Event2vec, journal of experiments

March 22, 2023

Matthieu Doutreligne
matt.dout@gmail.com
HAS, Inria

Abstract

Contents

1 Experiments

1.1 Experimental setup overview

1.1.1 Tasks

Task 1 is LOS interpolation for every complete hospitalization. There is only 25,000 patients in the effective cohort, since 8,000 patients included do not have any events in the selected feature tables (icd10, procedures or, procedures).

Task 2 is next visit prediction (all hospitalizations: complete or incomplete).

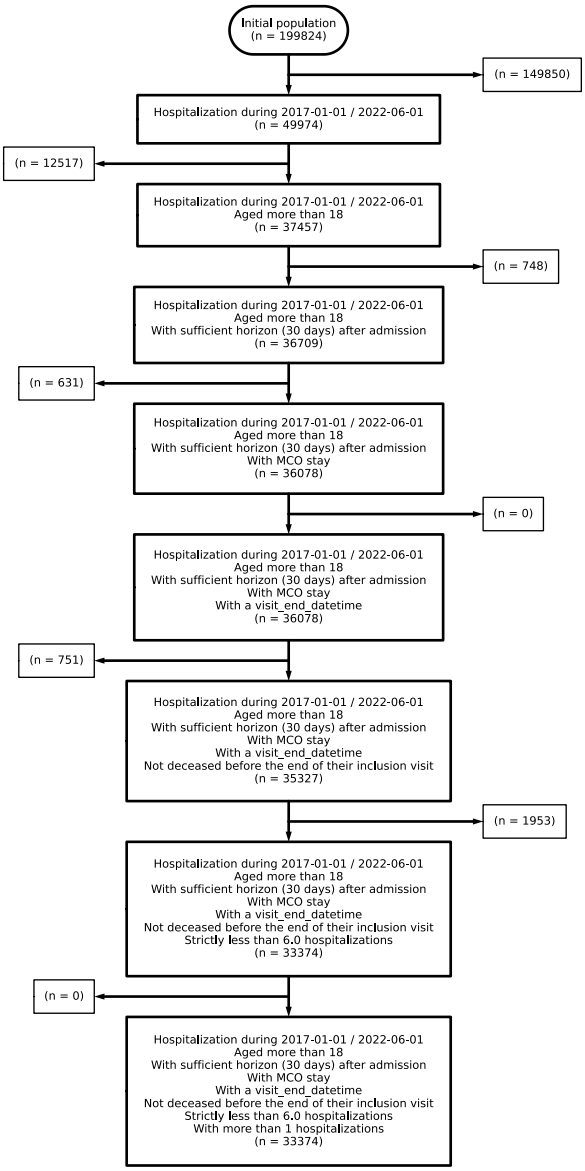


Figure 1: Inclusion criteria for T:LOS task.

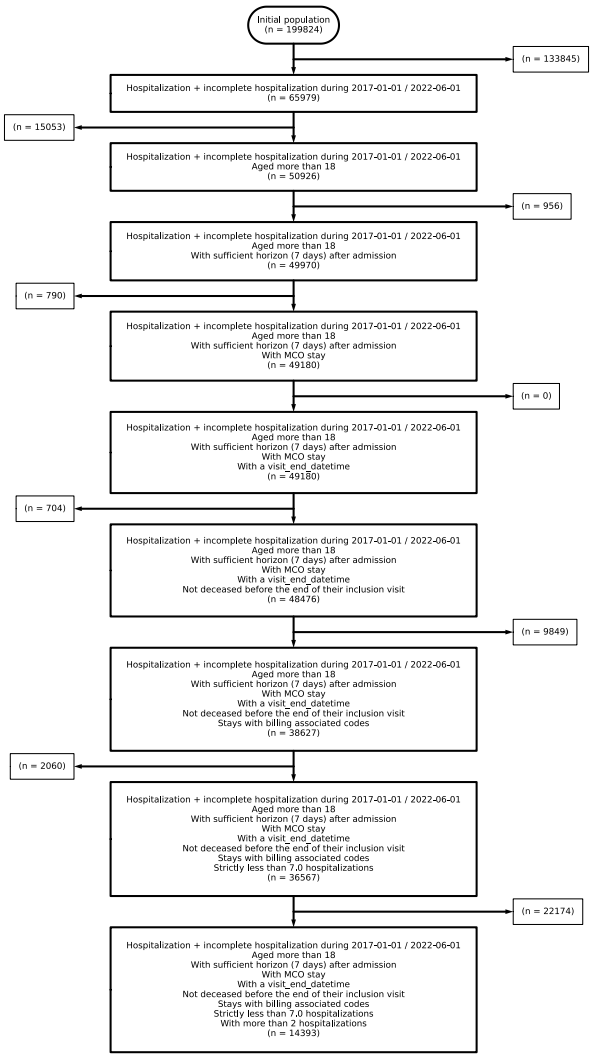


Figure 2: Inclusion criteria for T2:Prognosis task.

1.1.2 Evaluation setups

The first evaluation setup focus on the statistical efficiency gains of these embedding models. We fix a test set size (0.3 of total task population), then sample training sets of increasing sizes.

1.1.3 Features

By default, the features are the following:

- Billing codes (ICD-10)
- Procedure codes (CCAM)
- Administration drugs (ATC7)

We also add static features corresponding to the index stay of the task (T1=target stay, T2=first included stay): age, gender, admission reason, discharge destination, type and value. Finally, inclusion dates have been enriched to include the day of the week, and the month. The moment of the day is not included, since it is 22:00 or 23:00 for all visit starts.

A decay over the event is added to the features to include temporality: it is a decreasing exponential weight with half-life time depending of the task.

1.2 Task 1: LOS interpolation

1.2.1 Efficiency of the embedding models for T1:LOS

For this task, the exponential decay has been set to 7 days, focusing on events in the short term.

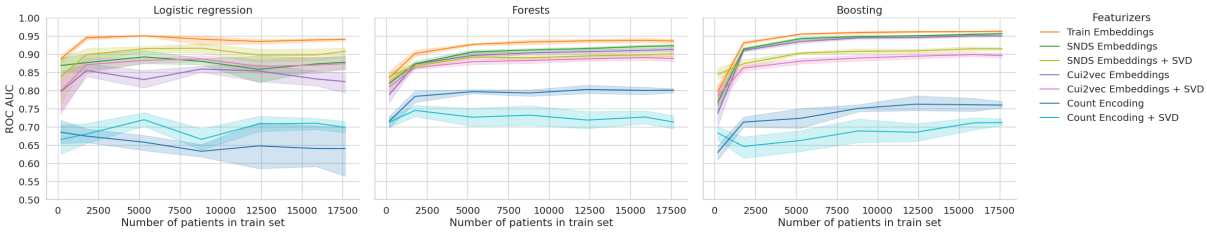


Figure 3: Performance of the embedding models for T1:LOS task. The study vocabulary is composed of 4734 codes, from which 4265 are found in SNDS embeddings and 2100 in cui2vec embeddings.

If I restrict all models to the vocabulary of 2100 codes that I successfully mapped from UMLS cui identifiers to the french vocabulary, I get the following results. THis gives a slight advantage to locally trained and cui2vec embeddings compared to SNDS where only 1987 study codes are found. For HGB, the performances are very similar between embedding featurizers (wo dimension reduction) for boosting. Forests have slightly worse performances and favor local embeddings.

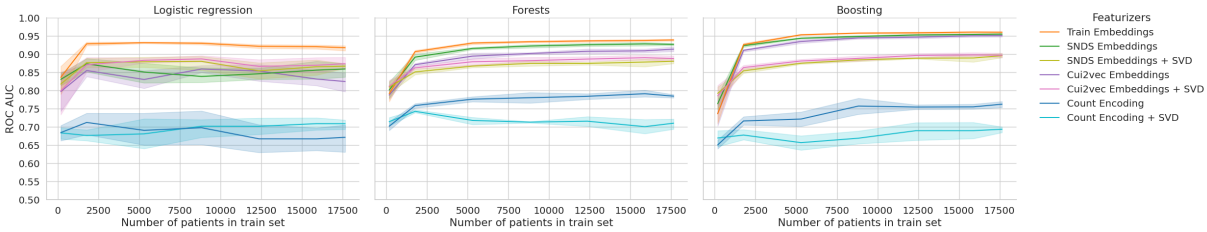


Figure 4: Performance of the embedding models for T1:LOS task. Vocabulary has been restricted to 2100 common codes between UMLS cui identifiers and the study vocabulary.

1.2.2 Transfer

1.2.2.1 Decay 7

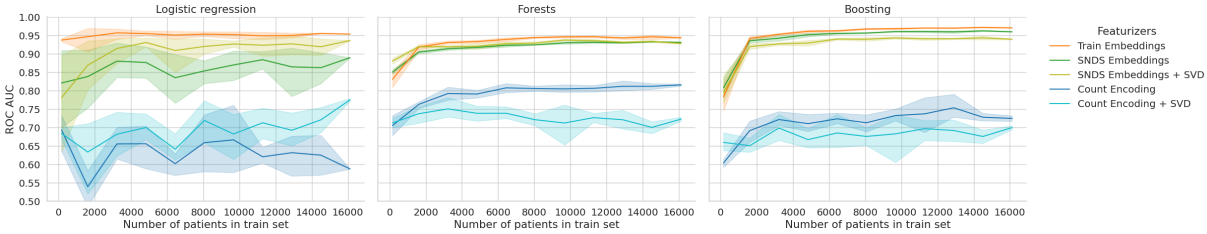


Figure 5: T1:LOS task, transfer between hospitals: The best model is the embeddings retrained from scratch on APHP, concordant with the efficiency setup. The SNDS embeddings with forest and boosting seems to close the gap compared to the efficiency setup.

Transfering with a restriction to the 2100 codes of cui2vec vocabulary.

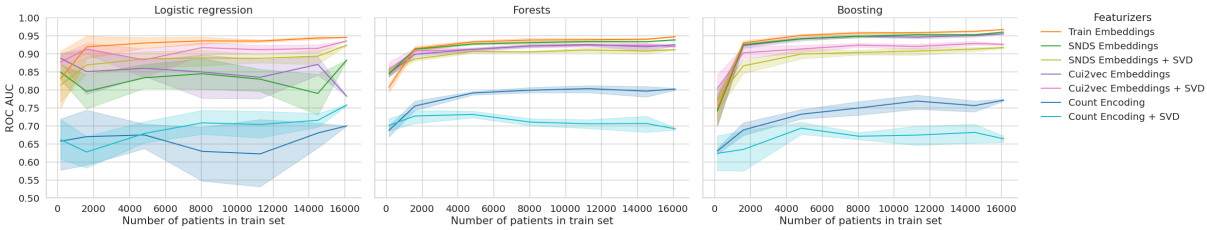


Figure 6: T1:LOS task, transfer between hospitals with 2100 codes only: The local, SNDS or cuiv2vec embeddings with forest and boosting are all equivalent in forests and boosting. For logistic regression, the local embeddings remain the bests.

Looking at the brier does not change the conclusion.

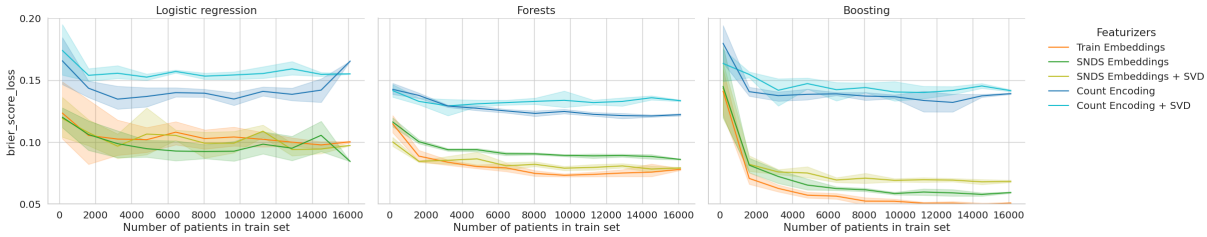
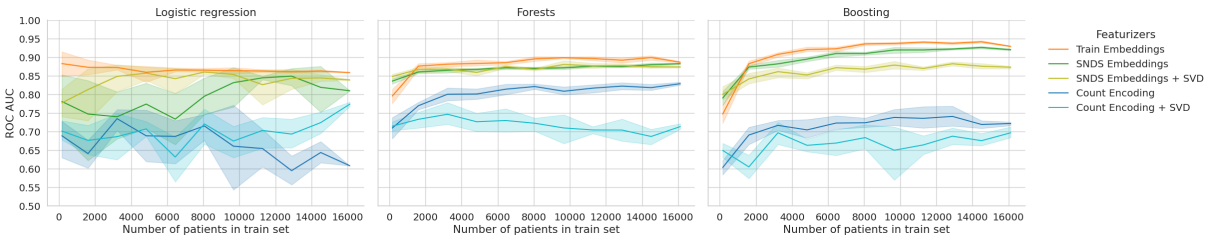


Figure 7: T1:LOS task, transfer between hospitals: There is no clear advantage in term of Brier score either for SNDS embeddings, beside a small advantage for embeddings+SVD. NB: These are even better results for brier score than the one obtained within the efficiency setup.

1.2.2.2 Decay is 30 (error on my part)

With the logistic regression, the SNDS embeddings transfer not very well. However, with forest or hgb, we have similar performances to the embeddings retrained from scratch on APHP. Forcing a distribution shift does not seems to hinder performances. For the Forests estimator, we got the best performances with a decay of 1 day. It changes the performances in the same direction but with different optimal value for each featurizer.



1.2.3 Ablation study

1.2.3.1 What effect of the decay parameter ?

We clearly see that the decay parameter should at least be crossvalidated on the task, since it has a huge impact on the performances.

