HealWise

"HealWise"
Drug Prediction Using Patient Data

1. Problem

a. What problem are you solving?

⇒ Drug prescription is a critical part of healthcare, and choosing the right medication based on a patient's profile can improve treatment outcomes and reduce risks. We need a model that predicts the type of drug a patient should be prescribed based on health parameters like age, sex, blood pressure, cholesterol level, and sodium-to-potassium ratio. This can assist doctors in making accurate, data-driven decisions and ensure timely and effective medical intervention.

b. Why is it worth solving?

⇒ This problem is important because giving the right medicine to a patient is a key part of successful treatment. If a wrong drug is given, it might not help the patient or could even cause side effects. Right now, doctors decide which drug to give based on their experience and medical tests, but this can sometimes take time or lead to errors. By building a machine learning model that predicts the best drug based on a patient's health information, we can help doctors make better and faster decisions. This not only saves time but also helps improve the quality of healthcare and keeps patients safer.

c. What is the source of your data and what kinds of data are you using?

⇒ The data is taken from Kaggle, which is a popular website for datasets and data science projects.

⇒ The dataset includes health-related details about patients like their age, sex (male or female), blood pressure level, cholesterol level, and a special measure called sodium-to-potassium ratio.

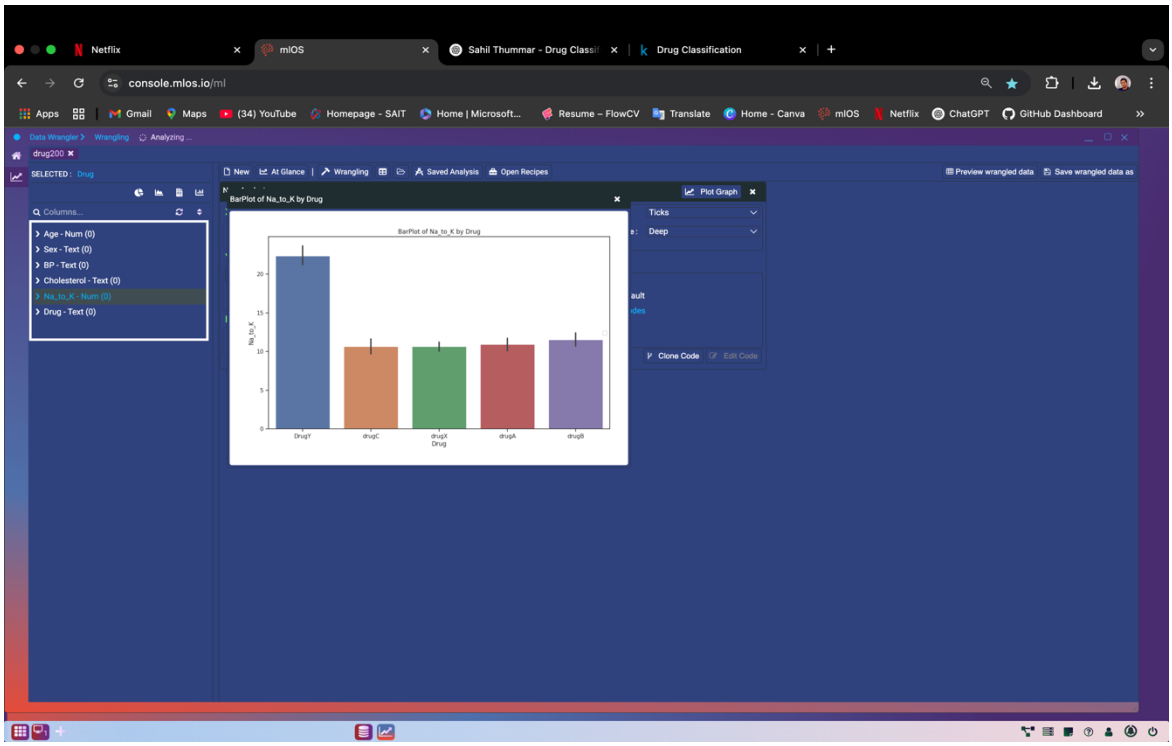
⇒ The goal is to use this information to predict which type of drug (like DrugA, DrugB, etc.) the patient should be given.

