of the original data.
  + Test Data: Comprises the remaining 20% of the data.
* This split ensures that the model learns from the training data and is then evaluated on the test data.

**Randomization:**

* We set a random seed (random\_state = 42) to ensure reproducibility.
* The shuffle parameter is set to True, which randomly shuffles the data before splitting.

**Shapes:**

* The dimensions of the data subsets are as follows:
  + x\_train: Shape (160, 6) - Training features (excluding the target variable ‘Drug’).
  + x\_test: Shape (40, 6) - Testing features.
  + y\_train: Shape (160, 1) - Training target labels (reshaped to a single column).
  + y\_test: Shape (40, 1) - Testing target labels (reshaped to a single column).

# **Model Implementation**

We explore the implementation of three different machine learning models for drug prediction: K-Nearest Neighbors (KNN), Random Forest, and Support Vector Machine (SVM). For each model, we employ the GridSearchCV method to fine-tune hyperparameters and ensure optimal performance. Additionally, we utilize 5-fold cross-validation to mitigate the impact of randomness and ensure robustness in our model evaluation.

**K-Nearest Neighbors (KNN) Classifier**

The KNN classifier is a non-parametric method used for classification tasks. We initially train the KNN model with default parameters and evaluate its performance using both training and testing datasets. Subsequently, we perform hyperparameter tuning using GridSearchCV to identify the optimal values for parameters such as the number of neighbors (n\_neighbors), distance metric (p), and weight function (weights).

After hyperparameter tuning, the KNN model demonstrates improved performance, with a training score of 75.63% and a testing score of 70.00%. Notably, the KNN classifier exhibits the lowest performance among the three classifiers evaluated.

**Default Performance:**

• Training Score: 59.38%

• Testing Score: 65.00%

**GridSearchCV Performance:**

• Best Hyperparameters:

• Number of Neighbors (n\_neighbors): 10

• Distance Metric (p): 1 (Manhattan distance)

• Weight Function (weights): 'distance'

• Training Score: 75.63%

• Testing Score: 70.00%

**Conclusion:**

The KNN classifier exhibits moderate performance by default, with a training score of 59.38% and a testing score of 65.00%. After hyperparameter tuning using GridSearchCV, the model's performance improves significantly, achieving a training score of 75.63% and a testing score of 70.00%.

**Confusion Matrix:**

The matrix is read as follows:

* The diagonal cells (from top left to bottom right) show the number of correct predictions for each drug type.
* Off-diagonal cells indicate misclassifications, where the predicted drug type does not match the actual drug type.

For example:

* A cell with a true label of 0 and a predicted label of 0 with a count of 15 means that there are 15 instances where ‘DrugY’ was correctly predicted.
* A cell with a true label of 1 and a predicted label of 4 with a count of 5 indicates that ‘drugA’ was incorrectly predicted as ‘drugX’ 5 times.

The color scale on the right side of the matrix helps to quickly identify the counts, with darker shades representing higher numbers of instances.

A graph of a graph with numbers and a number

Description automatically generated with medium confidence

**Random Forest Classifier**

The Random Forest classifier is an ensemble learning method based on decision tree classifiers. We begin by training the Random Forest model with default parameters and assess its performance using cross-validation. Subsequently, we conduct hyperparameter tuning via GridSearchCV to identify the optimal number of trees (n\_estimators) and criterion for splitting (criterion).

Upon hyperparameter optimization, the Random Forest model achieves superior performance, with a training score of 98.75% and a testing score of 97.50%. The Random Forest classifier emerges as the top-performing model, showcasing high accuracy and robustness in drug prediction.

**Default Performance:**

• Training Score: 98.13%

• Testing Score: 97.50%

**GridSearchCV Performance:**

• Best Hyperparameters:

• Number of Estimators (n\_estimators): 100

• Criterion: 'entropy'

• Training Score: 98.75%

• Testing Score: 97.50%

**Conclusion**:

The Random Forest classifier demonstrates excellent performance both by default and after hyperparameter tuning. With default parameters, the model achieves a training score of 98.13% and a testing score of 97.50%, which further improves to 98.75% training score and 97.50% testing score after hyperparameter tuning.

**Confusion Matrix**

* **True Labels (Actual)**: The rows represent the actual drug types, with ‘DrugY’ encoded as 0, ‘drugA’ as 1, ‘drugB’ as 2, ‘drugC’ as 3, and ‘drugX’ as 4.
* **Predicted Labels**: The columns represent the drug types predicted by the model, using the same encoding.
* A cell at the intersection of row 0 (True Label: ‘DrugY’) and column 0 (Predicted Label: ‘DrugY’) with a high value would mean that the model correctly predicted ‘DrugY’ many times.
* A cell at the intersection of row 1 (True Label: ‘drugA’) and column 4 (Predicted Label: ‘drugX’) with a non-zero value would indicate that the model incorrectly predicted ‘drugA’ as ‘drugX’ for that number of instances.