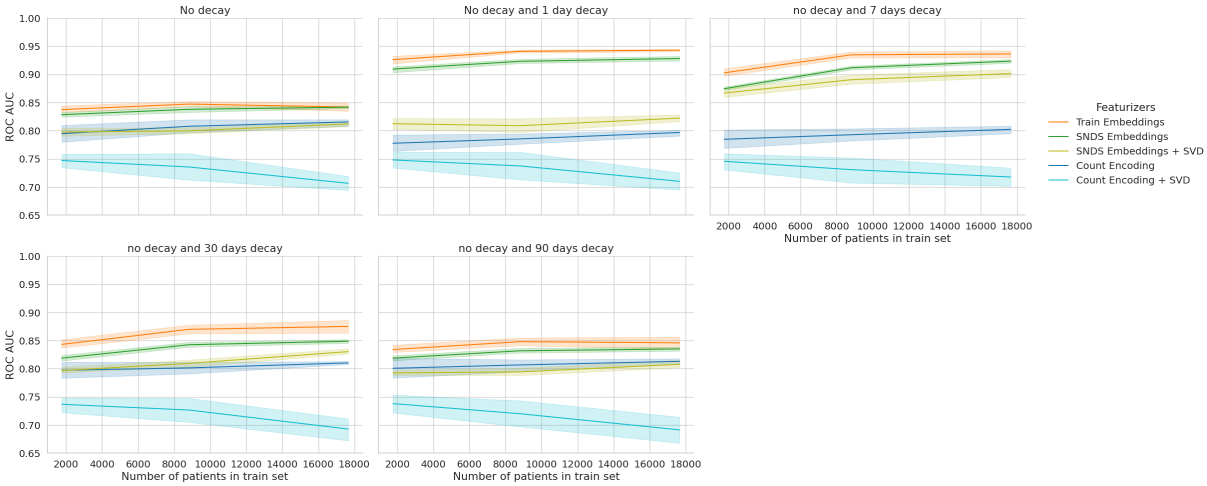


Figure 9: Ablation study on decay parameter for T1:LOS task.

1.2.3.2 What effect of the demographics ?

No effect for logistic regression (Figure 10) expect a slight effect for the Count encoding + SVD.featurizer. However, for random forests (Figure 11), the demographics improve the SNDS embedding method, bringing it to the same level as the embeddings retrained from scratch.

This suggests that simple demographics are easilly captured by concept embeddings, something already reported for attention based trajectory embeddings li2020behrt.

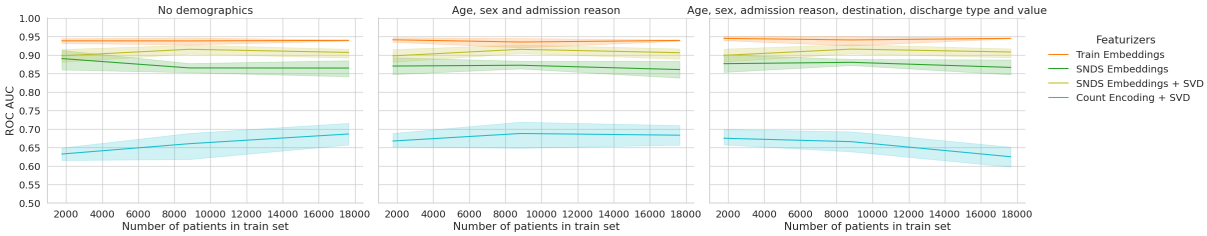


Figure 10: Ablation study on demographics with logistic regression for T1:LOS task.

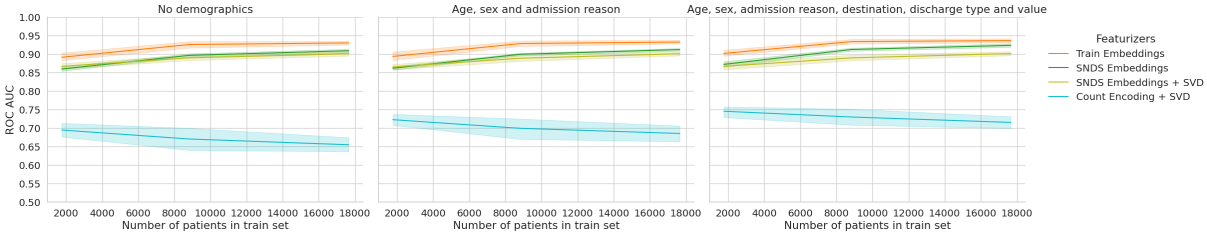


Figure 11: Ablation study on demographics with random forests for T1:LOS task.

Very little effect of the demographics for HGB for the embeddings featurizers.

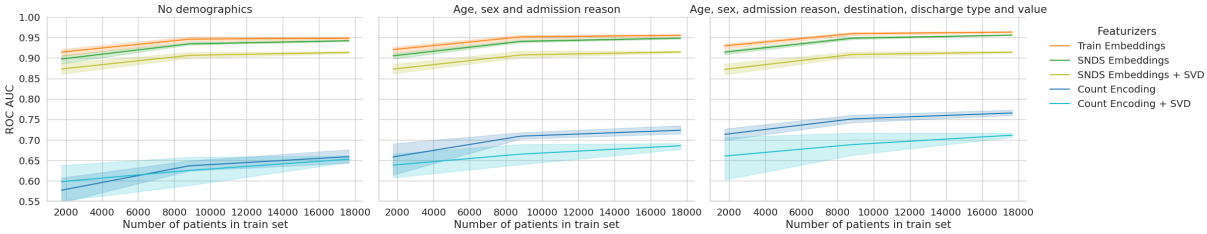


Figure 12: Ablation study on demographics with random forests for T1:LOS task.

1.3 Task 2: Next visit prognosis

1.3.1.1 Performances to expect from the Behrt paper

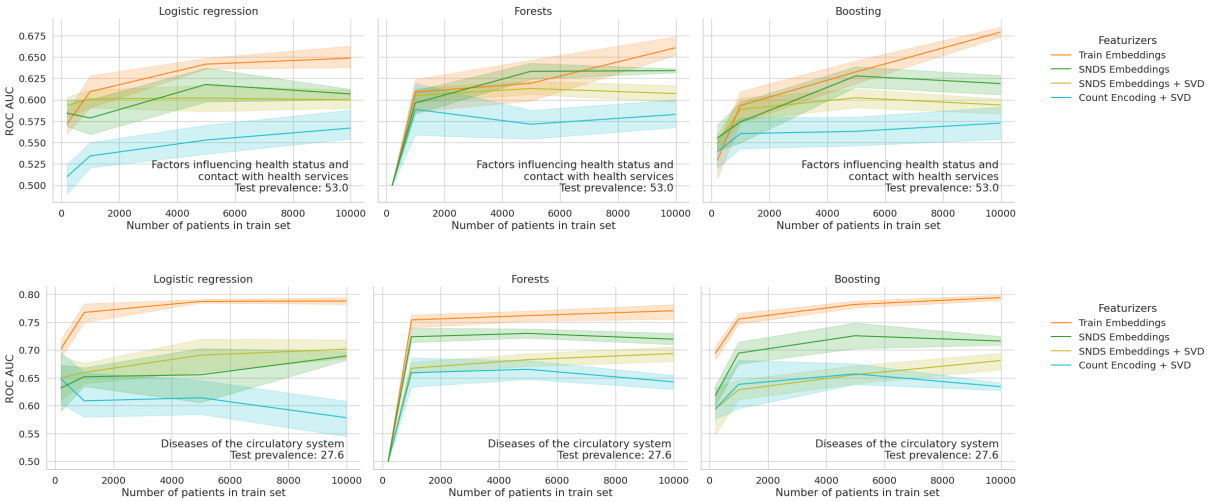
1.6m patients for Behrt pre-training (at least 5 visits and mapped to a icd10 or Read code). They reduce the vocabulary done to 301 codes for the whole analysis. For the next visit codes predictive task, they keep 700K patients and split them into 80/20 train/test, evaluated on ROC_AUC/APS (Average Precision Score). I am not entirely sure if they retrain from scracth or transfer the model. Looking at the next visit task notebook and the NextVisit model, it seems that it loads pretrained embeddings, so it might have a little bit of overfitting. However, this is not the case for both Deepr and Retain, that are trained from scratch. IMHO it is an unfair comparison in their paper with these other models.

For next visit, they focus on above 60 having =>1% prevalences and reach 0.954/0.462 in AUROC/APS, which is sensibly the same as Deepr (0.943/0.360) or RETAIN (0.921/0.382) for AUROC but much better in term of APS. Looking at the 6 months prediction (their second task), there is still important hetegoreneities in the performances depending of the Caliber chapter (nomenclature developped to merge UK codes). Overall, mental and behavioural disorders, then diseases of the circulatory system are well predicted.

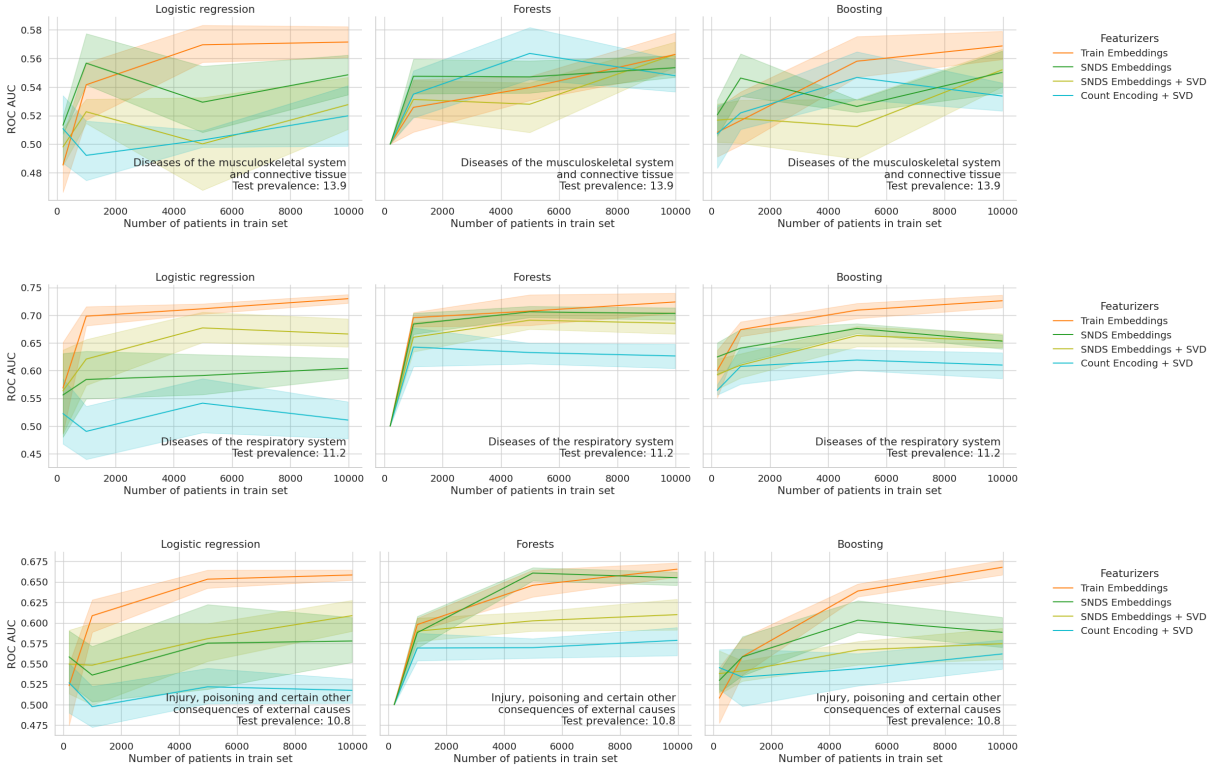
1.3.2 Efficiency of the embedding models for T2:Prognosis

Comparison of 7 vs 30 decay: 7 days decay is better for the circulatory (+5 ROC), endocrine (+2.5 ROC), mental (+3 ROC), respiratory (+3 ROC). If the effet of decay measured for RF on the LOS task are transferable, we could still gain some performances.

1.3.2.1 Logistic regression with 7 days decay :




Logistic regression
Forests
Boosting
ROC AUC
Number of patients in train set
Endocrine, nutritional and metabolic diseases
Test prevalence: 27.8
Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified
Test prevalence: 26.7
Neoplasms
Test prevalence: 16.5
Mental, Behavioral and Neurodevelopmental disorders
Test prevalence: 15.9
Diseases of the digestive system
Test prevalence: 15.3
Diseases of the nervous system
Test prevalence: 15.0
Diseases of the genitourinary system
Test prevalence: 14.1
Featurizers
Train Embeddings
SNDS Embeddings
SNDS Embeddings + SVD
Count Encoding + SVD