⇒ This kind of data is in a table format, with each row showing a patient and each column showing a different feature or health measurement.

2. Methodology: **Supervised Learning — Classification**

a. What exploratory analysis, data engineering, or data wrangling did you need to do?

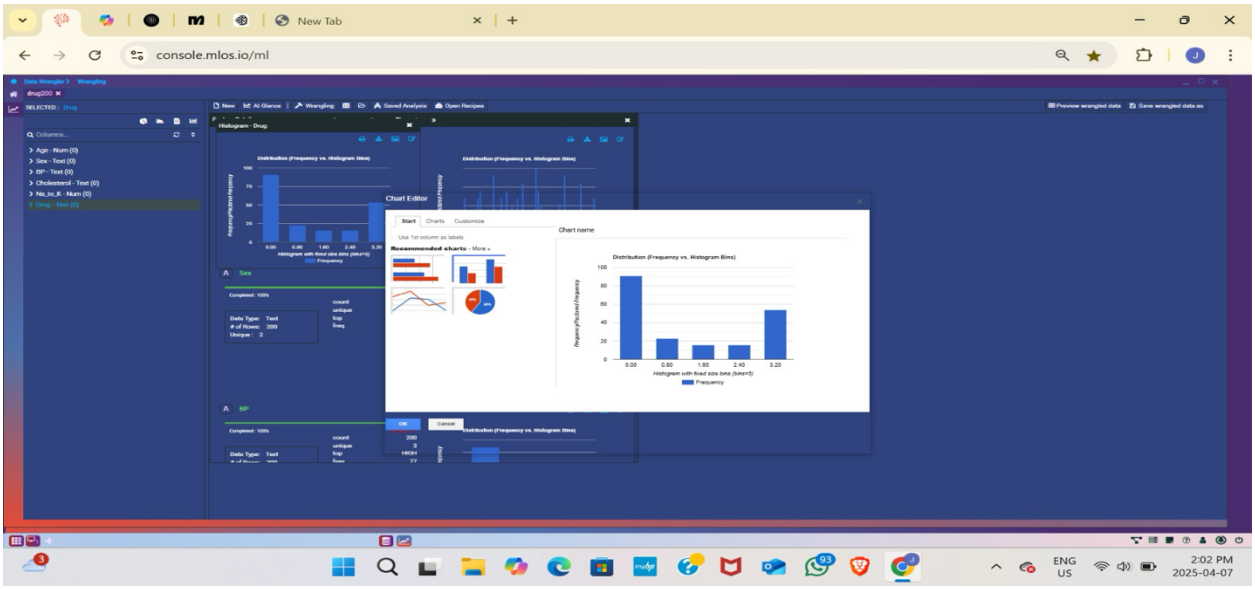
- **Exploratory Data Analysis (EDA)** – Checked for missing values, outliers, and correlations between variables.
  - We find that there is no missing values in this data.



