AI for Drug Interaction and Patient Safety

Current Situation and Motivation

When prescribing new medications, healthcare providers must be cautious about potential adverse interactions with a patient's pre-existing conditions or other medications. Identifying these interactions is essential for ensuring patient safety. However, detecting such issues can be complex, especially when dealing with large patient populations and diverse medical histories.

Using historic patient data and explainable machine learning techniques, AI can assist in determining which patient groups may safely receive a specific drug and which groups are at risk of adverse effects.

Your Task

The pharmaceutical corporation "PharmCo" has hired your team to help them identify which one of their newly developed treatments are safe for public release. The three new treatments work very well on patients suffering from an inexplicable and severe heart condition "H", when the patients do not have any prior conditions. Patients with additional chronic diseases apart from "H", however often experience very heavy side effects.

The treatments have shown exceptional improvements in quality of life and life expectancy for the patients, suggesting that even moderately severe side effects can be tolerated.

Your challenge is to develop an AI model that evaluates the safety of several drugs based on a dataset of patient medical histories, pre-existing conditions, and medications. The goal is to identify potential adverse interactions between the drugs, other pre-existing conditions, and other medications.

Using explainable machine learning (XAI) methods, your model should recommend which patient groups with prior conditions can safely receive the drug and which groups are at high risk of adverse interactions. Your presentation should clearly explain why the drug is or isn't suitable for certain patient groups.

You will be provided with a dataset representing a large group of patients, including their demographic information, medical conditions, medications, and health outcomes after taking the drug.

Helpful Hints and Tips

- Think about which columns in the dataset should be the prediction targets (hint: we suggest several targets). Based on this, you can create a recommendation which patient groups are safe to use the drugs, and which drugs are the safest.
- Use methods of explainable machine learning (e.g., SHAP values, LIME, feature importance) to justify your model's decisions.
- Consider the role of various patient factors (e.g., age, chronic conditions, co-medications) in determining adverse interactions.
- While you are not required to submit a fully functional application, your approach should be well-documented and easy to understand.

Evaluation Criteria

Accuracy of Recommendations: How well your model identifies patient groups that are at risk from the drug. A clear recommendation which drug is safe to release for whom.
Explainability: The clarity and depth of the model's explanations for its predictions.
Use of Data: How effectively the dataset was analysed and utilised.
Patient Safety: How effectively does your model reduce the risk of adverse interactions for the recommended patient groups.
Presentation Style: Professionalism and clarity in presenting your findings.

Deliverables

- **Presentation**: 5-minute pitch on stage, followed by two jury questions.
- **Work product**: Presentation slides detailing the model's approach and results, including visualizations of explainability (e.g., SHAP plots, LIME explanations).
- **Code**: Submit your code as a .zip (TeamName_Code.zip) archive, ensuring it includes the explainable ML methods you used.
- **Results:** You will need to predict multiple target classes. For each target class (e.g., "abc," "def," "ghi"), please follow the instructions below:
  - **Prediction Format**: Ensure that the predictions for each target class are in a separate numpy array file. Example: abc.npy, def.npy, ghi.npy (the file name should be the target class (all in lower cases) and file type should be .npy)
  - **Submission Structure**: Put all the .npy file into a .zip file named TeamName_Results.zip