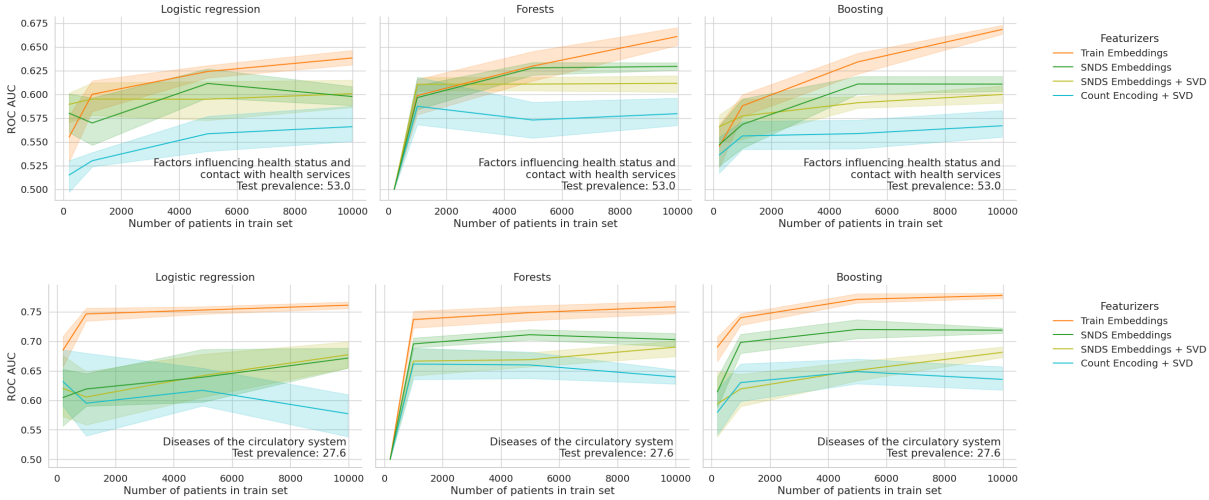


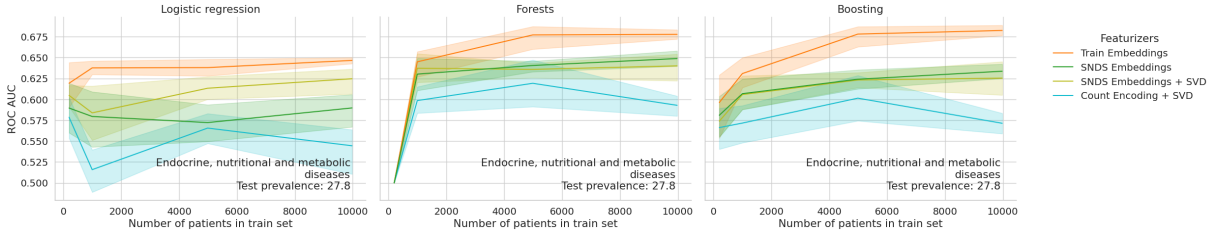
1.3.2.2 All models with 30 days decay

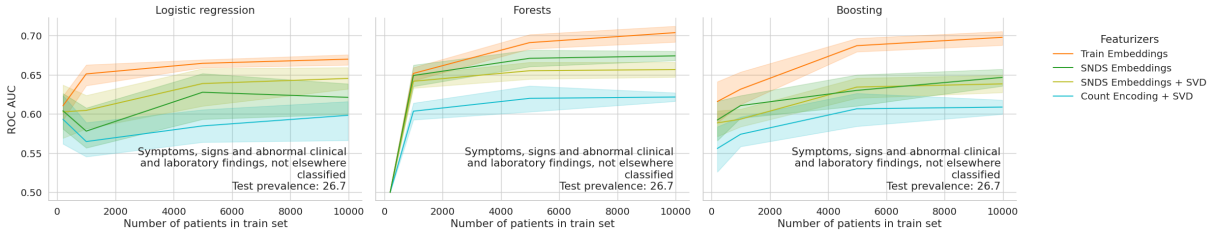
There are decent peformances of in-domain embeddings, for some icd10 chapters such as circulatory system (ROC_AUC=0.80), or endocrine, nutritional and metabolic diseases (ROC_AUC=0.70), Symptoms, signs and abnormal clinical and labo findings (ROC_AUC=0.70) or mental disorders (ROC_AUC=0.75), Disease of the respiratory system (ROC_AUC=0.75).

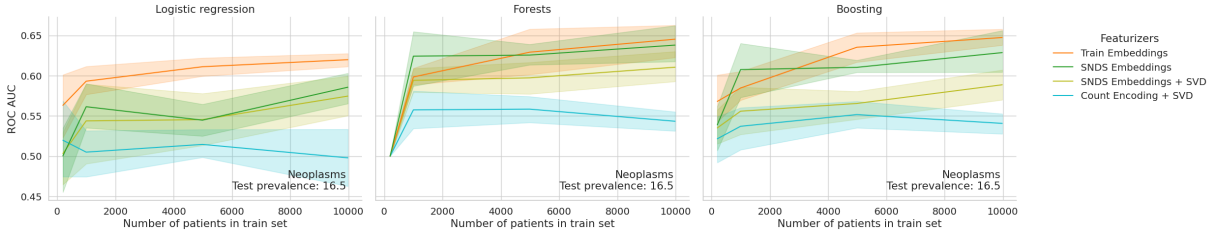
However these performances are never reached by the SNDS embeddings, that are seldom competitive with in-domain embeddings, and only in big samples setups (10,000 patients in train test). Sometimes, they strongly fail compared to in-domain training, as for the diseases of the circulatory system (ROC_AUC=0.70).

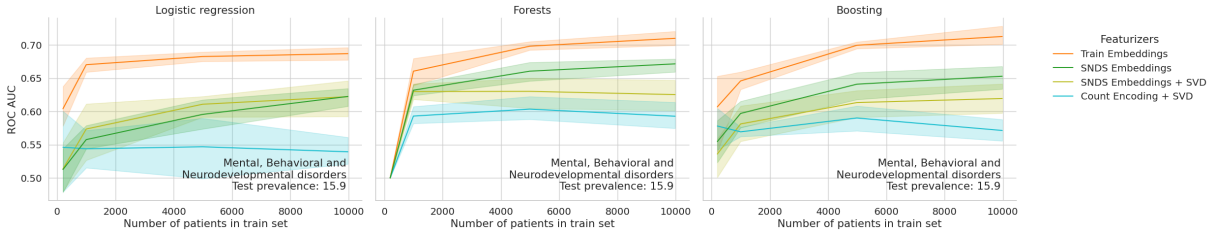
A general note: there is not signal ({ROC<=0.6}) in every task (eg. diseases of the musculoskeletal system.). Reassuringly, when there is signal, embeddings of the SNDS seems to largely outperform count encodings, but only at the price of big samples, compared to the in-domain embeddings.

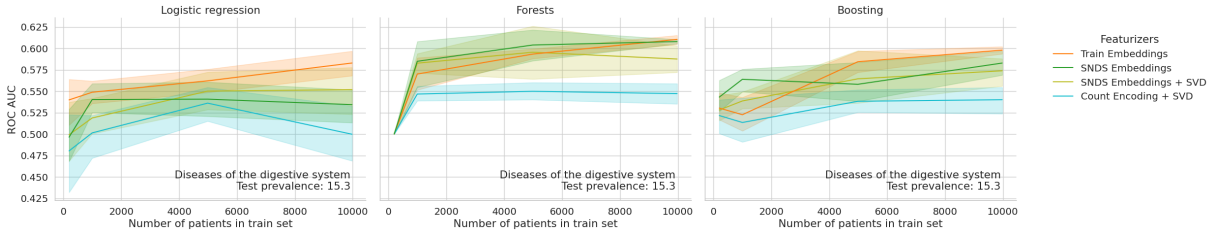


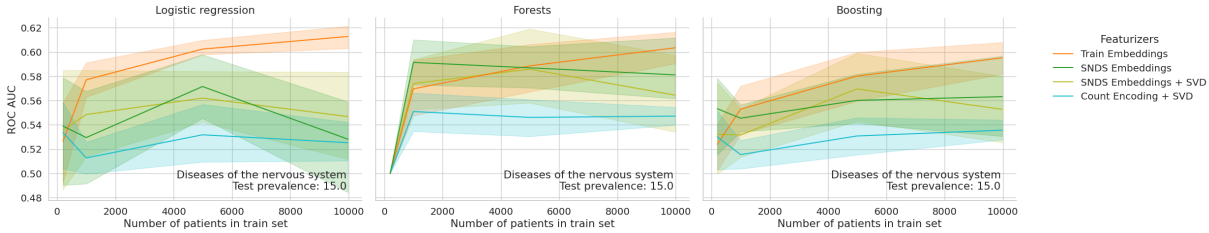

Logistic regression
Forests
Boosting
ROC AUC
Endocrine, nutritional and metabolic diseases
Test prevalence: 27.8
Number of patients in train set
Featurizers
Train Embeddings
SNDS Embeddings
SNDS Embeddings + SVD
Count Encoding + SVD

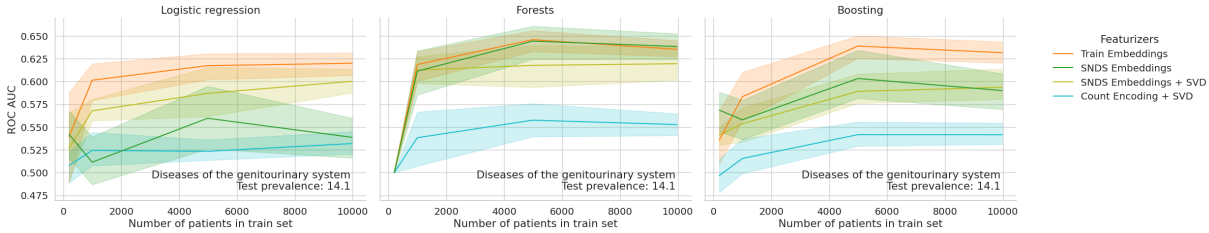

Logistic regression
Forests
Boosting
ROC AUC
Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified
Test prevalence: 26.7
Number of patients in train set
Featurizers
Train Embeddings
SNDS Embeddings
SNDS Embeddings + SVD
Count Encoding + SVD

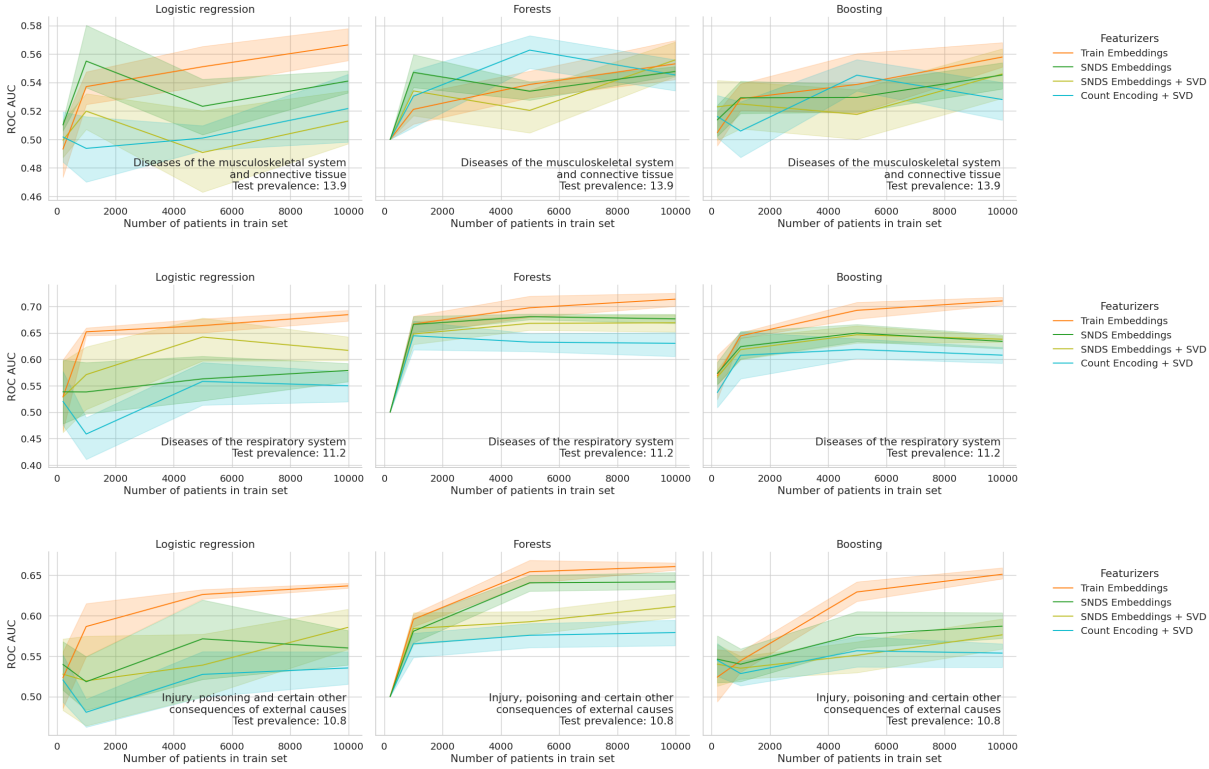

Logistic regression
Forests
Boosting
ROC AUC
Neoplasms
Test prevalence: 16.5
Number of patients in train set
Featurizers
Train Embeddings
SNDS Embeddings
SNDS Embeddings + SVD
Count Encoding + SVD


Logistic regression
Forests
Boosting
ROC AUC
Mental, Behavioral and Neurodevelopmental disorders
Test prevalence: 15.9
Number of patients in train set
Featurizers
Train Embeddings
SNDS Embeddings
SNDS Embeddings + SVD
Count Encoding + SVD


Logistic regression
Forests
Boosting
ROC AUC
Diseases of the digestive system
Test prevalence: 15.3
Number of patients in train set
Featurizers
Train Embeddings
SNDS Embeddings
SNDS Embeddings + SVD
Count Encoding + SVD


Logistic regression
Forests
Boosting
ROC AUC
Diseases of the nervous system
Test prevalence: 15.0
Number of patients in train set
Featurizers
Train Embeddings
SNDS Embeddings
SNDS Embeddings + SVD
Count Encoding + SVD


Logistic regression
Forests
Boosting
ROC AUC
Diseases of the genitourinary system
Test prevalence: 14.1
Number of patients in train set
Featurizers
Train Embeddings
SNDS Embeddings
SNDS Embeddings + SVD
Count Encoding + SVD



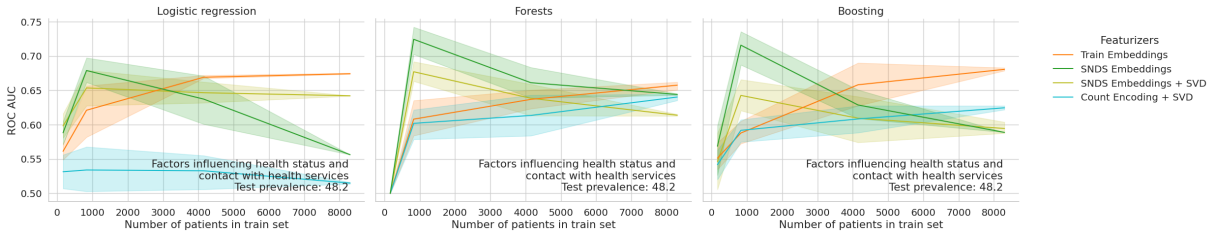
1.3.3 Transfer of the embeddings for T2:Prognosis