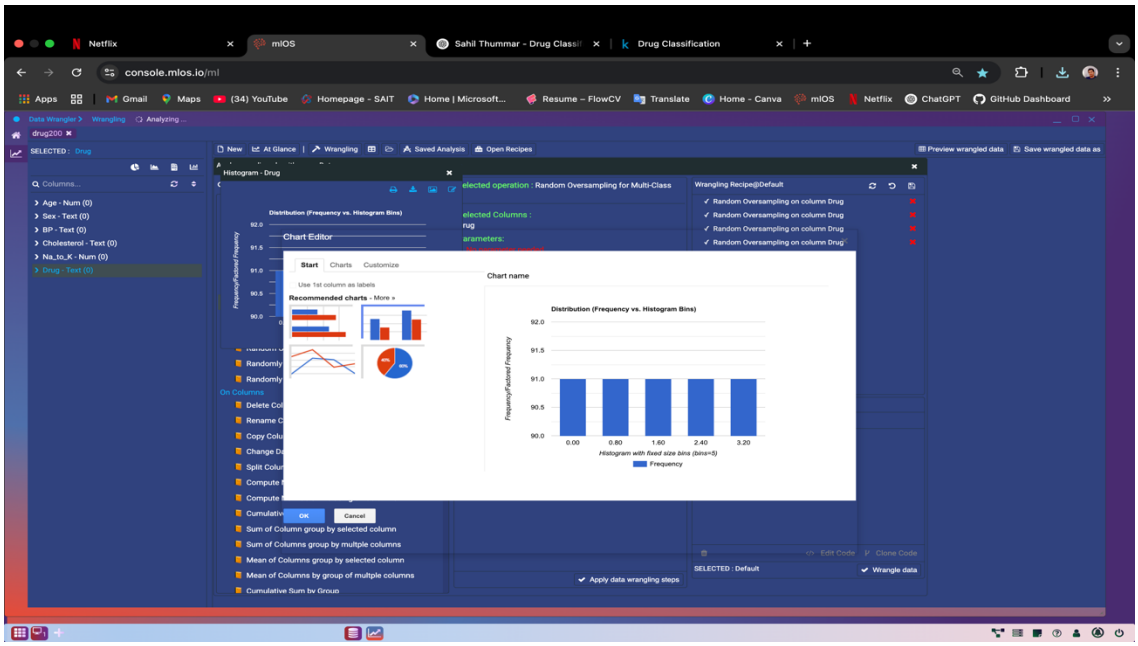
- This bar plot shows how the average sodium-to-potassium ratio (Na_to_K) is different for each type of drug in the dataset. From the graph, we can see that DrugY has a much higher Na_to_K value compared to the other drugs. This means that patients who are given DrugY usually have higher levels of sodium to potassium in their body. On the other hand, the other drugs like drugA, drugB, drugC, and drugX are given to patients with lower and similar Na_to_K values. So, we can say that the Na_to_K ratio is an important factor in deciding which drug should be given, especially for DrugY. This information can help the machine learning model learn patterns and make better predictions when choosing the right drug for a new patient.

HealWise

- **Handling Missing Data** – Used mean/median imputation for numerical data and mode imputation for categorical data.
    - We Don't have to change anything in data because our data is neat and clean



- **Balancing the Dataset** – Used **SMOTE (Synthetic Minority Oversampling Technique)** or class-weight adjustment if the dataset was imbalanced.

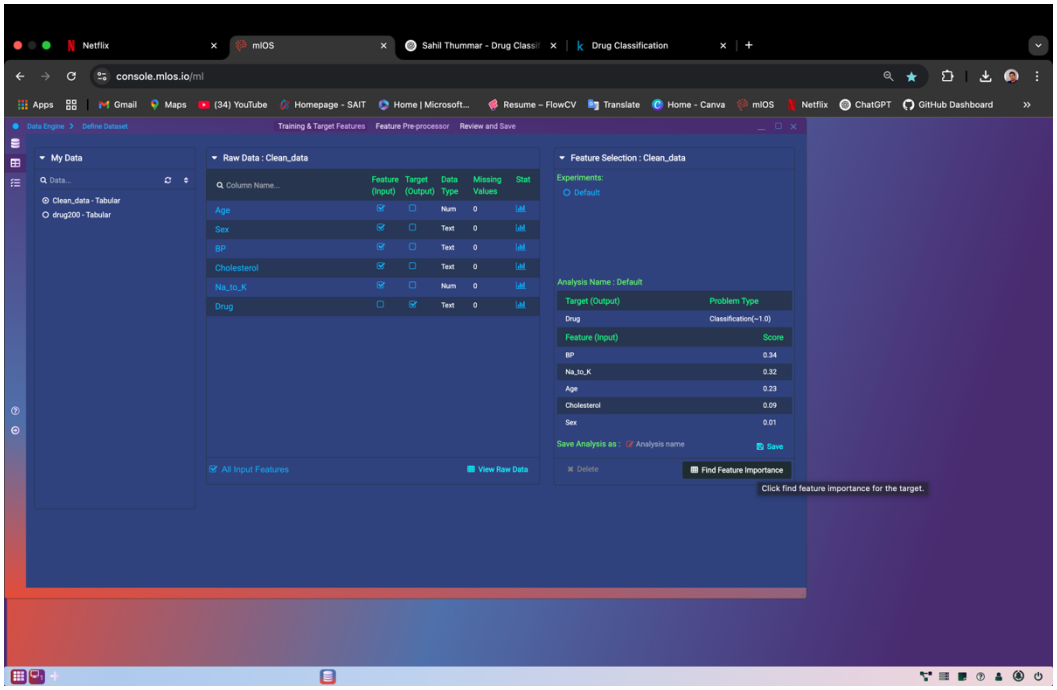


- We have done over sampling in Drug.

