# Explanation of what classification is

Supervised learning, a subset of artificial intelligence and machine learning, relies on human guidance to accurately label and train algorithms (Mitchell, 1997). This method involves training the machine using labelled input and output data to analyse the training data and make accurate predictions for new and unseen data. By leveraging historical data, the algorithm can learn patterns and classify various file types, such as images, words, and documents, enabling it to predict outcomes for new data (Hastie et al., 2009). Supervised learning is advantageous in generating reliable data output and addresses real-world computational problems (Mitchell, 1997). It also holds significance in developing business applications. However, training the model can be time-consuming, particularly when dealing with large datasets, which can pose additional challenges (Mitchell, 1997). One of the categories of supervised learning is classification.

Classification is a fundamental concept in machine learning and data analysis that involves categorizing or labelling data into predefined classes or categories. It is a supervised learning technique where the goal is to learn a mapping between input variables (features) and output variables (class labels) based on a given set of training data. In the context of machine learning, classification refers to the process of training a model on a labelled dataset to make predictions or decisions about the class or category of unseen or future instances. The labelled dataset consists of input features and their corresponding class labels, which serve as examples for the model to learn from.

The classification process typically involves several steps by Raza (2017):

1. Data Pre-processing: This step involves preparing the data for classification by handling missing values, normalizing, or standardizing features, and performing other necessary data transformations.
2. Feature Selection or Extraction: It is important to identify the relevant features that contribute to the classification task. Feature selection techniques help to choose the most informative features, while feature extraction methods create new features from the existing ones to improve classification performance.
3. Model Training: In this step, a classification algorithm is applied to the labelled training data to create a model. The model learns the patterns and relationships between the input features and the class labels.
4. Model Evaluation: The trained model is evaluated using a separate set of data called the test set. Evaluation metrics such as accuracy, precision, recall, and F1 score are calculated to assess the performance of the model in predicting the correct class labels.
5. Model Deployment: Once the model is deemed satisfactory, it can be deployed to make predictions on new, unseen data. The model takes the input features of a new instance and predicts its class label based on what it has learned during the training phase.

There are several reasons why classification is widely employed.

Firstly, classification enables automated decision-making based on patterns and relationships present in the data. By training a model on labelled data, we can predict the class or category of new, unseen instances, allowing us to make informed decisions or take appropriate actions. Secondly, classification provides a means of organizing and structuring data. By assigning class labels to instances, we can group similar data together and gain a better understanding of the underlying patterns and characteristics within each class. This organization facilitates data analysis and can lead to valuable insights and discoveries.

Furthermore, classification plays a vital role in various real-world applications. It is used in spam email filtering, sentiment analysis, credit scoring, fraud detection, medical diagnosis, image recognition, and many other domains. By accurately classifying data, we can automate tasks, improve efficiency, and enhance decision-making processes in numerous fields. Moreover, classification serves as a foundation for more advanced machine learning techniques. Many complex algorithms and models, such as ensemble methods like random forests and gradient boosting, rely on classification as a building block. By mastering classification, we can build upon this knowledge and explore more sophisticated techniques.

In the field of machine learning, there are various classification algorithms available, including decision trees, random forests, support vector machines (SVM), logistic regression, and neural networks. Each algorithm has its own strengths, weaknesses, and assumptions, making them suitable for different types of classification tasks.

Classification is utilized by various individuals and organizations across different domains and industries. Some examples of users who would benefit from classification include Data scientists, machine learning engineers, researchers across various fields, business analysts, and fraud analysts all utilize classification techniques to analyse data, build predictive models, and gain insights. These professionals employ classification algorithms to solve a wide range of problems, make data-driven decisions, categorize, and analyse data, understand patterns, make predictions, draw meaningful conclusions, gain insights into customer behaviour and market segmentation, assess product performance, detect fraudulent activities, and develop robust fraud detection systems, especially in the financial industry.

Classification is used in various scenarios where there is a need to categorize or label data into distinct classes or categories. It is employed when we want to automate decision-making processes based on patterns and relationships within the data. Classification finds its application in spam email filtering, where emails are classified as either spam or legitimate. It is used in sentiment analysis to classify text as positive, negative, or neutral based on the expressed sentiment. Classification is also utilized in credit scoring, where borrowers are categorized into different creditworthiness classes. In medical diagnosis, classification helps in identifying diseases or conditions based on patient symptoms and medical test results. Furthermore, classification is used in image recognition tasks, such as identifying objects in photos or videos. In essence, classification is employed whenever there is a need to assign labels or make predictions based on patterns in the data, spanning across industries like finance, healthcare, marketing, and more.