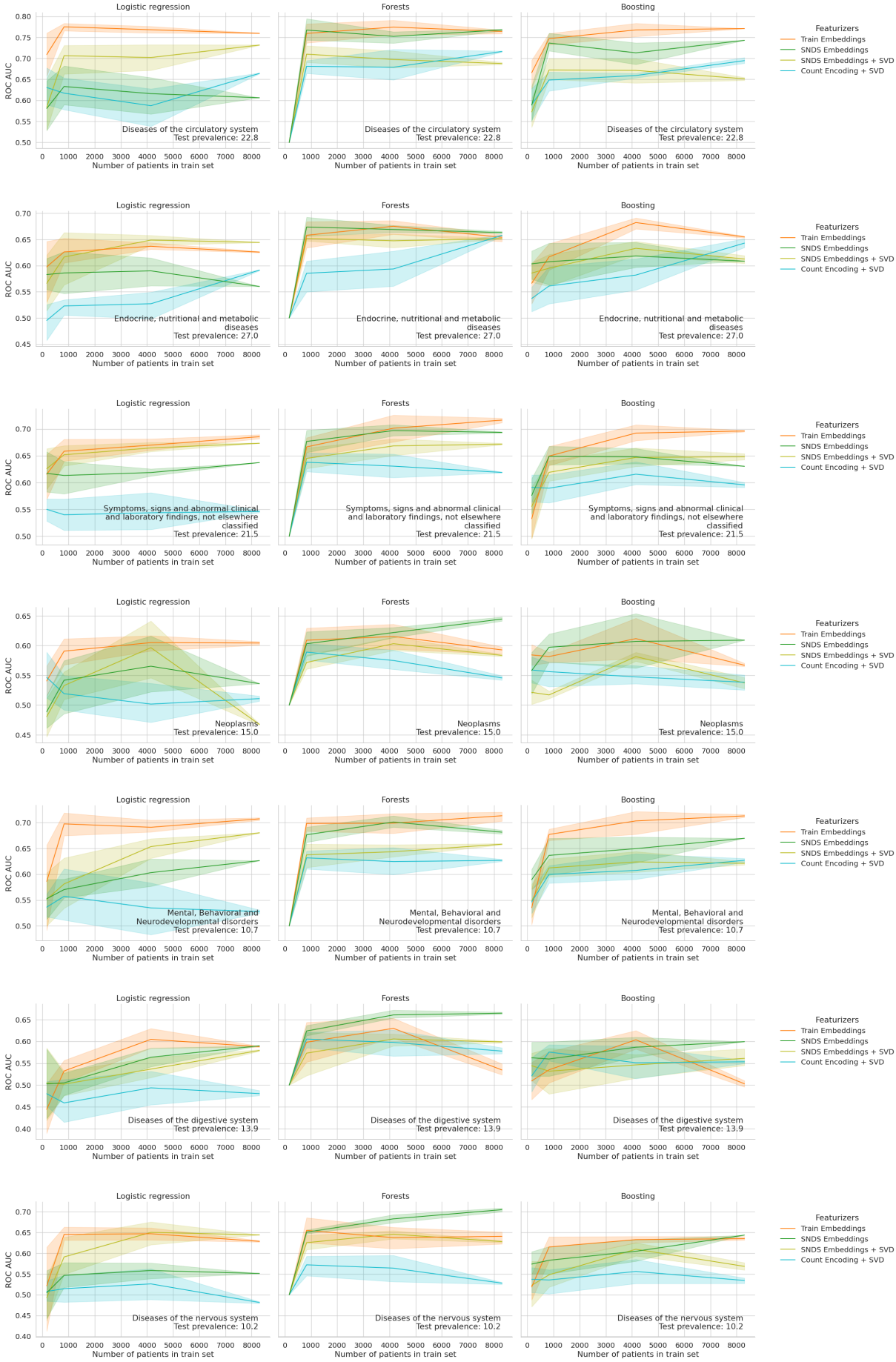
The SNDS embeddings transfer performances depends on the estimator.

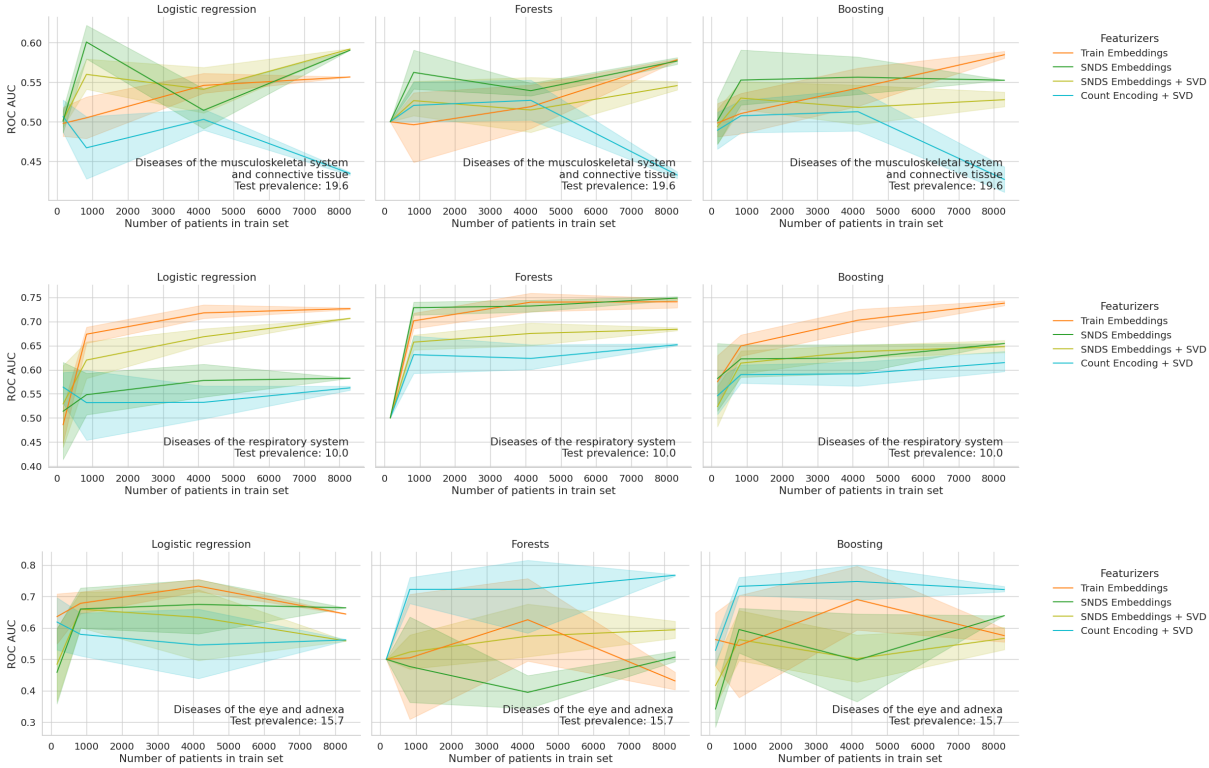
For forest, they are systematically competitive or better than local embeddings. Sometimes the improvement over locals is striking in small sample regimes (factor influencing health), other time in big sample regimes (neoplams + RF, disease of the nervous system + RF, digestive system + RF) suggesting real predictive power beyon mere extrapolation of linked codes, diseases of the nervous system. However, for logistic regression and boosting, they perform poorly compared to local embeddings.

For each ICD10 chapter, the best performances are almost the same between local embeddings + boosting or SNDS + forests for circulatory system, endocrine diseases, and respiratory system. Expect for neoplasms, digestive systems and nervous system where SNDS + forests is better. For mental and symptoms, the best performances are obtained with local + forests.

Very interstingly, the performances of the SNDS embeddings outperforms the performances the perforamnces in the efficiency setup in several cases: fator influencing health (+5 ROC), neoplasm (+2.5 ROC), digestive (+2.5 ROC), nervous system (+5 ROC), respiratory (+2.5 ROC). In every case, SNDS stands out significantly from the local embeddings (except for the respiratory system).



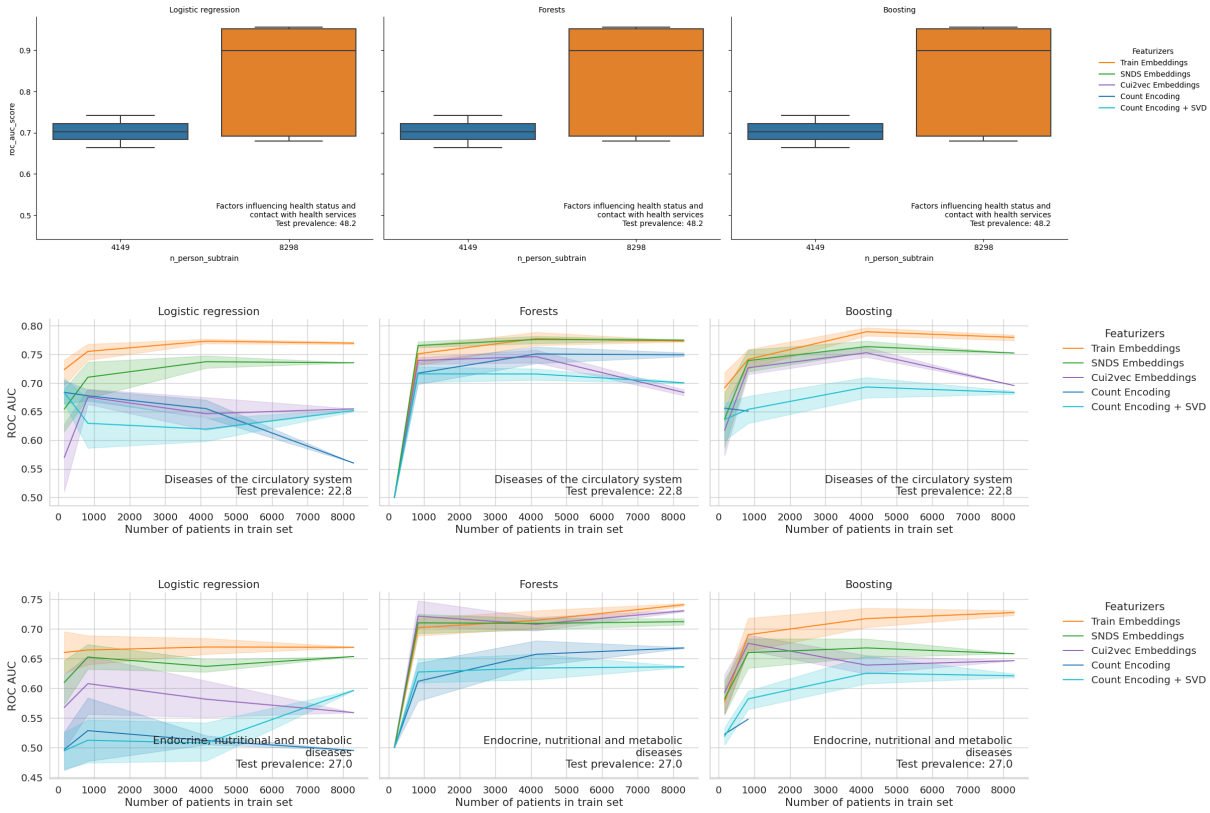

Logistic regression
Forests
Boosting
ROC AUC
Number of patients in train set
Diseases of the circulatory system
Test prevalence: 22.8
Endocrine, nutritional and metabolic diseases
Test prevalence: 27.0
Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified
Test prevalence: 21.5
Neoplasms
Test prevalence: 15.0
Mental, Behavioral and Neurodevelopmental disorders
Test prevalence: 10.7
Diseases of the digestive system
Test prevalence: 13.9
Diseases of the nervous system
Test prevalence: 10.2
Featurizers
Train Embeddings
SNDS Embeddings
SNDS Embeddings + SVD
Count Encoding + SVD

