Treat First What Kills First: Predicting Disease Severity with DDXPLUS

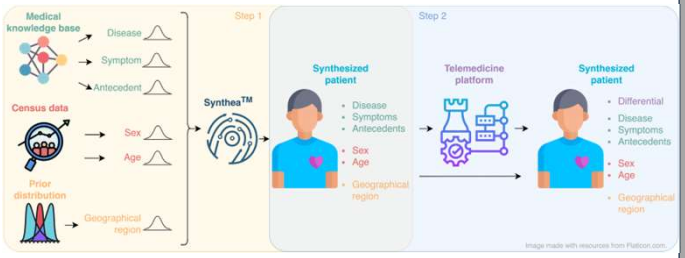
Leandra Hogrefe
hogrefe.16@osu.edu
Department of Physics, The Ohio State University

Motivation

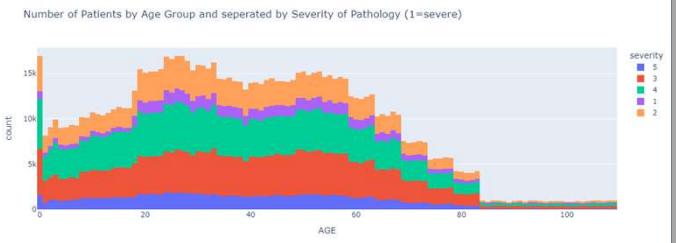
In the medical field, the accurate and timely prediction of diseases is essential to ensure patients have the best possible chances of getting healthy. At the same time, medical professionals are needed more than ever, and many medical institutions are short-staffed increasing the workload of each doctor and nurse. With machine learning on the rise, there have been many efforts to support medical professionals by producing differential diagnoses or predicting/ruling out severe diseases.
This project makes use of the DDXPLUS data set to explore exactly this opportunity. How well can a machine learning model predict a disease's severity? The data set comes with a wide variety of evidence assigned to each virtual patient. This evidence is then used in this project as the input for models to separate the patients into different severity classes.

Dataset

The patients in the DDXPlus dataset are artificially generated to ensure individual anonymity. The first graphic shows how the "patients" are generated.



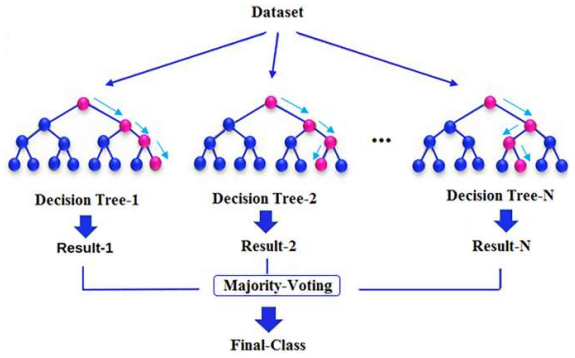
The dataset is imbalanced in pathologies as well as disease severity. This second graphic shows how they are distributed relative to each other and across age groups:



There are a total of 223 symptoms and antecedents (evidence) in the dataset. To use these as features, we one-hot encode the evidence and also normalize them for the CNN.

Methods

A Random Forest is a collection of many Decision Trees which separate the data by splitting it based on a criterion. Each of them predicts a class and the result of the Random Forest is determined by voting for a majority result as shown in the graphic below.



As the name suggests, CNNs take advantage of a process called "convolution". The CNN features several convolutional layers that learn the patterns present in training data to apply it to unseen data later.

Results

The Random Forest metrics were very similar for train and test data. I confirmed these with a k-fold validation with 5 folds. Each time the precision was about 92% and the fitting process took about 10min. A brief look into feature importance showed that that maximum importance of a single feature only got up to ~3% because of the large number of features in total. The important features were from a surprisingly wide spread of feature areas.
For the CNN, I experimented with the complexity of the architecture and ended up settling in the structure shown below (layer in the red box was removed) the fitting took much longer with about 1h and 47min but the validation accuracy got up all the way to 0.997 after 14 epochs. The model got stopped due to early stopping I implemented in the model to avoid unnecessary fitting process.

```
model_m = keras.models.Sequential()
#
# Our first layer gets the input from our samples
model_m.add(keras.layers.Conv1D(100, 10,
                    activation='relu', input_shape=(462,1)))
#
# Another convolutional layer
model_m.add(keras.layers.Conv1D(100, 10, activation='relu'))
#
# Max pooling
model_m.add(keras.layers.MaxPooling1D(3))
#
# Two more convolutional layers
model_m.add(keras.layers.Conv1D(80, 10, activation='relu'))
model_m.add(keras.layers.Conv1D(80, 10, activation='relu'))
#
# Global average pooling use this instead of "Flatten"
model_m.add(keras.layers.GlobalAveragePooling1D())
#
model_m.add(keras.layers.Dropout(0.5))
model_m.add(keras.layers.Dense(num_classes, activation='softmax'))
```

Discussion

	Pred:1	Pred:2	Pred:3	Pred:4	Pred:5
True:1	13113	568	2358	120	0
True:2	0	34723	2314	4627	0
True:3	0	249	59136	2593	0
True:4	0	0	1925	61661	10
True:5	0	0	3	3951	17770

The CNN took about 1h 47min to achieve a high accuracy (see confusion matrix below). The Random Forest got some very far off classifications as seen in the confusion matrix above but only took about 10min to train.
This raises the question whether speed or accuracy should be prioritized and the answer for the medical context would likely be both to give the patient the best possible care. This makes it difficult to evaluate which model is preferable.

	Pred:0	Pred:1	Pred:2	Pred:3	Pred:4
True:Most	16142	12	4	1	0
True:More	4	41637	3	20	0
True:Intermediate	0	21	61933	21	3
True:Less	0	28	7	63098	463
True:Least	0	0	3	18	21703

Conclusions and Future Work

The CNN outperformed the Random Forest in identifying disease severity classes. However, it required significantly more resources to do so. In a medical context this raises the question: How long would you be able to wait for a diagnosis to make sure it is as accurate as possible? A goal for a continuing project would be to experiment with models more that can achieve a high precision/recall quickly. To increase performance, there are also some aspects in the contexts of this medical situation that could be reflected better by the data/model structure. Implementing a higher penalization for “far off” classes (True: 1, Predict: 5) could boost performance. Additionally, the current one-hot encoding method causes the fitting to fail if testing data had more or less evidence columns due to the expected input shape. In a real-world application a new symptom might show up and the model still needs to perform. A more flexible model structure or developing a pre-modeling process to deal with these features would be an interesting project to tackle this problem.

References:
[1] Tchango, Arsene F. et al. DDXPlus Dataset \url{https://github.com/mila-iqia/ddxplus}, 2022.
[2] Classifying Sequences. Carmen Canvas: AU24 PHYSICS 5680 - Big Data Analytics. module8_sequences_2024.pdf, 2024.
[3] Random Forest. Carmen Canvas: AU24 PHYSICS 5680 - Big Data Analytics. module4_random_forests_only.pdf, 2024.