Deliverable 2

Team Member: Shiqi Tan, Theresa Quan, Angela Hu

1. **Problem Statement**

   The project's goal is to develop a reliable medicine search platform that offers a cost-effective and timely alternative to traditional doctor appointments. Our focus is on designing a user-friendly search interface, enabling individuals to input their symptoms and receive insights into potential medical conditions they might be experiencing. Following this, our system will suggest appropriate medications for their conditions.

2. **Dataset we currently use:**

   From Symptom to Drugs(disease tags included): MIMIC III
   https://drive.google.com/file/d/1968Yw3rxdAC3gc2leyarPHc6jCOfufqn/view?usp=sharing

   From Symptom to Disease:
   https://drive.google.com/file/d/1wLTx7udOfZQ_VLYtYx5a2s7zLnc2oiVb/view?usp=share_link
   https://drive.google.com/file/d/1yYuiwlAExQIY0lW-Fj2c-cWYivBzIDUu/view?usp=share_link
   https://drive.google.com/file/d/1vmGaf7LEUbTwfdm0uUO5Qk-9vWopOHfG/view?usp=share_link

   From Disease to Drugs:
   https://drive.google.com/file/d/1BsBOIJCXHbRmvcQJPcM-vQw5WCwXl2rQ/view?usp=share_link
   https://drive.google.com/file/d/16vFGhTY9mGL4iOHSTMuzD1-1VYaY02qL/view?usp=share_link

   Data Preprocessing:

   1. Content of the dataset(number of samples, labels, etc)

   - The dataset contains information related to medical symptoms and their associated diagnoses and treatments.
   - We are working with a subset of 2500 rows of the original dataset.
   - We specifically focus on three columns: ‘symptom’, ‘NDC’, and ‘ICD9_CODE’.
     - symptom: lists the symptoms described by patients.
     - NDC: denotes specific drug codes.
     - ICD9_CODE: represents diagnosis codes.

   2. Describe and justify data processing methods

   - Subsetting the data: we choose a subset of 2500 rows of data from MIMIC III dataset

- Column Selection: we select three columns: ‘symptom’, ‘NDC’, and ‘ICD9_CODE’.
- Data Parsing: using the ‘ast.literal_eval()’ function, we parsed the ‘symptom’ and ‘NDC’ columns, implying these columns might have been stored in a string representation of lists. Parsing them ensures they are in their original data format for further processing.
- Text Vectorization: to convert the 'symptom' column, which contains lists of symptoms, into a format suitable for machine learning, we employed the CountVectorizer. The tokenizer argument was modified to suit the list format of our data. This method effectively created a Bag of Words model where each unique symptom corresponds to a feature/column.
- Label Binarization: The 'NDC' column, which represents the drug’s ATC third level codes in a list format, was binarized using the MultiLabelBinarizer. This transformed each unique NDC value into a separate binary column, making it suitable as a target for multilabel classification tasks.

3. *Machine learning model:*

a. Specify the framework and tools that you used to implement your model. (For instance, did you use any libraries such as PyTorch, Keras, etc. to implement the model? Any other tools? What does the architecture of your model look like? How many layers/modules? etc.) Explain and provide architecture graphs as appropriate.

-Our model uses matplotlib for visualization, sklearn and xgboost for machine learning operations. In particular, we uses RandomForestClassifier from sklearn.ensemble, LogisticRegression from sklearn.linear_model, and XGBClassifier from xgboost.

b. Justify any decision about training/validation/test splits, regularization techniques, optimization tricks, setting hyper-parameters, etc.

-Split Ratio: the data is split into a training set and a test set with a 65-35% split respectively. We choose 65-35% split because the datase is large enough and a larger test set is better to evaluate the model's performance.
-Setting Hyper-parameters:
- Random State: The `random_state=24` parameter ensures reproducibility. It means that every time the code is run, the train and test split will produce the same results. This is crucial for consistent experiments.

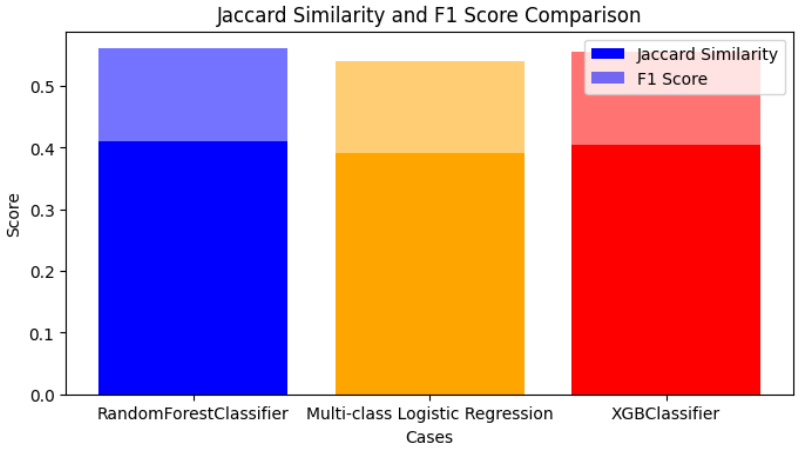
c. Description of validation methods How did you test your model? Is your model overfitting or underfitting?