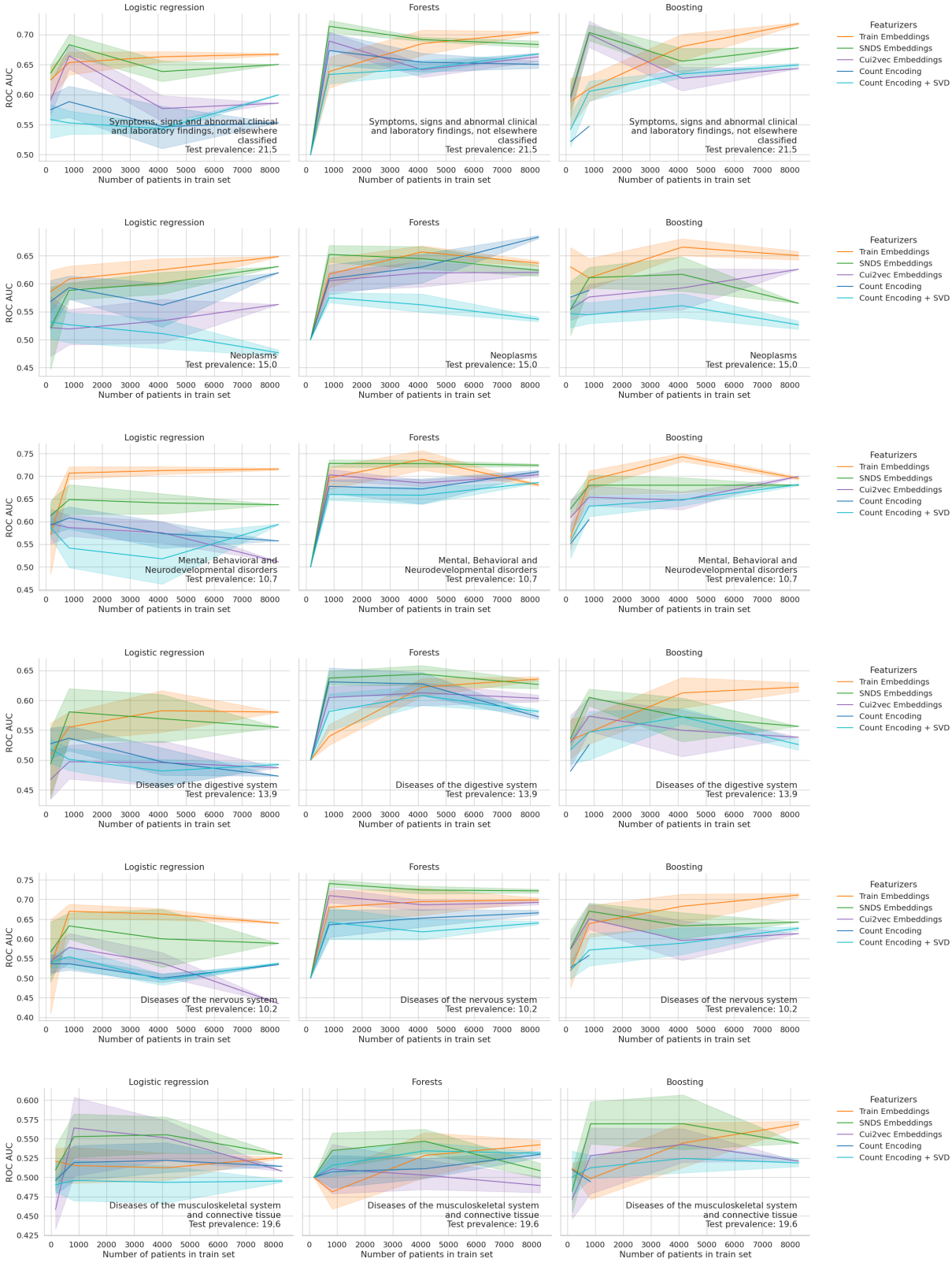


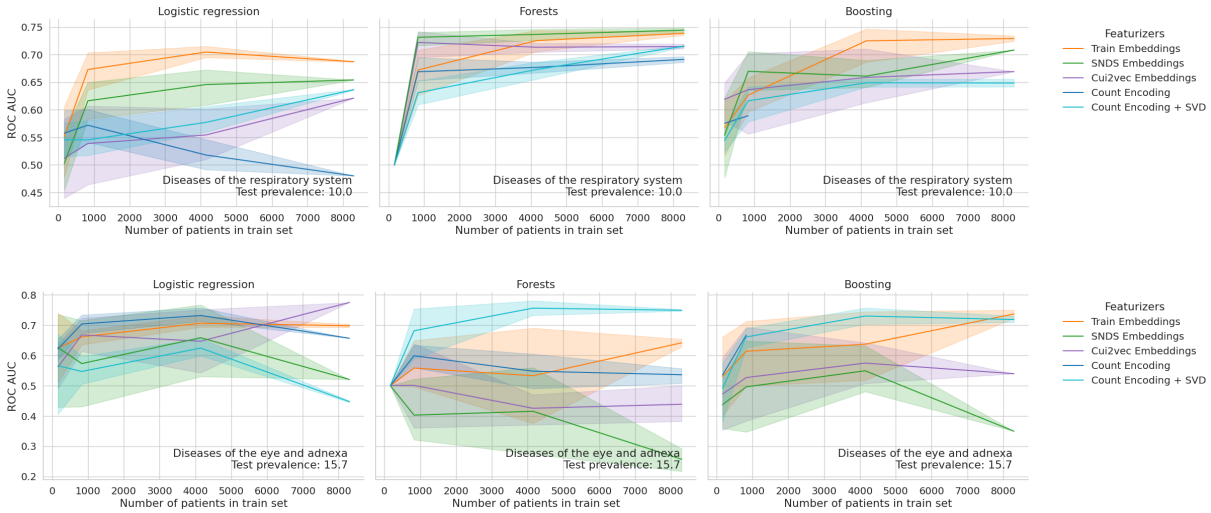
1.3.4 Transfer of the embeddings for T2:Prognosis, restricting to 2100 cui2vec codes.

NB: due to time constraint, the HGB with counts did not ran.

cui2vec is less interesting than the local or SNDS embeddings in this setup with better performances than the local embeddings, only for the strange case of disease of the eye, where counts seems to be sufficient to draw robust conclusions.




Logistic regression
Forests
Boosting
ROC AUC
Number of patients in train set
Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified
Test prevalence: 21.5
Neoplasms
Test prevalence: 15.0
Mental, Behavioral and Neurodevelopmental disorders
Test prevalence: 10.7
Diseases of the digestive system
Test prevalence: 13.9
Diseases of the nervous system
Test prevalence: 10.2
Diseases of the musculoskeletal system and connective tissue
Test prevalence: 19.6
Featurizers
Train Embeddings
SNDS Embeddings
Cui2vec Embeddings
Count Encoding
Count Encoding + SVD



1.3.5 Cehr-Bert attention model

1.3.5.1 Instantiating the model on the APHP sample

I forked the CEHR-BERT github of cehr-bert an attention-based model, that added visit type prediction, to the BEHRT model. My reasonning was that it is OMOP based so maybe more easy to use on APHP data, furthermore they included other domain than cim10 in the models (something not done by BEHRT).

I rely on train/test split by hospital id for the pretraining and the fine-tuning.

- I Initiated a eds readme.md for adaptation of cehr-bert to APHP-EDS data. Need a bit of cleaning for newcomers.

Key points:

- **pretraining:** it was quite easy to launch a pretraining on the APHP data (5h of work maximum to adapt the code). I only ran the pretraining on the training data of icd10chapter prognosis task ie 8297 persons, 622 000 events. I ran 2 epochs in 40 minutes on 10 CPUs (following their tutorial HPs). On one physical GPU (device: 0, name: Tesla T4, pci bus id: 0000:12:00.0, compute capability: 7.5) with 14GB, it tooks 6min.
- **finetuning:** Their code for cohort generation is hard to read and follow, so I added a code snippet in cehr-bert fork to create a binary task from an existing CohortEvent object which is basically two dataframes : person and event where person has a target column called y and events contain the eligible events for prediction (should be only after followup start if predictive task).

I finetuned on the same cohort of 8297 persons, 622 000 events reducing the multi-label tasks to next visit icd10 chapters only; 9-circulatory system and 2-neoplasms. The performances for the best concept embeddings models are respectively 77.5% and 65%. The performances reported for next 6 month prediction with Behrt are: 2) Neoplasms: APS=0.20/0.58, ROC_AUC=0.86/0.97 for primary malignancy other skin and primary malignancy prostates (prevalence around 1%); 9) Disease of the circulatory system: mean APS=0.37, mean AUROC=0.88, mean ratio=% over 6 caliber chapters.

I ran 10 epochs respictevely in 1h45 on 5 CPUs or in 15 minutes on one T4 GPU (following their tutorial HPs). Their final predictor is a BiLstm model on top of the bert pretrained model:

- **Evaluation**: I transfered the learned model on the test cohort of 1714 patients 79 946 events. I got excellent results for this task. I doubled check, but it seems that the model never saw the test example.

ICD10 chapter	recall	precision	f1-score	pr_auc	roc_auc
2: Neoplasm	0.583658	0.761421	0.660793	0.670023	0.867384
9: circulatory system	0.851282	0.935211	0.891275	0.947144	0.97181

Table 1: Fine tuning cehr-bert model works gives consistent results with the BEHRT paper.

1.3.5.2 Exhaustive evaluation with full training data for icd10 next chapter prediction

I struggled a little bit on putting the model on GPU. I had an error mentionning symbolic tensors. I solved it by forcing tensorflow==2.3.0, cudnn==7.6.5 and numpy==1.18.5. In the process, I retrained a preprocessing model from the beginning on GPU.

I ran this later model on the 21 chapters and got the following results.

target	prevalence	time_stamp	recall	precision	f1-score	pr_auc	roc_auc
21	0.5397	04-28-2023-20-39-44	0.8196	0.8908	0.8537	0.9408	0.9396
9	0.2946	04-28-2023-20-51-27	0.859	0.891	0.8747	0.935	0.9688
4	0.2812	04-28-2023-21-06-09	0.6818	0.8514	0.7572	0.8553	0.9243
18	0.2687	04-28-2023-21-20-40	0.7554	0.9175	0.8286	0.9054	0.946
2	0.1716	04-28-2023-21-35-21	0.0	0.0	0.0	0.2412	0.6412
5	0.1681	04-28-2023-21-47-00	0.4837	0.7063	0.5742	0.6666	0.9103
6	0.1628	04-28-2023-22-01-27	0.5371	0.7015	0.6084	0.6747	0.895
11	0.1603	04-28-2023-22-11-50	0.0	0.0	0.0	0.1765	0.6253
14	0.1561	04-28-2023-22-26-20	0.0	0.0	0.0	0.1446	0.598
10	0.117	04-28-2023-22-33-54	0.0	0.0	0.0	0.1452	0.6041
13	0.1162	04-28-2023-22-48-20	0.0	0.0	0.0	0.2126	0.5275
19	0.104	04-28-2023-22-58-43	0.0	0.0	0.0	0.1687	0.5845
3	0.0971	04-28-2023-23-11-44	0.0	0.0	0.0	0.1724	0.6489
1	0.0887	04-28-2023-23-26-17	0.0	0.0	0.0	0.1101	0.628
15	0.0552	04-28-2023-23-32-26	0.15	0.2	0.1714	0.1389	0.7765
12	0.0453	04-28-2023-23-42-43	0.0	0.0	0.0	0.0563	0.5077
20	0.0438	04-28-2023-23-54-27	0.0	0.0	0.0	0.0944	0.5982
7	0.0419	04-29-2023-00-09-00	0.0	0.0	0.0	0.4637	0.8122
22	0.0409	04-29-2023-00-23-27	0.0	0.0	0.0	0.0587	0.6313
17	0.0162	04-29-2023-00-37-54	0.0	0.0	0.0	0.0106	0.5404
8	0.0133	04-29-2023-00-52-20	0.0	0.0	0.0	0.0141	0.6133

There is a strange result on chapter 2, less performant with gpu-pretrain (AUROC=[0.62,0.64], and recall=precision=0) on two distinct finetuning runs compared to the results obtained with the cpu-train (AUROC=[0.86, 0.85], recall=precision=0.6). I am not sure what is the reason for this difference, besides different network initialization. I tried a long run for chapter 2 (20 epochs) using the gpu-pretrain to see if I recover the results from the cpu-pretrain. I got an AUC of 0.93. However, after having fix the seeds, I am not able to recover this result.

I ran 5 different seeds with epoch=10 on target 2 to see how much variability I get. Pushing to 20 epochs gives almost the same results (mean ROC_AUC=0.6163).

n	prevalence, mean (SD)	precision, mean (SD)	f1-score, mean (SD)	pr_auc, mean (SD)	roc_auc, mean (SD)
5	0.1716 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)	0.2164 (0.0110)	0.6135 (0.0182)

Table 2: Forcing different seeds inside the Bert evaluator gives consistent bad results for the Neoplasm chapter.

When doing the same with a different pretrained model, I got better results. Recall that the only change was the pretrain model seed (and the fact that this model has been trained on cpu, but that should not impact performances).

n	prevalence, mean (SD)	precision, mean (SD)	f1-score, mean (SD)	pr_auc, mean (SD)	roc_auc, mean (SD)
5	0.1716 (0.0000)	0.1624 (0.2223)	0.0333 (0.0591)	0.2708 (0.0203)	0.6725 (0.0194)

Table 3: Using a different seed for the pretrain model, I got much better results for the Neoplasm chapter.

1.3.5.3 Evaluating the full CEHR-BERT pipeline (pretraining, finetuning)

1.3.5.3.1 Comparaison with cui2vec favorables codes (information leakage)

I compared only to best performing methods with cui2vec favorables codes (restriction to 2100 codes).

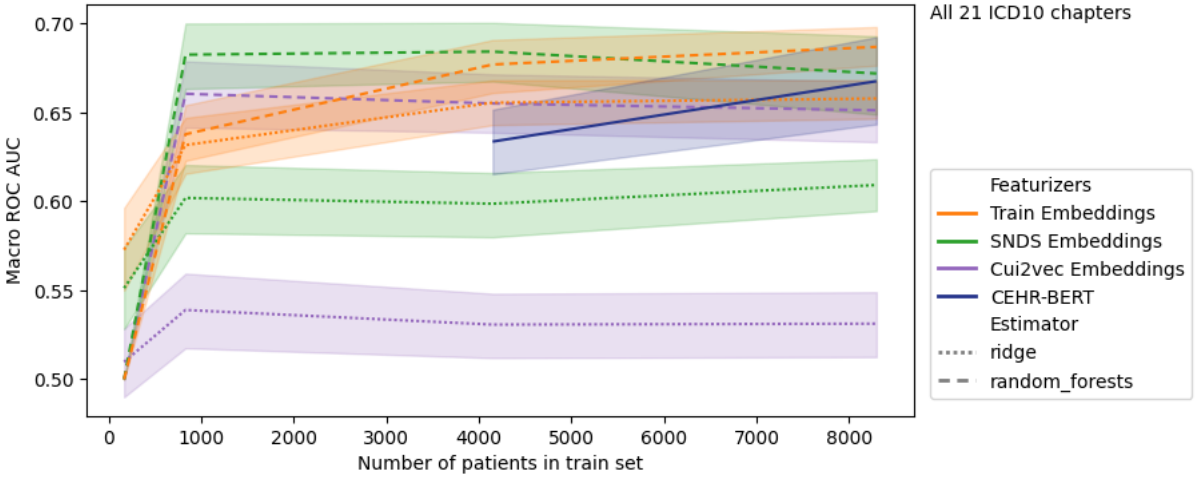


Figure 59: The CEHR-BERT pipeline gives better results than the best performing methods (restriction to 2100 codes favorables to cui2vec).

I then moved to all codes (4734 ICD10, procedure and drugs). Before the validation split by hospital ID for cehr_bert:

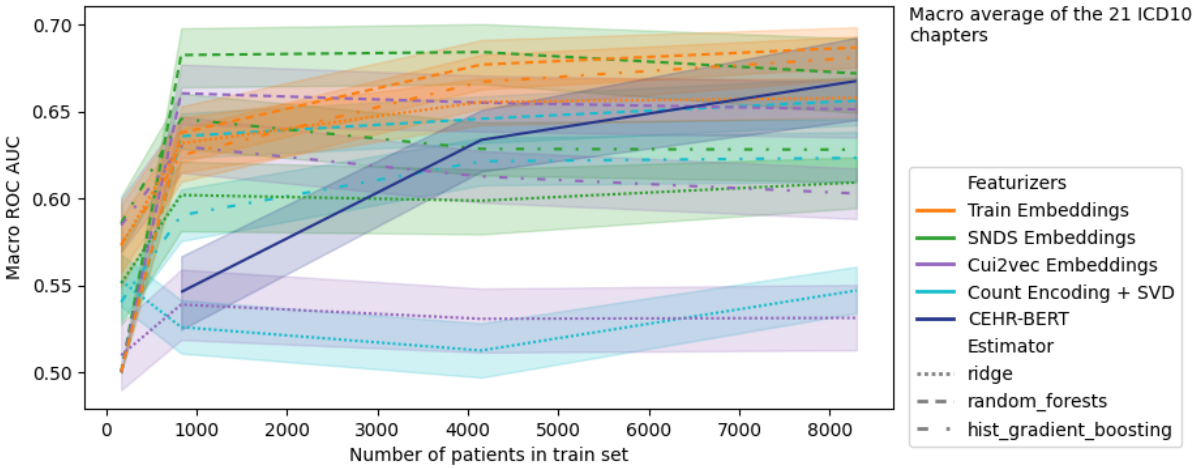


Figure 60: Macro AUC before validation split by hospital for CEHR-BERT.