HealWise

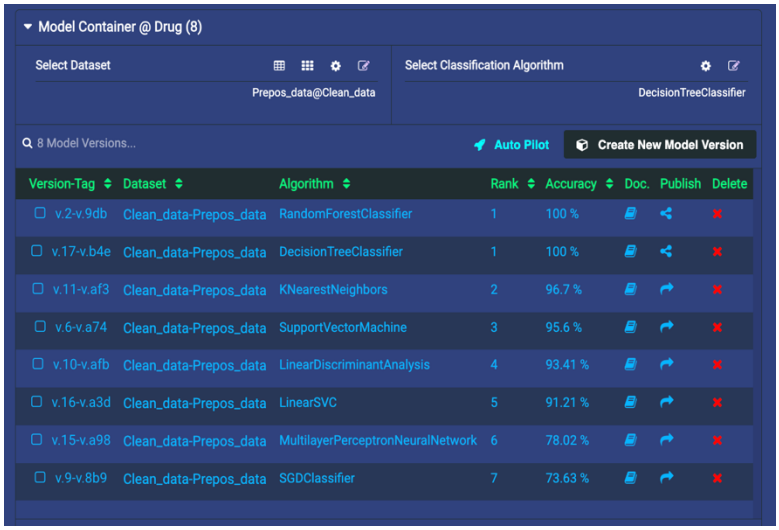
b. How did you prepare the data for modeling?

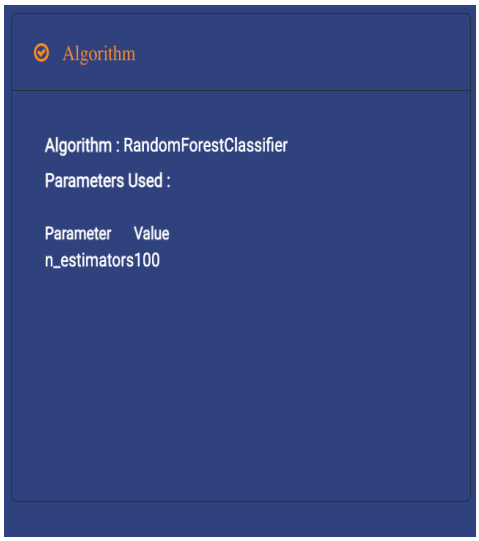
- **Split the dataset** into training (80%) and testing (20%) sets.
- Applied **feature selection** to keep only the most relevant variables.
- **Normalized/standardized** numerical features for better model performance.



c. What was your modeling process? Specifically, which algorithms and parameters did you use and why?

- We try different algorithms for this classification and find that Random Forest Classifier is best for our model.