A graph of a graph with numbers and a number

Description automatically generated with medium confidence

**Support Vector Machine (SVM) Classifier**

The SVM classifier is a powerful algorithm for both classification and regression tasks. Initially trained with default parameters, we evaluate the SVM model's performance using cross-validation. We then utilize GridSearchCV to fine-tune hyperparameters such as the regularization parameter (C), kernel type (kernel), degree of the polynomial kernel (degree), and kernel coefficient (gamma).

Following hyperparameter tuning, the SVM model demonstrates significant improvement, achieving a training score of 98.75% and a testing score of 97.50%. The SVM classifier exhibits comparable performance to the Random Forest model, highlighting its efficacy in drug prediction.

* **Default Performance**:
  + **Training Score**: 71.25%
  + **Testing Score**: 65.00%
* **GridSearchCV Performance**:
  + **Best Hyperparameters**:
    - Regularization Parameter (C): 1
    - Kernel: 'linear'
    - Degree: 1
    - Gamma: 0.01
  + **Training Score**: 98.75%
  + **Testing Score**: 97.50%

**Conclusion**:

The SVM classifier's default performance is moderate, with a training score of 71.25% and a testing score of 65.00%. However, after hyperparameter tuning, the model's performance improves significantly, achieving a training score of 98.75% and a testing score of 97.50%.

**Confusion Matrix:**

* A cell with a true label of 0 and a predicted label of 0 with a high count indicates many instances of class 0 were correctly classified as class 0.
* A cell with a true label of 3 and a predicted label of 0 with a count of 1 indicates one instance of class 3 was incorrectly classified as class 0.

A diagram of a model confusion matrix

Description automatically generatedThe goal is to have high numbers along the diagonal (correct classifications) and low numbers off the diagonal (misclassifications). The matrix provides insight into which classes are being confused with others, which can inform further model tuning.

# **Conclusion**

Upon comparing the performance of the three classifiers, we observe that the Random Forest classifier and SVM classifier, after hyperparameter tuning, outperform the KNN classifier. Notably, both the Random Forest and SVM models achieve high accuracy scores of 97.50%, demonstrating their effectiveness in predicting strokes. These findings underscore the importance of employing advanced machine-learning techniques for accurate and reliable drug prediction.

1. **Model Performance Comparison**:
   * **Random Forest Outperforms**: Among the three models evaluated, the Random Forest classifier demonstrates the highest performance, achieving consistently high accuracy scores both with default parameters and after hyperparameter tuning.
   * **SVM Follows Closely**: The Support Vector Machine (SVM) classifier also exhibits commendable performance, especially after hyperparameter tuning, with comparable accuracy scores to the Random Forest model.
   * **KNN Lags Behind**: In contrast, the K-nearest neighbors (KNN) classifier lags in terms of performance, achieving comparatively lower accuracy scores even after hyperparameter tuning.
2. **Effectiveness in Drug Prediction**:
   * **Random Forest and SVM**: Both the Random Forest and SVM classifiers demonstrate effectiveness in predicting drugs, with high accuracy scores exceeding 97.50% after hyperparameter tuning.
   * **KNN Reliability**: While the KNN classifier's performance improves after hyperparameter tuning, its reliability in drugs prediction remains lower compared to Random Forest and SVM models.
3. **Considerations for Model Selection**:
   * **Task Complexity**: Depending on the task complexity and dataset characteristics, the choice of model may vary. For drug prediction tasks with structured data, ensemble methods like Random Forest or robust algorithms like SVM may offer better performance.
   * **Computational Efficiency**: While Random Forest and SVM classifiers demonstrate high accuracy, their computational complexity may be higher compared to KNN, which may impact scalability for larger datasets.
   * **Interpretability**: KNN provides straightforward interpretability, making it suitable for tasks where model transparency is crucial. However, its performance may be compromised in complex classification tasks.

**Default Train Score:**

A graph with different colored bars

Description automatically generated

**Default Test Score :**

A graph of different colored bars

Description automatically generated

**We have also predicted value using our final model and we get output like this**

A screenshot of a computer

Description automatically generated

**Reference:**

* 1. Arezalo. (2023). Drug Classification: EDA, KNN, NB, Logistic.Kaggle. <https://www.kaggle.com/code/arezalo/drug-classification-eda-knn-nb-logistic>
  2. SaiyanPrince91. (2024, April 18). DC-COLLEGE GitHub.

<https://github.com/SaiyanPrince91/DC-COLLEGE>