-We generated various visualizations and metrics to assess its performance:
Proportions of true positives (Correct Predictions for Each Symptom Set).
Overlap between true labels and predicted labels using Venn diagrams.
Confusion matrices for the top 6 drugs.

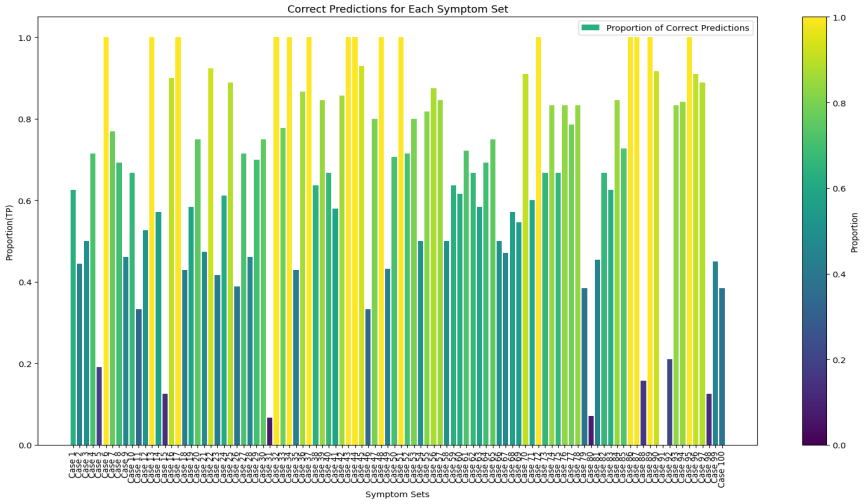
d. Did you face any challenges implementing the model? If so, how did you solve it?

-originally, we tried to use another dataset, but that dataset was too small with only 350 data, and the training result wasn't good, so we changed to MIMIC and redo the whole model.

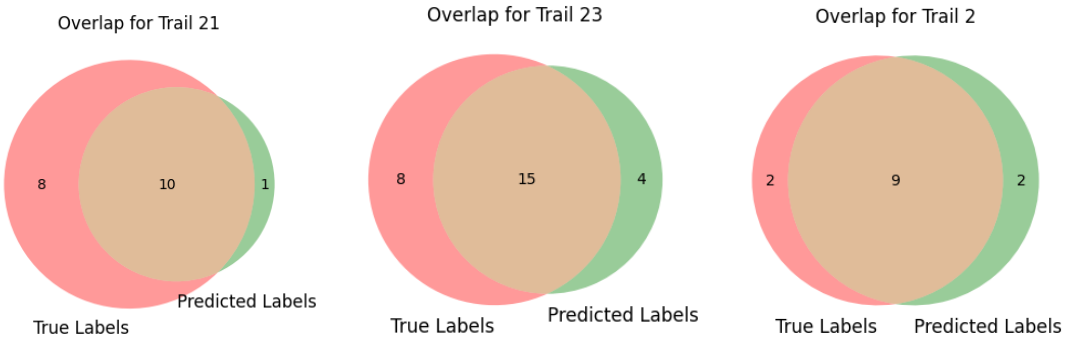
- **Preliminary results**
  - **Jaccard Similarity and F1 Score Comparison between models**
  - After processing 2500 rows of data, our analysis focused on the Jaccard Similarity and F1 Score for the RandomForestClassifier, Multi-class Logistic Regression, and XGBClassifier. The results revealed a Jaccard Similarity of 0.4100, 0.3922, and 0.4039, and F1 Scores of 0.5607, 0.5396, and 0.5552 respectively. Visually represented through bar graphs, these metrics showcased the models' comparative performance.



