Documentation supporting Problem Statement 1

Problem Statement

A. Come up with a positive and negative set for developing the model, here the positive point is the patient who has taken 'Target Drug". Make sure you are also taking into account the time aspect while coming up with a positive & negative set because the aim is to predict 30 days in advance whether a patient is going to be eligible or not.
B. Come up with the right kind of feature engineering for developing your model. The features can be frequency-based, time-based etc. If possible, can also leverage deep learning techniques
C. Evaluate the model on validation set & come up with the right strategy to reduce false positives & false negatives.
D. Once you have developed your predictive model, use your model to generate predictions for patients in test.parquet, each patient in test.parquet should be labelled as 1 or 0 using your predictive model and the final generated predictions should be submitted in final_submission.csv
E. The evaluation metric for the assignment is F1-Score(candidates with the highest F1 score would be prioritised).

Steps I followed to come up with a solution

1. Brainstormed ideas as to how to go about this problem statement and devised a plan with pen and paper.
2. Understood the data and addressed duplication in instances and index and there were no inconsistencies other than that.
3. Derived some inferences from the data by using unique value numbers and the description of data using df.describe() method and have provided my inferences in notebook clearly.
4. Segregated just the incidents related to patient who took target drug and modified it to take into account the condition that we have to train the model in such a way that it says whether a patient is eligible to take target drug or not 30 days prior. Modified in the sense, have removed all incidents up to 30 days prior to first date of target drug administration and then merged it with the original df after removing already existing incidents related to positive set patient. This way we have cleverly handled positive and negative set.
5. Using pandas groupby(), size() and unstack() method, done feature engineering and made 57 columns including the target variable(target drug) and then if patient had target drug atleast once, I put 1 in target drug column or else put 0 and made it a binary classification target variable.
6. Checked for range of values across columns to know whether to scale or not and then did X, y and training, validation split and then scaled the data using standard scaler object so as to make it work even with gradient descent based algorithms. Then I have noticed class imbalance and addressed it by finding best sampling technique for the set we had(Used Oversampler).
7. Now looped through some well known models and chose GradientBoostingClassifier, which comparatively performed well and had negligible overfitting.
8. Did Hyperparameter tuning using GridSearchCV and found the best parameters and used it to build a robust model which has least probability of overfitting and better accuracy and F-1 score.
9. Now treated test set in similar way, how I treated train set and scaled down it and predicted labels for test set and created final_submission.csv
10. This was so challenging and great opportunity to learn and implement new things. I have tried to improve the model performance using PCA in a separate notebook but in vain. I also tried ANN but it didn't perform well.