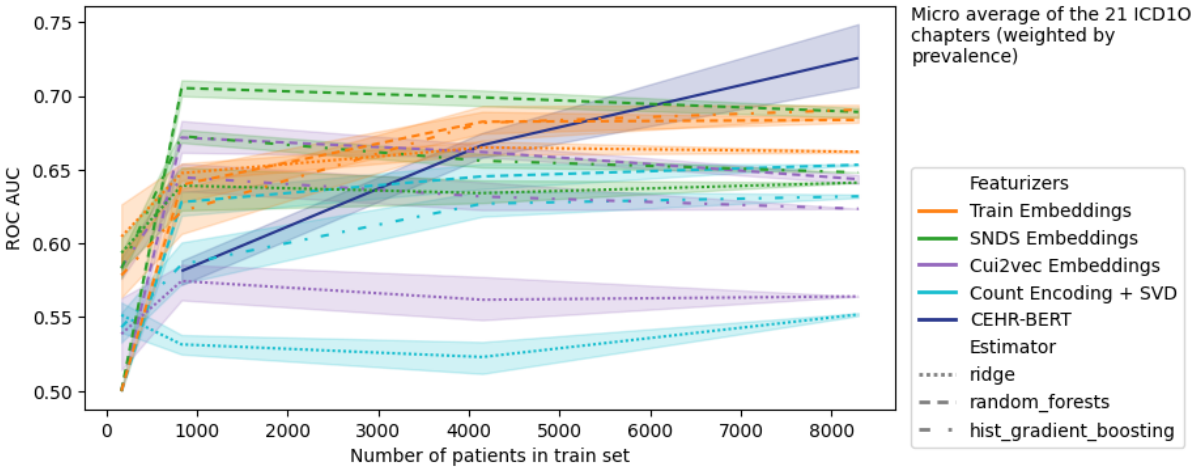


Figure 61: Micro AUC before validation split by hospital for CEHR-BERT.

A mini-sensitivity analysis of the importance of the decay parameter shows that it matters quite a lote with a 3 to 6% gain in ROC-AUC at the micro level. The largest differences are for logistic regression and in-domain train embeddings (comparison of red to orange dotted curves).

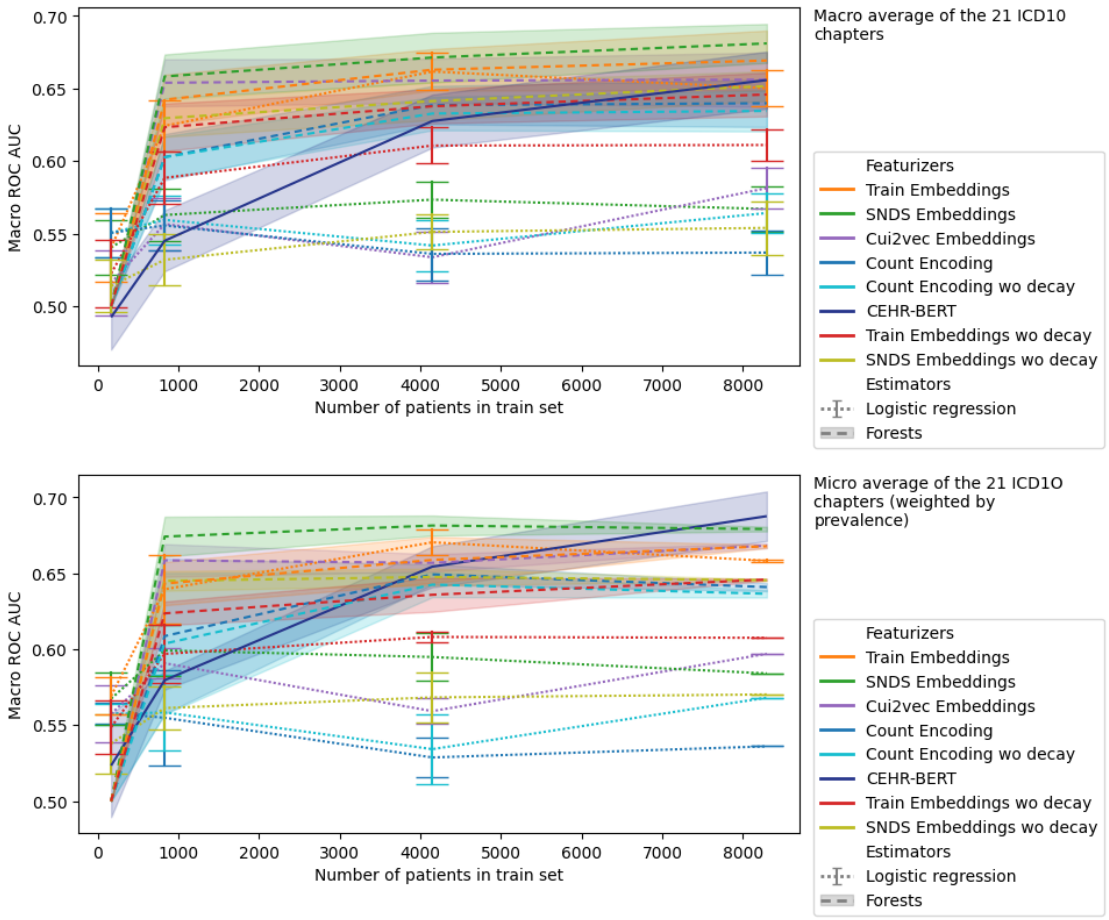


Figure 62: Adding a simple decay to the event improves performances at the micro and macro level from 3 to 6 points in ROC_AUC.

- with box plots:

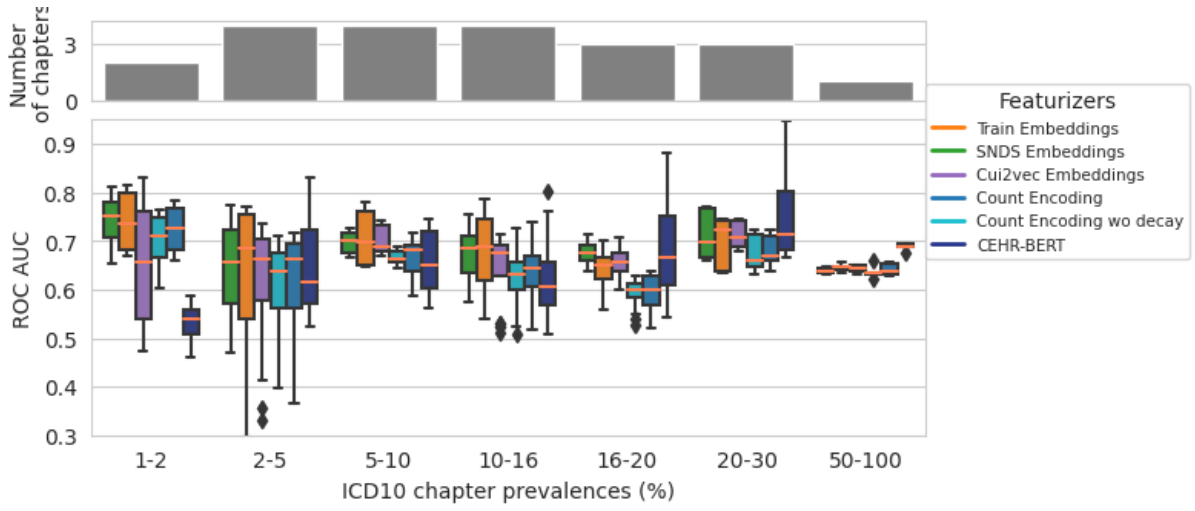


Figure 63: Except for CEHR-BERT, the performances of all featurizers chained with random forest estimators are independant from the chapter prevalence. Below 15% of prevalences, random forets manage to extract information from pretrained concept embedding methods and outperform CEHR-BERT.

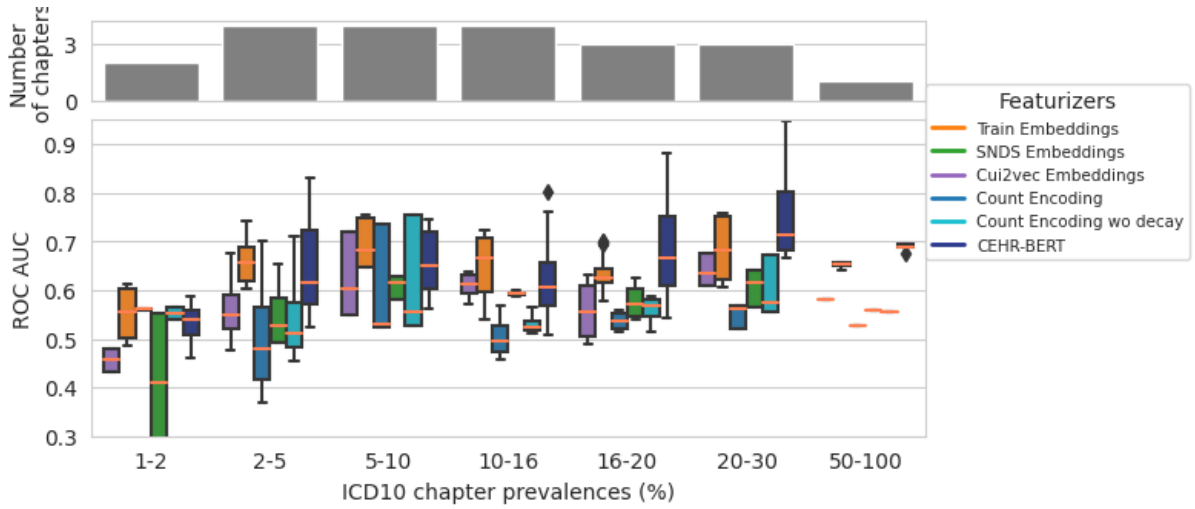


Figure 64: The performances of all featurizers chained with logistic regression benefit from higher chapter prevalences. The benefit are higher for CEHR-BERT.

Training time is also important to consider:

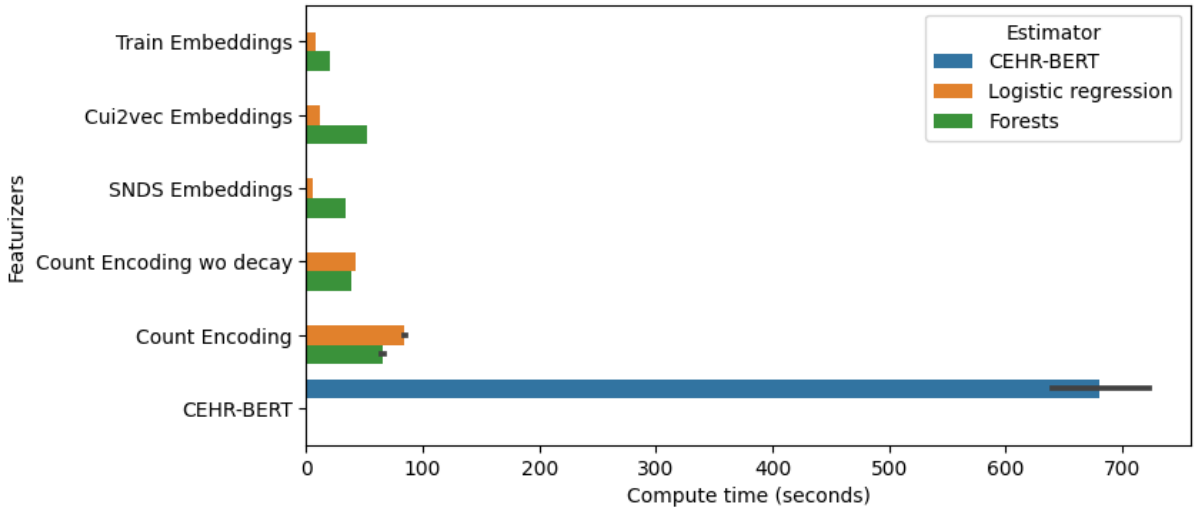


Figure 65: The training time (pretraining and finetuning) of CEHR-BERT is much higher than the other featurizers.

1.3.5.3.2 Final experience with all codes and hospital transfer (without information leakage)

The micro and macro average results as well as results on the 21 chapters are as follow.

The Naive Baseline (previous stay) is stronger than all embedding models for almost all chapters. It is beaten only for “infectious and parastic diseases” (not by a large margin), external causes of morbidity, and pregnancy (where count are doing great, which probably indicate that some procedures markers are strongly predictive of pregnancy related codes), and diseases of the ear and mastoid process (only good performances of embeddings).

