After exploring the data, and learning the different features and the target I have been able to divide the data to different subgroups that can be handled separately.

I immediately noted that we are in a highly imbalanced problem, and not only that, the target variable Y can be 1 because of different hypertension conditions.
In each subgroup I have handled the nulls by either imputing or removing rows, and I created multiple missingness flags to features I knew that their null might still be meaningful to keep.

I checked the different features distributions, and their correlation to the target variable to note if there are any specific features that I should specifically handle, for having outliers, unnatural distribution or having only 1 value/acting as a unique ID.

After all the exploration and the understanding of the data I have decided on my next steps:

- I will build additional features for each subgroup, especially new features coming from the textual clinical sheet, based on keyword search, and whole text embeddings.
- I will use SMOTENC to upsample the minority class in the training set- for that I can't have nulls, so I temporarily imputed the nulls into values that are relatively naturally clustered (Median) and after applying the SMOTENC I used the missingness flag feature to recreate the nulls where they used to be in addition to the synthetic samples
- I will use the Catboost Classifier model - this model can handle NULLs natively, and actually use them to its own advantage thanks to its ordered boosting mechanisms. In addition this model is very known to prevent overfitting by also using symmetrical tree splits and ordered boosting (which prevents leakage). I can use a PRAUC (precision-recall curve) as its evaluation metric to try and manage the imbalance.
- I will look at the precision - recall - F1 score metrics to understand if the model works well, and with the prediction probability I can create a ranking mechanism to tell who is more likely to have hypertension conditions than others, with a budget restricting the amount of patients we can send onwards to the expensive test and check if the model is giving a high enough recall and precision to this top percentage group.

In the end in terms of results the model didn't do so well, it did manage to find 70% of the patients with hypertension (high recall), but its precision was low and created a 70% false alarm rate. (precision was 30%).

When looking at the prediction probabilities, I could see that for higher probabilities the model was more prone to correctly predict a patient who experienced the more severe types of hypertension conditions. And using an example for a budget limit (only 20% of patients) the model actually got good results and was able to predict 84% of the patients with any hypertension condition

My recommendations are to improve the model results by doing better feature selection, while checking correlations and redundancy and the SHAP values of the current iteration. I would try additional models (like isolation forest) and different loss function (focal loss) while doing a better parameter optimization. And finally I would try creating better features for the model, probably with better processing of the clinical sheet with a better trained transformer model, that can give more trustworthy and specific information about the text.

Thanks for reading, I am available for any questions.
Noam Fradkin