# Why the drugs dataset

The chosen data set, "Drug Classification," is appropriate for analysis with classification for several reasons. Firstly, the dataset contains information about certain drug types, making it relevant for predicting the outcome of drugs for patients. This aligns with the goal of the analysis, which is to utilize machine learning techniques to predict the accurate drug type for a given patient.

Secondly, the dataset includes various features that are commonly considered in medical decision-making. These features, such as age, sex, blood pressure levels (BP), cholesterol levels, and the sodium-to-potassium ratio, provide valuable information that can potentially influence the choice of drug type. By incorporating these features into the classification model, we can explore their relationships with the target variable (drug type) and gain insights into the factors that contribute to the selection of specific drugs.

Furthermore, the dataset is suitable for beginners in machine learning, as mentioned in the description. This implies that the dataset and the accompanying problem statement are designed to provide a learning opportunity and a chance to practice classification techniques. By working with this dataset, beginners can gain hands-on experience in pre-processing the data, selecting appropriate features, training classification models, and evaluating their performance.

# Analysis that will be conducted

The purpose of the analysis in this project is to develop a machine learning model that can predict the appropriate type of drug for a patient based on their general information and diagnosis. By exploring and analysing the given dataset, the objective is to gain insights into the relationship between various patient attributes (such as age, gender, blood pressure, cholesterol levels, and sodium to potassium ratio) and the type of drug prescribed.

The analysis aims to leverage different types of data visualization techniques to visually explore the dataset and understand the distributions, patterns, and potential correlations between the variables. This exploration can provide valuable insights into the characteristics and trends within the data.

Furthermore, the project involves building and evaluating various machine learning models, including K-Nearest Neighbours, and naive Bayes (categorical and Gaussian). These models are trained using the provided dataset to learn the patterns and relationships between patient attributes and drug types.

The goal of this analysis is to create a reliable and accurate predictive model that can assist medical professionals in making informed decisions regarding the most suitable type of drug for a patient. By automating the prediction process, healthcare professionals can potentially save time and improve the efficiency of drug prescription, leading to better patient outcomes and personalized treatment plans.

# Steps of the process

The steps that will be conducted on the data set involves building and evaluating machine learning models for drug classification based on patient information and diagnosis. The objective is to predict the outcome of the drug type that might be suitable for a patient. To achieve this, the following steps are performed:

* Data Exploration and Visualization: The data set will be explored using various visualization techniques to gain insights into the distribution and relationships between variables. This may include creating histograms, bar plots, scatter plots, or correlation matrices to understand the data's characteristics.
* Data Preparation: The data will be prepared by performing data binning, involves dividing continuous numerical variables into discrete bins or intervals, one-hot encoding, used to convert categorical variables into numerical representations that can be easily understood by machine learning algorithms, and applying the SMOTE technique for handling imbalanced data. This step ensures that the data is in a suitable format for modelling.
* Model Building: Two different classification models will be built: Naïve Bayes Classifier and K-Nearest Neighbours (KNN) algorithm. The Naïve Bayes Classifier is a probabilistic model that assumes independence between features, while KNN is a distance-based algorithm that classifies new instances based on the majority class of its neighbours.
* Model Evaluation: The performance of the models will be evaluated using evaluation metrics such as accuracy, precision, recall, F1-score, and support. These metrics provide insights into the models' ability to correctly classify the drug types and their overall predictive power.
* Model Comparison: The accuracy and other evaluation metrics of the Naïve Bayes Classifier and KNN algorithm will be compared to determine which model performs better on the given data set. This comparison helps in selecting the most suitable algorithm for drug classification.
* Model Improvement: Techniques such as hyperparameter tuning, and feature selection may be applied to improve the models' performance. This involves adjusting the models' parameters or selecting the most relevant features to enhance the models' predictive capabilities.

Through this analysis, we aim to develop accurate and reliable models for drug classification based on patient information. The insights gained from the data exploration, model evaluation, and comparison will provide valuable information for healthcare professionals in predicting suitable drug types for patients.