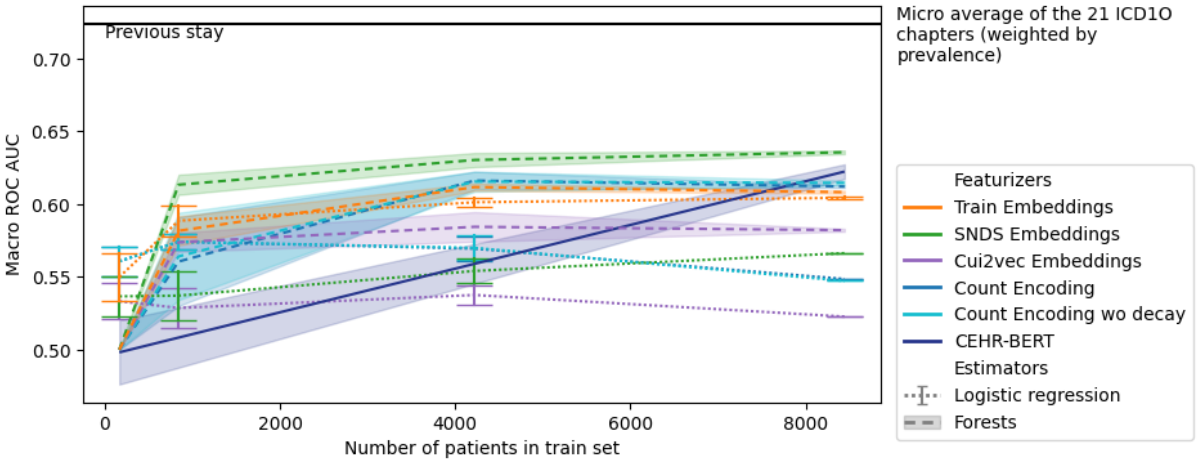
SNDS Embeddings are doing better than every other methods on macro and micro average. Beam embeddings performed the worst. Train embeddings are not better than count featurizer + random forests. I did not retry boosting.

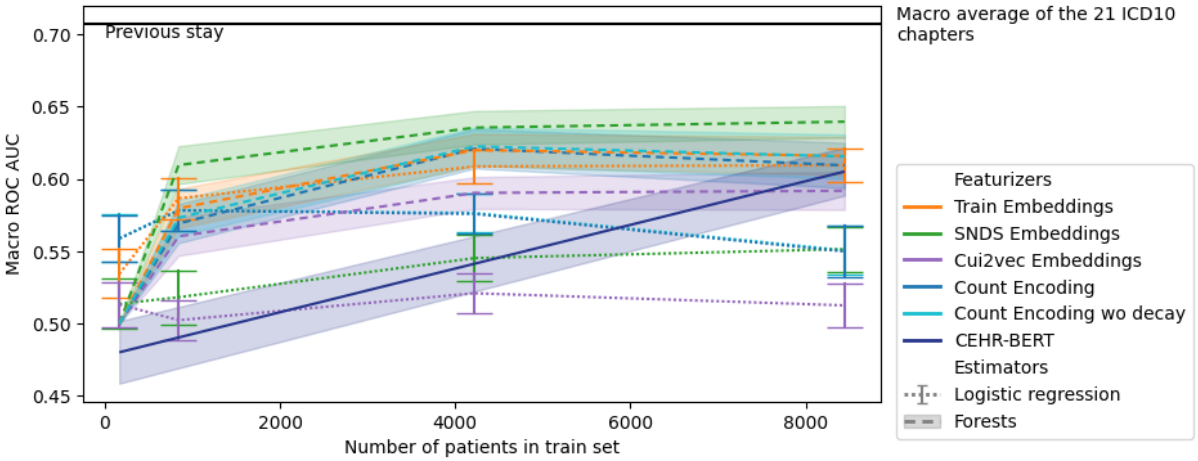
Perspectives: These results are a bit deceiving and underline the need to change task:

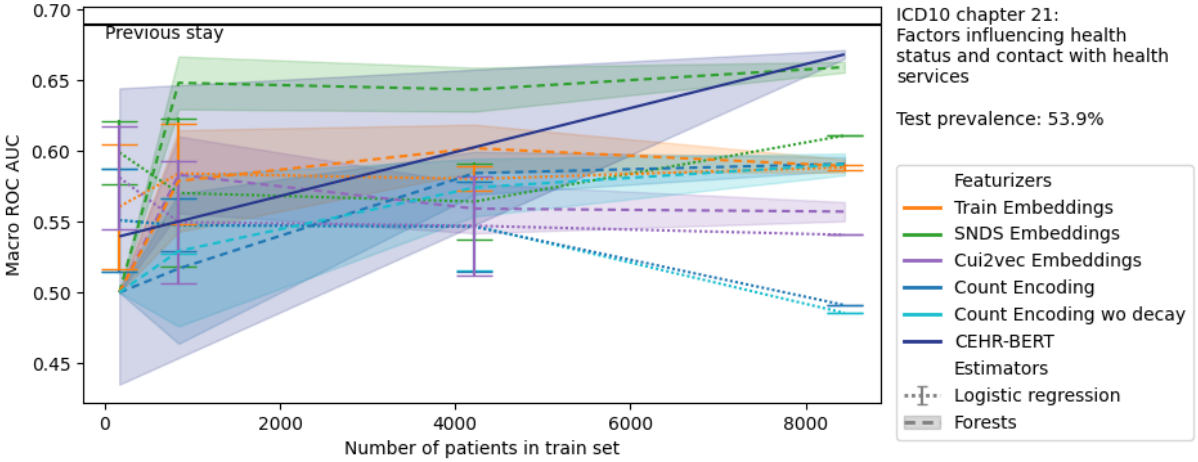
- Retry rehospitalization ?

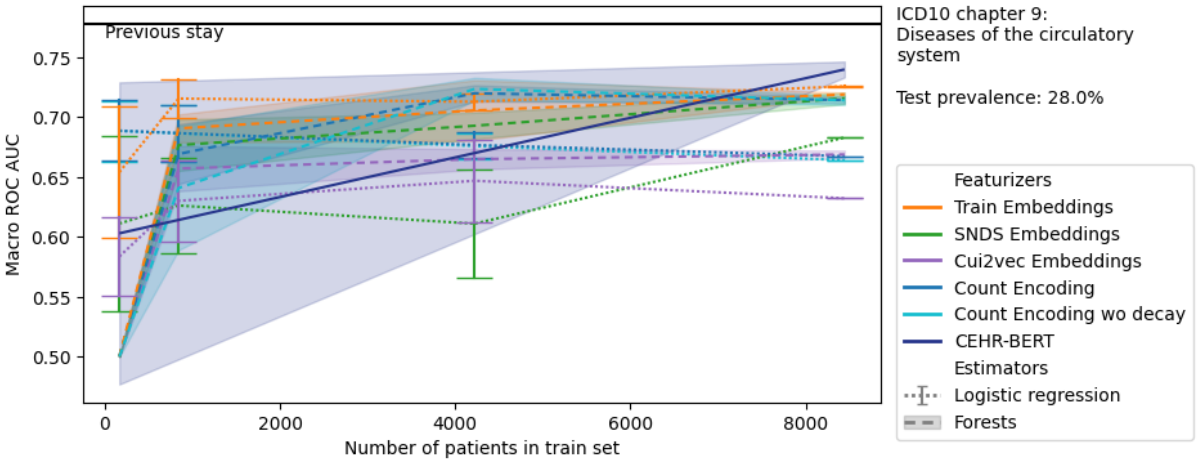
- Wait for progress from Julie on rehospit
- Try to add more codes (eg. prescribed drugs, until now, there are only administrated drugs.)

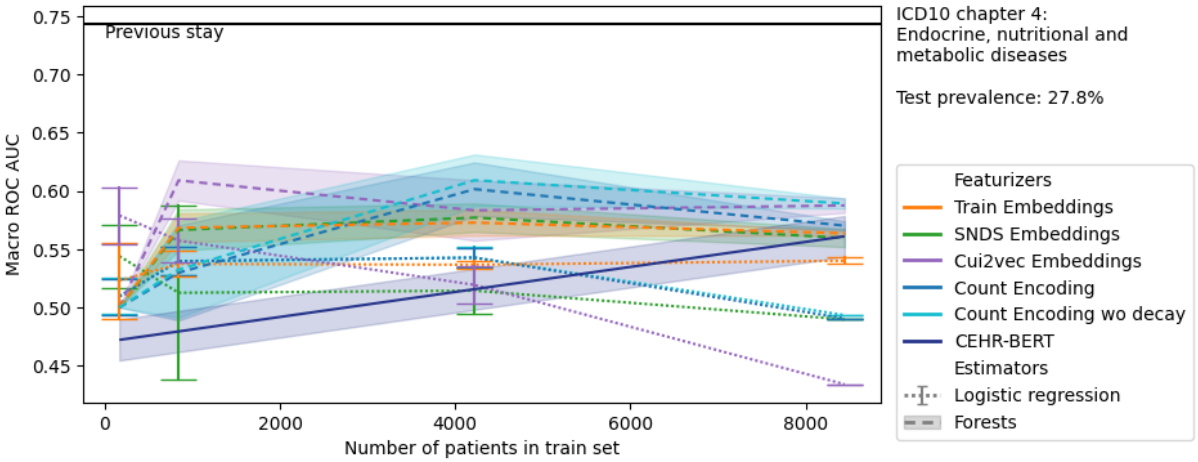
Results:

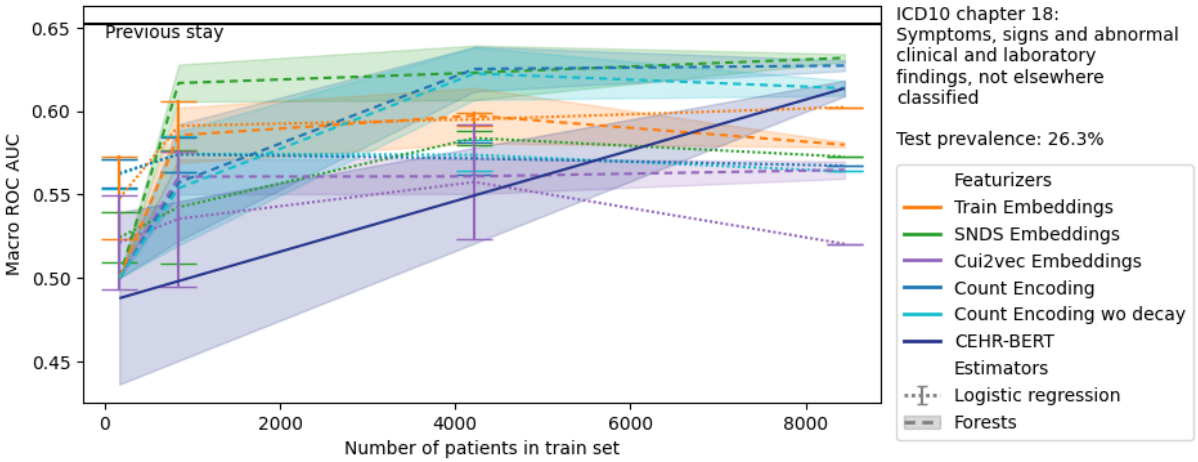


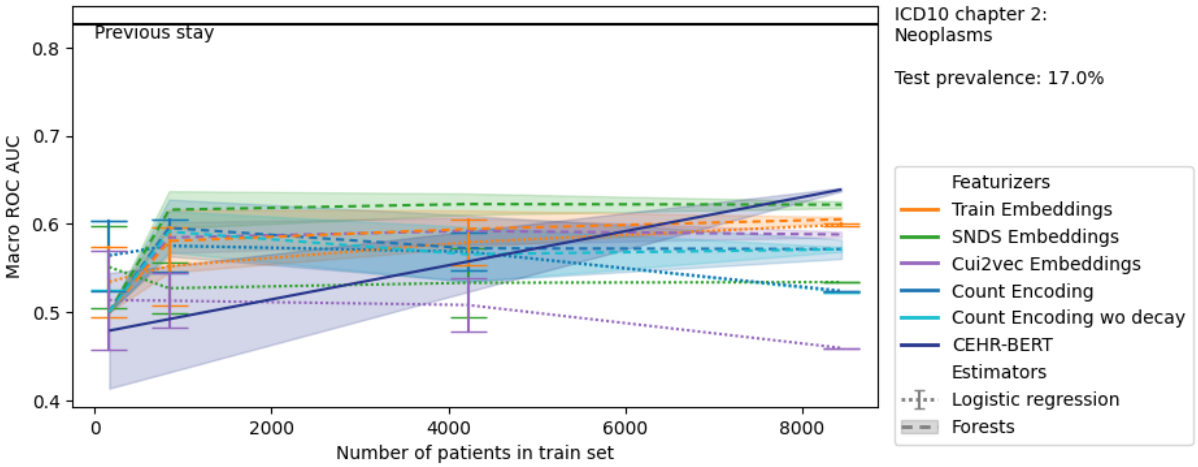


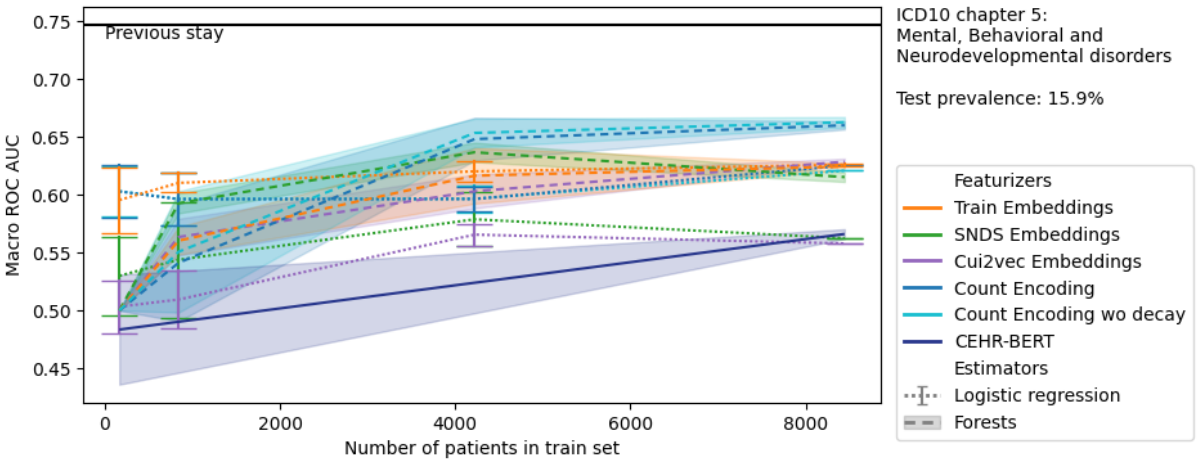


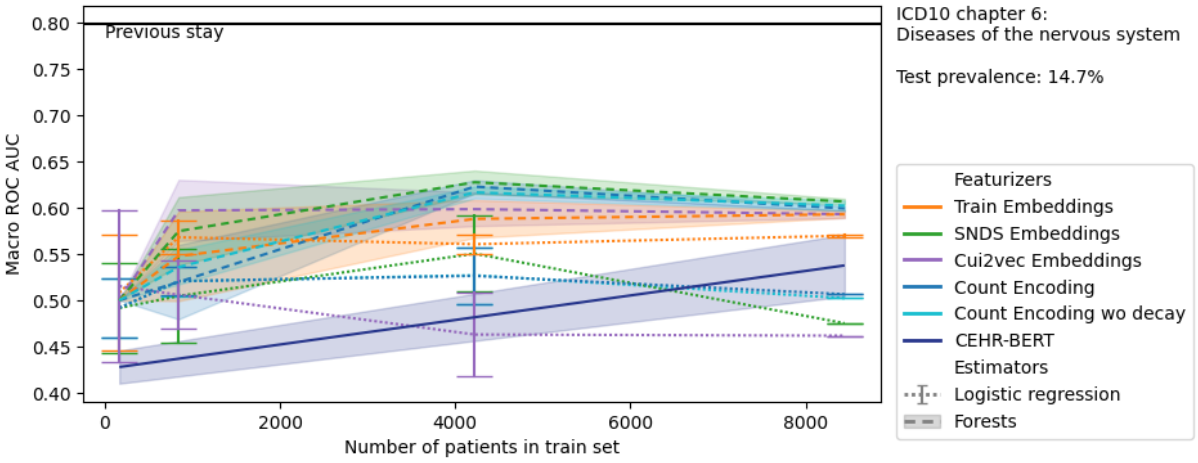

Previous stay
Macro ROC AUC
0.75
0.70
0.65
0.60
0.55
0.50
0
2000
4000
6000
8000
Number of patients in train set
ICD10 chapter 9:
Diseases of the circulatory system
Test prevalence: 28.0%
Featurizers
Train Embeddings
SNDS Embeddings
Cui2vec Embeddings
Count Encoding
Count Encoding wo decay
CEHR-BERT
Estimators
Logistic regression
Forests

