Problem 1

The development of drugs is critical in providing therapeutic options for patients suffering from chronic and terminal illnesses. “Target Drug”, in particular, is designed to enhance the patient's health and well-being without causing dependence on other medications that could potentially lead to severe and life-threatening side effects. These drugs are specifically tailored to treat a particular disease or condition, offering a more focused and effective approach to treatment, while minimising the risk of harmful reactions.
The objective in this assignment is to develop a predictive model which will predict whether a patient will be eligible*** for “Target Drug” or not in next 30 days. Knowing if the patient is eligible or not will help physician treating the patient make informed decision on the which treatments to give.

Code Overview

Step A: Data Preparation

- The code initializes the sequence_length variable to determine the number of past days to consider for predictions.
- It defines the features list, which includes the relevant features to be used for modeling ('Incident' and 'DaysSinceLastEvent').
- Empty lists, X and y, are created to store the input sequences and corresponding labels.

Step B: Creating Sequences

- The code iterates over each unique patient in the dataset by grouping the data based on 'Patient-Uid'.
- For each patient, the data is sorted chronologically based on the 'Date' column.
- If the patient has enough data points (at least sequence_length), a patient sequence is created.
- The patient sequence includes the last sequence_length values of the selected features.
- The patient's eligibility label for the last entry is added to the y list, representing the target value.
- The patient sequence is added to the X list.

Step C: Splitting the Dataset

- The X and y lists are converted to NumPy arrays for further processing.
- The dataset is split into training and validation sets using the train_test_split function from scikit-learn. The split ratio is 80:20, and a random state of 42 is set for reproducibility.

Step D: Model Architecture

- The code defines a function, create_model, to create the LSTM model using the Keras Sequential API.
- The model consists of an LSTM layer with a specified number of units and an input shape of (sequence_length, len(features)).
- A dropout layer is added to prevent overfitting.

The output layer uses a sigmoid activation function to predict the eligibility (1 or 0) for the "Target Drug".
The model is compiled with binary cross-entropy loss, the Adam optimizer, and accuracy as the metric.

Step E: Hyperparameter Tuning

The KerasClassifier wrapper is used to wrap the model for compatibility with scikit-learn's grid search.
The hyperparameter grid is defined, including the number of units in the LSTM layer and the dropout rate.
The F1-score is selected as the scoring metric for the grid search.
Grid search is performed using GridSearchCV, with 5-fold cross-validation.
The best hyperparameters and the best model are obtained from the grid search results.

Step F: Model Training and Evaluation

The best model is trained using the training set and evaluated on the validation set.
The training process is monitored, and the best hyperparameters and model performance are printed.

Step D: Generating Predictions

The trained model is used to generate predictions for the patients in the test dataset.
Each patient in the test dataset is labeled as 1 or 0 based on the predictions of the model, indicating their predicted eligibility for the "Target Drug."
The final predictions are saved in the final_submission.csv file.

Best model and Various analysis on prediction of the X_val data

- Best hyperparameters: The best hyperparameters found by the grid search are a dropout rate of 0.3 and 128 units in the LSTM layer. These hyperparameters yielded the highest F1 score on the validation set.
- Best score: The best F1 score achieved by the model on the validation set is 0.7520681761806567. This score represents the overall performance of the model in terms of precision and recall.
- Precision: For class 0 (not eligible for "Target Drug"), the precision is 0.86, indicating that 86% of the predicted instances for class 0 are correct. For class 1 (eligible for "Target Drug"), the precision is 0.78, indicating that 78% of the predicted instances for class 1 are correct.
- Recall: For class 0, the recall is 0.90, meaning that 90% of the actual instances of class 0 are correctly identified by the model. For class 1, the recall is 0.71, indicating that 71% of the actual instances of class 1 are correctly identified.
- F1-score: The F1-score is a balanced measure that combines precision and recall. For class 0, the F1-score is 0.88, and for class 1, the F1-score is 0.74. These scores represent the overall performance of the model in terms of correctly identifying instances of each class.
- Accuracy: The overall accuracy of the model is 0.83, indicating that 83% of the predictions made by the model are correct.
- F1-score: The F1-score is 0.744, which represents the balanced measure of precision and recall.

- ROC AUC Score: The ROC AUC score is 0.898, which measures the model's ability to discriminate between the two classes. A score of 0.5 represents a random classifier, while a score of 1.0 represents a perfect classifier.
- Balanced Accuracy Score: The balanced accuracy score is 0.805, which takes into account the imbalance in class distribution. It provides a fair evaluation of model performance when dealing with imbalanced datasets.
- Average Precision Score: The average precision score is 0.822, which measures the model's ability to rank the positive instances higher than the negative instances.
- Kappa Score: The Kappa score is 0.623, which measures the agreement between the predicted and actual classes, considering the possibility of the agreement by chance. A score of 1.0 represents a perfect agreement, while a score of 0.0 represents an agreement by chance.

Conclusion

The developed model exhibits strong predictive performance for determining eligibility for the "Target Drug." It achieves an accuracy of 83.5% and a precision of 78.1%, demonstrating its ability to accurately identify eligible patients. With a recall of 71.0%, the model effectively captures a significant proportion of eligible patients. The F1-score of 0.744 reflects a balanced combination of precision and recall. Additionally, the model achieves a high ROC AUC score of 0.898, indicating its excellent discriminatory power. The balanced accuracy score of 0.805 confirms its reliability across imbalanced classes. With an average precision score of 0.822 and a Kappa score of 0.623, the model consistently ranks positive instances higher and demonstrates substantial agreement with the actual classes. These results validate the model's proficiency in accurately predicting eligibility for the "Target Drug."