# Model evaluation and improvement

The model evaluation is a crucial step in assessing the performance of the machine learning models and determining their effectiveness in solving the drug classification problem. By applying the theory correctly and incorporating appropriate techniques, we can significantly improve the model's performance. Here is a description of the model evaluation process:

Evaluation Metrics: Various evaluation metrics are used to measure the model's performance. These include accuracy, precision, recall, and F1-score. Precision is the ratio of true positives to the total predicted positives. It represents the model's ability to correctly identify positive samples. Higher precision values indicate fewer false positives. Recall, also known as sensitivity or true positive rate, is the ratio of true positives to the total actual positives. It measures the model's ability to correctly identify positive samples. Higher recall values indicate fewer false negatives. The F1-score is the harmonic mean of precision and recall. It provides a balance between precision and recall, where higher values indicate better performance. Support refers to the number of samples in each class. It represents the distribution of classes in the test dataset. Each metric provides insights into different aspects of the model's performance, such as overall accuracy, the ability to correctly identify positive instances, and the balance between precision and recall.

Hyperparameter Tuning: Hyperparameter tuning involves optimizing the model's hyperparameters to find the best configuration that maximizes performance. Techniques like grid search or random search are applied to explore different combinations of hyperparameters and identify the optimal set that improves the model's performance.

Feature Selection: Feature selection techniques are used to identify the most relevant features that contribute to the model's predictive power. By selecting a subset of informative features, the model can focus on the most discriminative aspects of the data, reducing noise and improving performance.

By applying these strategies, we can significantly improve the performance of the model. The evaluation process helps us identify the model's strengths and weaknesses, fine-tune its parameters, select the most informative features, and leverage ensemble methods to achieve better predictive performance. Ultimately, the goal is to build a highly accurate and reliable model that can effectively classify drug types based on patient information.

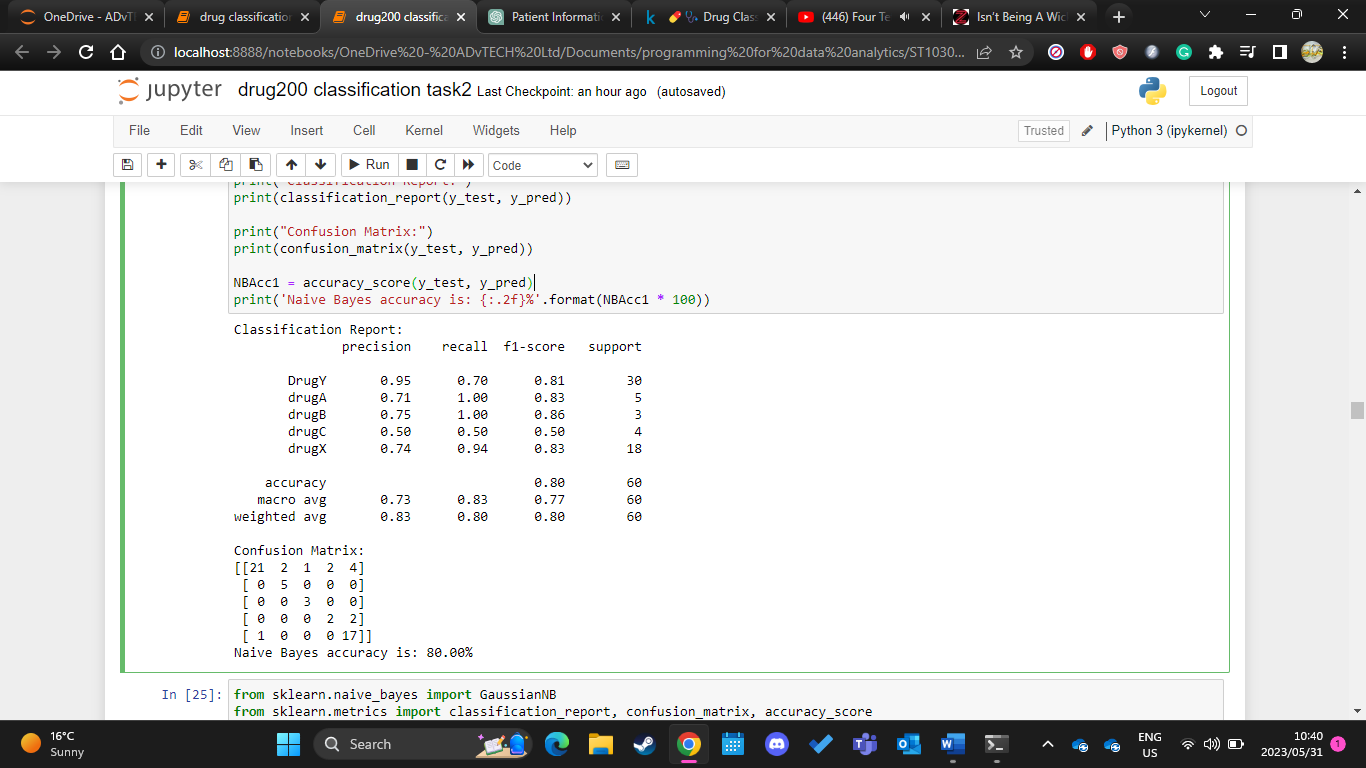
## Gaussian Naïve bayes theorem model (Before improvements)