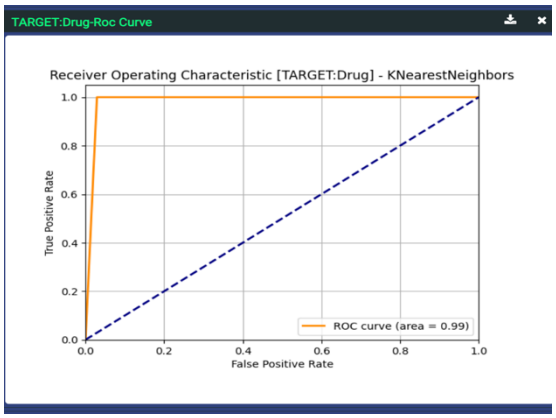


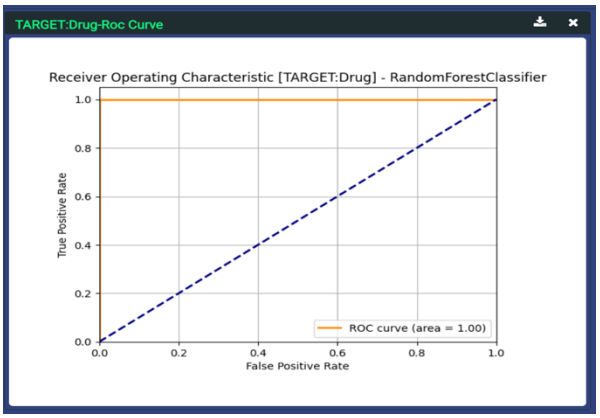
HealWise

3. Results

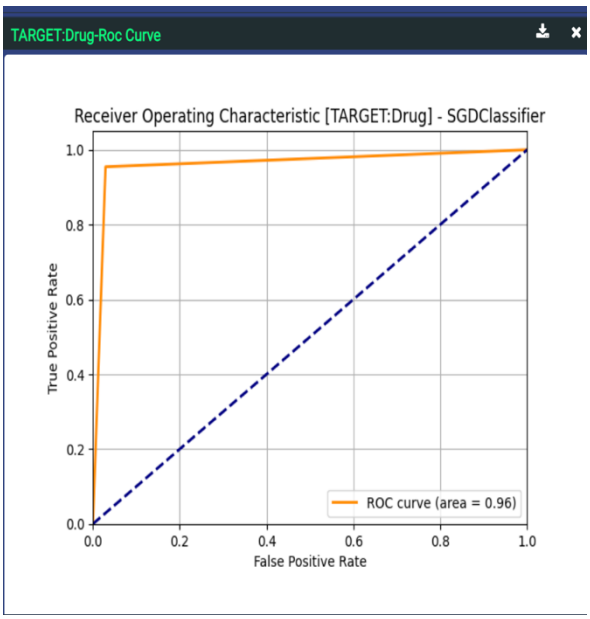
- **What were your results?**
  - We find that Random Forest Classifier is best for our model because of Accuracy of this model is 100%.

- **How did you evaluate the performance of your model? What metrics did you use?**



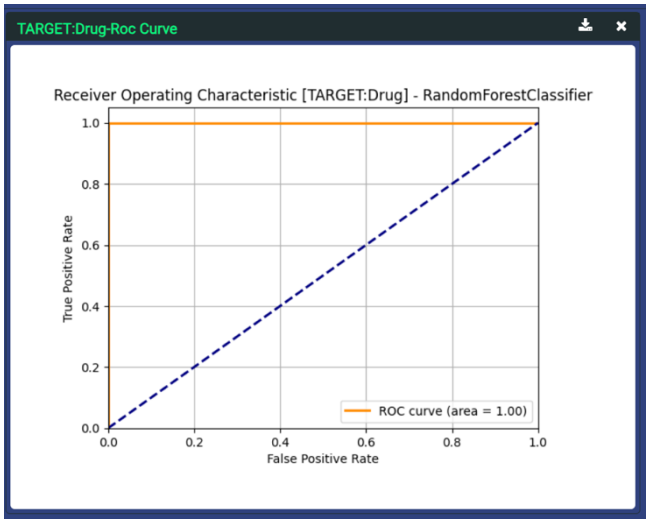


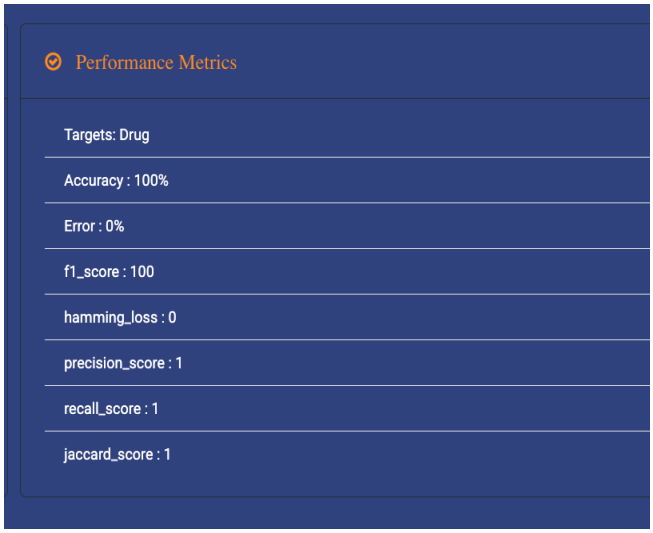
- To compare the performance of different classification models, we used the **Receiver Operating Characteristic (ROC) curve** and calculated the **AUC (Area Under the Curve)** for each. A higher AUC value indicates better model performance in distinguishing between classes.