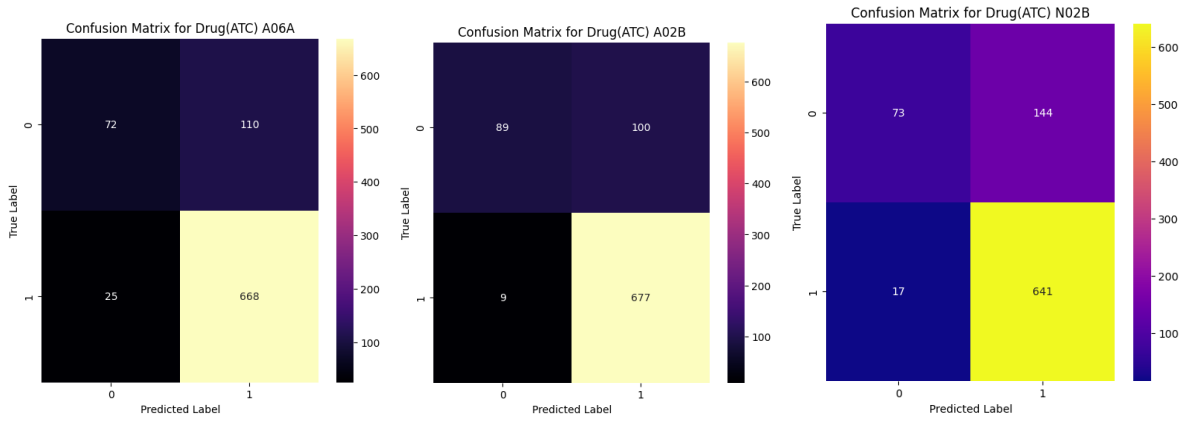
  - The RandomForestClassifier emerged as the leader, demonstrating the highest Jaccard Similarity and F1 Score, indicating its ability to capture intricate data patterns. The XGBClassifier performed competitively, while the Multi-class Logistic Regression fell in between the other two models. These results, while promising, are preliminary and based on a limited dataset. The RandomForestClassifier's proficiency in handling complex data patterns is encouraging for the project's feasibility.
  - **Correct Predictions for Each Symptom Set**

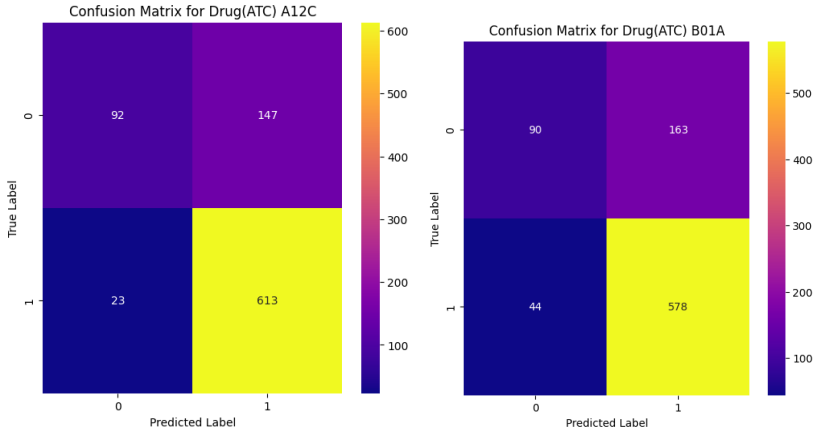


- The graph, showing true positive proportions (0.4 to 0.6) for the first 100 rows, demonstrates our model's consistent and reliable accuracy in predicting suitable drug sets for various symptoms. Few false positives emphasize the model's precision, ensuring trustworthy medical predictions. This reliability is crucial for practical applications.
- **Venn diagrams for randomly selected trails by RandomForestClassifier**



- **Confusion matrix for prediction performance by RandomForestClassifier on Top 5 most common Drugs appeared in dataset, for a total of 875 datapoints in y_test**





- The Confusion matrix and Venn diagrams generated from the RandomForestClassifier's predictions on the top 5 most common drugs in the dataset highlight the model's strong performance. Notably, the high true positive rate serves as a clear indication of the model's proficiency.
- In the context of drug prescription tasks, overlooking essential medications can result in grave repercussions, potentially leading to ineffective treatments or exacerbation of the patient's condition. The relatively low number of false negatives, as evident from the plotted data, underscores the significance of our accurate predictions in avoiding these critical oversights.

- **Next steps:**
  - Our immediate focus involves fine-tuning our models to optimize their accuracy. Hyperparameter tuning, utilizing techniques like Grid Search and Random Search, will be central to this process. We aim to enhance the models' predictive abilities by systematically adjusting parameters and exploring ensembling methods to leverage their collective strengths.
  - **The absence of APIs for mapping drugs' NDC codes to the ATC third level format poses a significant challenge,** especially for the development of our GUI/webapp, crucial for user accessibility.
  - While manual searches are possible through existing databases, utilizing their APIs incurs charges, hindering our ideal functionality.
    - The websites https://www.whocc.no/atc_ddd_index/ and https://go.drugbank.com/atc are great resources for converting ATC codes to drug names. However, we lack the coding expertise needed to extract the specific data we require from these sources.