A screenshot of a computer

Description automatically generated

* Precision: The classifier achieved the highest precision for drugA (1.00) and drugX (1.00), indicating that all instances predicted as drugA and drugX were correct. The precision for drugB was 0.75, indicating that 75% of the instances predicted as drugB were correct. The precision for DrugY and drugC was lower at 0.64 and 0.67, respectively.
* Recall: The highest recall was observed for drugC (1.00), indicating that all actual instances of drugC were correctly predicted. The recall for DrugY and drugB was also relatively high at 0.90 and 1.00, respectively. However, the recall for drugA was lower at 0.20, and for drugX, it was 0.39.
* F1-score: The F1-scores were highest for drugB (0.86) and drugC (0.80), indicating a good balance between precision and recall for these classes. The F1-score for DrugY was 0.75, and for drugX, it was 0.56. However, the F1-score for drugA was lower at 0.33.
* Support: Most instances belonged to DrugY (30 samples), followed by drugX (18 samples), drugA (5 samples), drugC (4 samples), and drugB (3 samples).
* Accuracy: The overall accuracy of the classifier was 0.70 (or 70%), indicating the proportion of correctly predicted instances out of the total number of instances.

## Categorical naïve bayes theorem (before improvement)



* Precision is a measure of how many correctly predicted instances belong to a particular class. The classifier achieved high precision for DrugY (0.95), drugA (0.71), drugB (0.75), and drugX (0.74). However, the precision for drugC was relatively low at 0.50.
* Recall, also known as sensitivity, indicates the ability of the classifier to correctly identify positive instances. The classifier achieved high recall for drugA, drugB, and drugX (all above 0.90). However, the recall for DrugY (0.70) and drugC (0.50) was comparatively lower.
* The F1-score is the harmonic mean of precision and recall and provides a balanced measure of the classifier's performance. The F1-scores were relatively high for DrugY, drugA, drugB, and drugX (ranging from 0.81 to 0.86). However, the F1-score for drugC was lower at 0.50.
* The support indicates the number of samples for each class in the test set. It shows that many instances belonged to DrugY (30 samples) and drugX (18 samples), while drugA, drugB, and drugC had smaller sample sizes.
* The overall accuracy of the classifier was 0.80 (or 80%), indicating the proportion of correctly predicted instances out of the total number of instances.

In summary, the classifier performed well in terms of precision, recall, and F1-score for most classes, except for drugC. The support indicates the distribution of samples among the classes. The overall accuracy of 80% suggests a reasonably accurate prediction capability of the Naive Bayes classifier for this task.

**Overall, the Gaussian Naive Bayes classifier in this case has lower performance compared to Categorical Naïve bayes theorem**

## KNN (before improvement)

A screenshot of a computer

Description automatically generated

* Precision: The classifier achieved the highest precision for drugB (1.00), indicating that all instances predicted as drugB were correct. The precision for DrugY and drugX was relatively high at 0.79 and 0.75, respectively. However, the precision for drugA and drugC was lower at 0.50.
* Recall: The highest recall was observed for drugA (1.00), indicating that all actual instances of drugA were correctly predicted. The recall for drugX was also relatively high at 0.83. However, the recall for DrugY, drugB, and drugC was lower, ranging from 0.63 to 0.67.
* F1-score: The F1-scores were highest for drugB and drugX at 0.80 and 0.79, respectively. The F1-score for drugA was 0.67, indicating a reasonable balance between precision and recall. However, the F1-scores for DrugY and drugC were lower at 0.70 and 0.50, respectively.
* Support: Many instances belonged to DrugY (30 samples) and drugX (18 samples), followed by drugA (5 samples), drugC (4 samples), and drugB (3 samples).
* Accuracy: The overall accuracy of the classifier was 0.72 (or 72%), indicating the proportion of correctly predicted instances out of the total number of instances.