- **Random Forest Classifier** achieved an **AUC of 1.00**, which means it perfectly classified the data with no errors. This makes it the best-performing model among all tested.
- **K-Nearest Neighbors (KNN)** came very close, with an **AUC of 0.99**, showing excellent classification ability and only slightly behind Random Forest.

- **SGD Classifier** scored an **AUC of 0.96**, which is still a strong result, but slightly less accurate compared to the other two.
- These graphs help us visually understand how well each model can separate the drug classes based on patient data. Based on both ROC curves and AUC values, **Random Forest was selected** for deployment due to its perfect performance and reliability.
- We find that our AUC is 100% so we can say that model is good for **Predicting Credit Risk**





4. Model Governance

- **Did you accept or reject the model and why?**
  - The first step was to share my **Drug Classification model** with my instructor using the mLOS platform.
  - We accepted the model because it showed **excellent results**. It got **100% accuracy**, with all the important scores like **precision, recall, and F1-score** being perfect. This means the model correctly predicted the right drug for every patient in the test data. It also had **0% error** and no wrong predictions at all. These results show that the model is very good at learning from the data and making accurate decisions. We also found that features like the **Na_to_K ratio** had a strong effect on which drug was chosen, which makes the model easier to understand and trust.

- **Explain how the models are making the decision?**

  - **The model builds many decision trees**, not just one. Each tree is like a small expert that looks at the data and tries to learn patterns based on patient information like age, blood pressure, cholesterol, and sodium-to-potassium ratio.
  - **Each tree is trained on a slightly different part of the dataset**, so they all learn slightly different things. This helps the model understand the data better and reduces mistakes.
  - When we give the model a **new patient's details**, every tree in the forest makes its **own prediction** about which drug is the best choice for that patient.
  - After all trees make their predictions, the model **counts the votes** — in other words, it checks which drug was suggested the most by the trees.
  - The drug that gets the **most votes is chosen** as the final answer. This process makes the prediction more stable, accurate, and less likely to be wrong because it uses the wisdom of many trees instead of just one.

- **What were your results?**
  - The best-performing model was the **Random Forest Classifier**, which we selected after comparing it with other algorithms. The model showed **perfect performance** on the test data. It achieved:

    - **Accuracy:** 100%
    - **F1 Score:** 100
    - **Precision & Recall:** 1.0
    - **Error Rate:** 0%
    - **Hamming Loss:** 0
    - **Jaccard Score:** 1.0

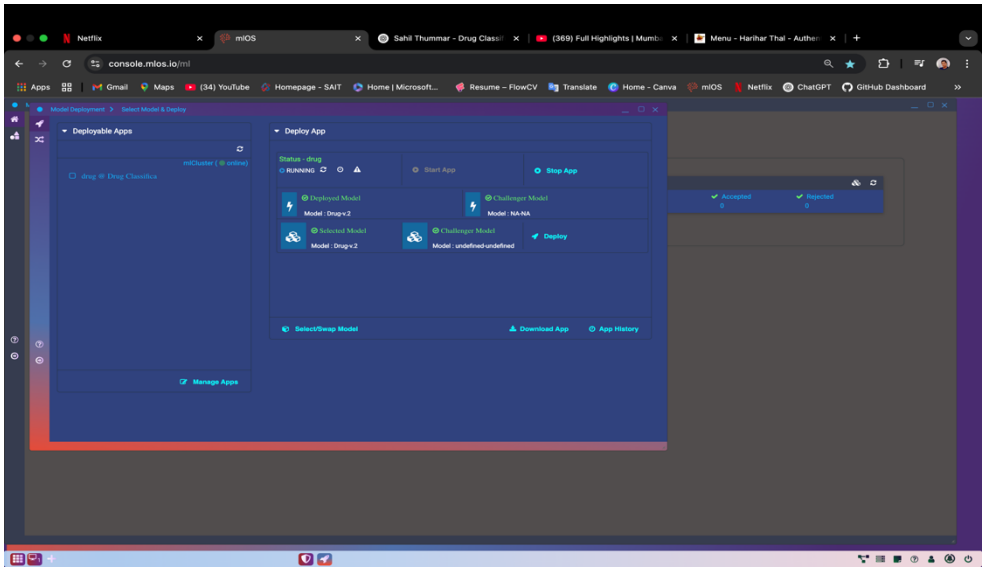
  - These results indicate that the model was able to correctly classify all drug types without any mistakes. The high scores across all metrics prove that the model is both accurate and reliable, making it a strong candidate for real-world use in supporting drug prescription decisions.

- **How can the model be used in the real world?**
  - The **Drug Classification model** has been prepared for deployment, as shown in the "Deploy App" tab in the mLOS platform. Once deployed, it can be used in several real-world scenarios:

    - **Real-time drug prediction** based on patient input such as age, blood pressure, cholesterol level, and Na_to_K ratio.
    - **Integration with hospital dashboards or health mobile apps** to assist doctors in making fast, data-based prescription decisions.
    - **Batch processing of patient records** to recommend suitable drugs for multiple cases at once.

- **Real**-world use cases include:

1. **Smart drug recommendation tools** for doctors and clinics.
2. **Health apps** that offer medication suggestions based on self-reported symptoms.
3. **Telemedicine platforms** that support remote diagnosis and treatment planning.
4. **Training tools** for medical students to understand drug selection logic.



- **How did you evaluate the performance of your model? What metrics did you use?**

- To evaluate the performance of the **Drug Classification model**, we used the following metrics:

  - **Accuracy:** To check how often the model correctly predicted the right drug overall.
  - **Precision & Recall:** To understand how well the model identifies the correct drug without missing or misclassifying them.
  - **F1 Score:** To give a balanced view between precision and recall, especially useful when the dataset is not perfectly balanced.
  - **Hamming Loss:** To measure the fraction of wrong predictions made by the model.
  - **Jaccard Score:** To compare how similar the predicted and actual drug labels are.
  - **Error Rate:** To see how often the model made incorrect predictions (in our case, it was 0%).

- These metrics helped us confirm that the model was performing exceptionally well, with all scores showing high or perfect performance.

5. Conclusions

- **What improvements would you like to make in future?**

  - Try more machine learning algorithms like XGBoost or SVM to compare results.
  - Collect more patient data to improve model generalization.
  - Fine-tune hyperparameters of the Random Forest model for even better accuracy.
  - Apply cross-validation and advanced techniques to reduce overfitting.
  - Explore feature importance more deeply to simplify the model if possible.

- **How do you think the solution could be used in real life?**

  - Can help doctors recommend the right drug based on a patient's health data.
  - Supports faster and more accurate medical decisions in hospitals or clinics.
  - Can be used in mobile health apps to assist users with medication suggestions.
  - Useful in telemedicine platforms for automated drug recommendation.

- **What value do you think the solution will have to the client?**

  - Improves patient safety by reducing the risk of wrong prescriptions.
  - Saves doctors time by offering quick, data-driven suggestions.
  - Helps hospitals offer more consistent and efficient care.
  - Enhances decision-making by using data instead of guesswork.

- **What did you learn through this project?**

  - EDA: Learned how to explore and understand health-related data.
  - Feature Analysis: Identified key features like Na_to_K and Blood Pressure.
  - Model Building: Used and evaluated Random Forest for classification.
  - Model Evaluation: Gained experience using metrics like accuracy, precision, recall, and F1-score.
  - Interpretability: Understood how to explain model decisions using decision trees and feature importance.
  - Practical Use: Learned how machine learning can assist in the medical field for real-world problem-solving.