In summary, the classifier exhibited relatively high precision, recall, and F1-scores for drugB, indicating accurate predictions for this class. However, there were variations in performance for the other classes, with lower scores for drugA, DrugY, and drugC. The support metric provides insight into the distribution of samples among the classes. The overall accuracy of 72% suggests a moderately accurate prediction capability of the classifier for this task.

For the model improvement I’ve decided to use hyperparameter tuning instead of feature selection because it provided a better result that is more applicable to these models. In addition, I’ll be only using the categorical naïve bayes theorem model because it has a better overall performance.

## Categorical Naïve bayes theorem model (after hyperparameter tuning)

A screenshot of a computer

Description automatically generated

* In terms of precision, the hyperparameter-tuned model achieved slightly higher precision for DrugY (0.82) compared to the previous model (0.95). The precision for drugA and drugC improved to 1.00, indicating that all predicted instances for these classes were correct. The precision for drugB remained the same at 0.75, while drugX achieved perfect precision in both models (1.00).
* Regarding recall, the hyperparameter-tuned model showed improved recall for DrugY (0.93) compared to the previous model (0.70). The recall for drugA, drugB, and drugC also improved, with all achieving perfect recall in the hyperparameter-tuned model. The recall for drugX remained high in both models (0.91 and 0.94).
* The F1-scores for most classes also saw improvements in the hyperparameter-tuned model. DrugY, drugA, drugB, and drugC all achieved higher F1-scores compared to the previous model. DrugX, which had a high F1-score in the previous model (0.83), achieved an even higher score of 0.95 in the hyperparameter-tuned model.
* The support values indicate the number of samples for each class. It appears that the sample sizes for each class remained the same between the two models.
* The overall accuracy of the hyperparameter-tuned model improved to 0.90 (90%) compared to the previous model's accuracy of 0.80 (80%). This suggests that the hyperparameter tuning enhanced the classifier's ability to predict the correct drug type across all classes.

## KNN (after hyperparameter tuning)

A screenshot of a computer

Description automatically generated

* Before tuning, the precision values for DrugY, drugA, drugB, drugC, and drugX were 0.79, 0.50, 1.00, 0.50, and 0.75, respectively. After tuning, the precision values changed to 0.68, 0.83, 1.00, 1.00, and 0.70, respectively. The precision for DrugY and drugX decreased slightly after tuning, while the precision for drugA, drugB, and drugC improved.
* In terms of recall, the before-tuning recall values for DrugY, drugA, drugB, drugC, and drugX were 0.63, 1.00, 0.67, 0.50, and 0.83, respectively. After tuning, the recall values changed to 0.87, 0.83, 0.67, 0.60, and 0.64, respectively. The recall for DrugY and drugC improved after tuning, while the recall for drugA, drugB, and drugX decreased slightly.
* The F1-scores for DrugY, drugA, drugB, drugC, and drugX before tuning were 0.70, 0.67, 0.80, 0.50, and 0.79, respectively. After tuning, the F1-scores changed to 0.76, 0.83, 0.80, 0.75, and 0.67, respectively. The F1-score for DrugY and drugC improved after tuning, while the F1-scores for drugA, drugB, and drugX decreased slightly
* The support values indicate the number of samples for each class. The support remained the same between the before and after tuning scenarios.
* The overall accuracy before tuning was 0.72 (72%), which increased to 0.75 (75%) after tuning. This suggests that the tuning process improved the overall accuracy of the KNN classifier.

# Reference

Bahbah, I. (2020, Version 1). drug200 Dataset, Version 1. Retrieved May 26, 2023, from Kaggle: <https://www.kaggle.com/datasets/ibrahimbahbah/drug200>