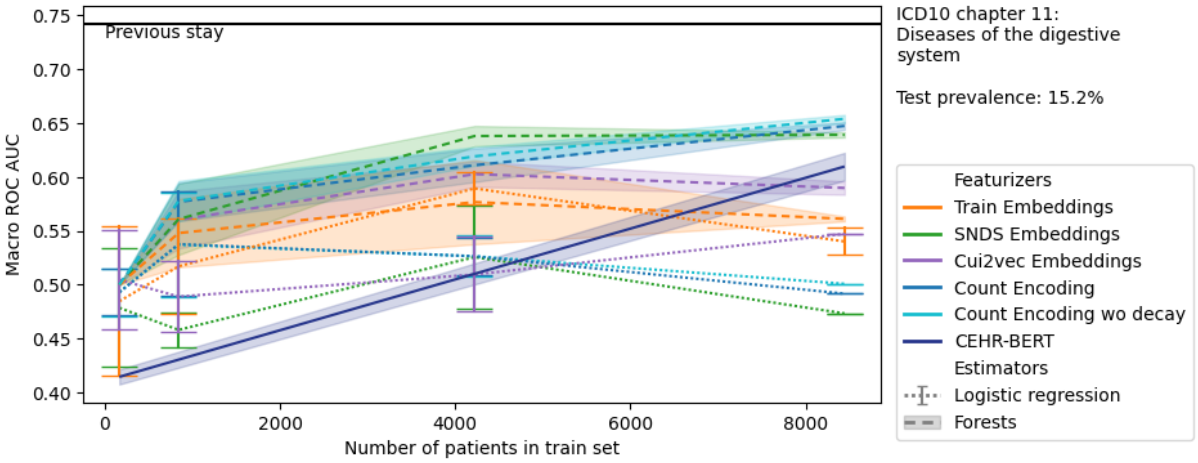

Previous stay
Macro ROC AUC
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0
2000
4000
6000
8000
Number of patients in train set
ICD10 chapter 4:
Endocrine, nutritional and metabolic diseases
Test prevalence: 27.8%
Featurizers
Train Embeddings
SNDS Embeddings
Cui2vec Embeddings
Count Encoding
Count Encoding wo decay
CEHR-BERT
Estimators
Logistic regression
Forests

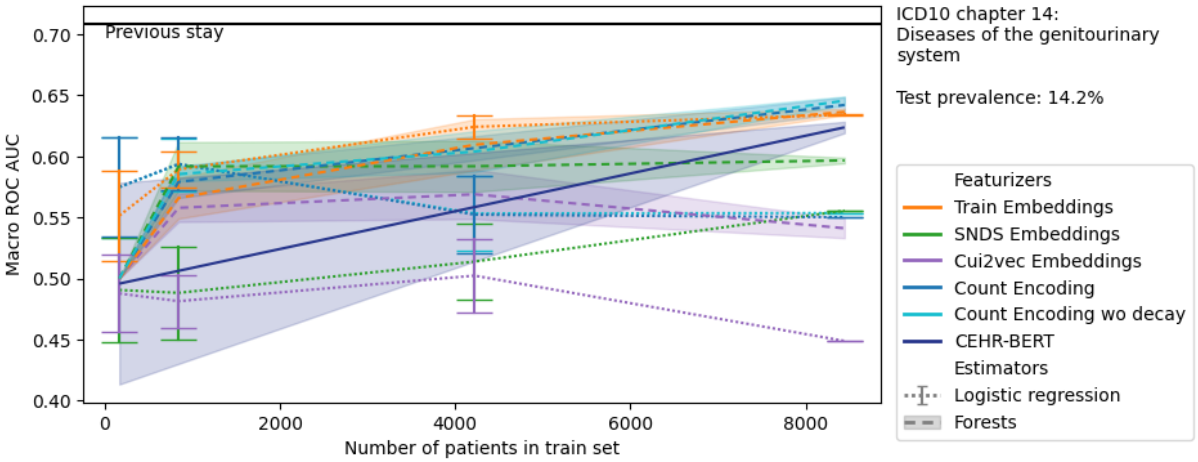

Previous stay
Macro ROC AUC
0.65
0.60
0.55
0.50
0.45
0
2000
4000
6000
8000
Number of patients in train set
ICD10 chapter 18:
Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified
Test prevalence: 26.3%
Featurizers
Train Embeddings
SNDS Embeddings
Cui2vec Embeddings
Count Encoding
Count Encoding wo decay
CEHR-BERT
Estimators
Logistic regression
Forests

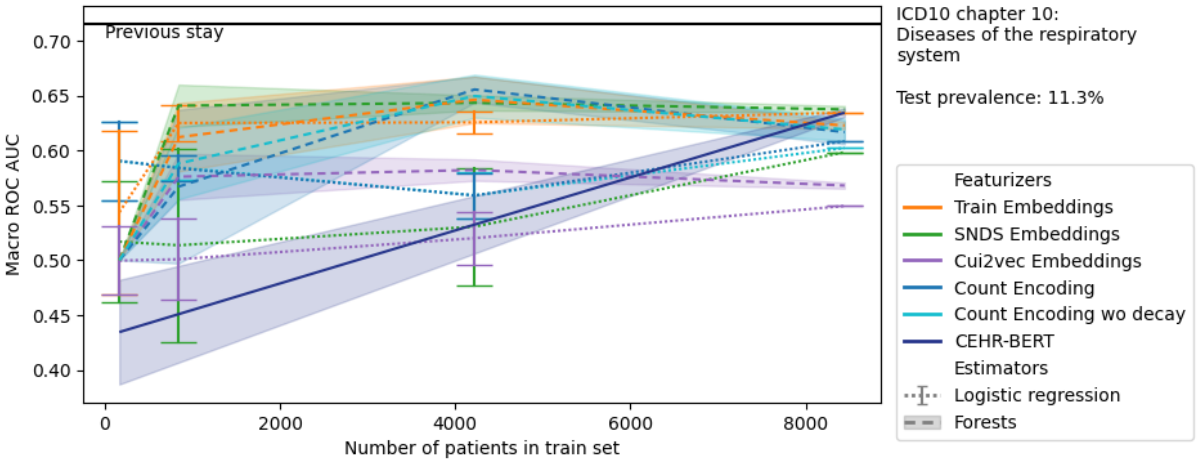

Previous stay
Macro ROC AUC
Number of patients in train set
ICD10 chapter 2:
Neoplasms
Test prevalence: 17.0%
Featurizers
Train Embeddings
SNDS Embeddings
Cui2vec Embeddings
Count Encoding
Count Encoding wo decay
CEHR-BERT
Estimators
Logistic regression
Forests

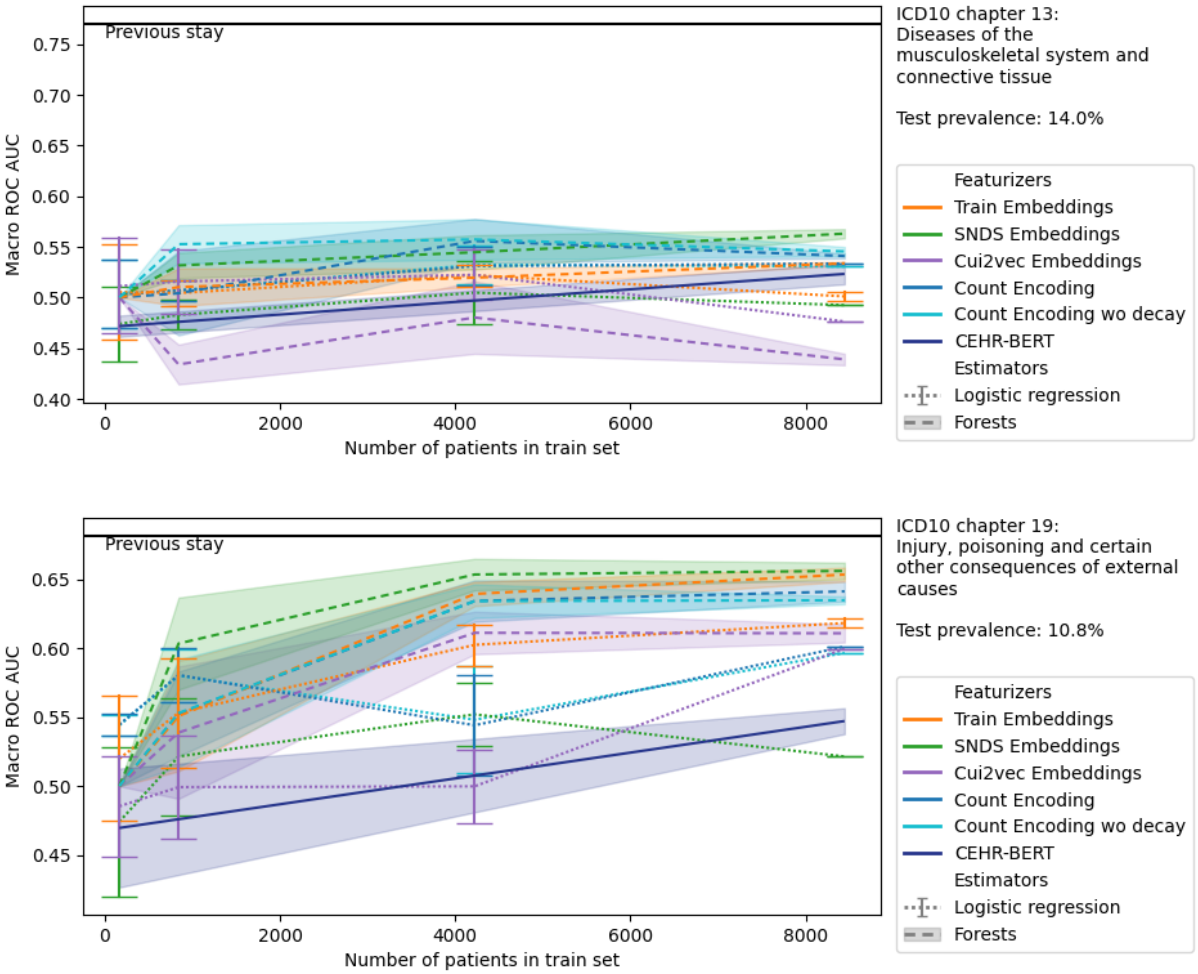

Previous stay
Macro ROC AUC
Number of patients in train set
ICD10 chapter 5:
Mental, Behavioral and
Neurodevelopmental disorders
Test prevalence: 15.9%
Featurizers
Train Embeddings
SNDS Embeddings
Cui2vec Embeddings
Count Encoding
Count Encoding wo decay
CEHR-BERT
Estimators
Logistic regression
Forests


Previous stay
Macro ROC AUC
Number of patients in train set
ICD10 chapter 6:
Diseases of the nervous system
Test prevalence: 14.7%
Featurizers
Train Embeddings
SNDS Embeddings
Cui2vec Embeddings
Count Encoding
Count Encoding wo decay
CEHR-BERT
Estimators
Logistic regression
Forests


Previous stay
Macro ROC AUC
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0.40
0
2000
4000
6000
8000
Number of patients in train set
ICD10 chapter 11:
Diseases of the digestive system
Test prevalence: 15.2%
Featurizers
Train Embeddings
SNDS Embeddings
Cui2vec Embeddings
Count Encoding
Count Encoding wo decay
CEHR-BERT
Estimators
Logistic regression
Forests


Previous stay
Macro ROC AUC
0.70
0.65
0.60
0.55
0.50
0.45
0.40
0
2000
4000
6000
8000
Number of patients in train set
ICD10 chapter 14:
Diseases of the genitourinary system
Test prevalence: 14.2%
Featurizers
Train Embeddings
SNDS Embeddings
Cui2vec Embeddings
Count Encoding
Count Encoding wo decay
CEHR-BERT
Estimators
Logistic regression
Forests


Previous stay
Macro ROC AUC
0.70
0.65
0.60
0.55
0.50
0.45
0.40
0
2000
4000
6000
8000
Number of patients in train set
ICD10 chapter 10:
Diseases of the respiratory system
Test prevalence: 11.3%
Featurizers
Train Embeddings
SNDS Embeddings
Cui2vec Embeddings
Count Encoding
Count Encoding wo decay
CEHR-BERT
Estimators
Logistic regression
Forests



When we look at the difference of performances depending on the prevalences, we have the following results for the 21 chapters. Different versions of this figure :

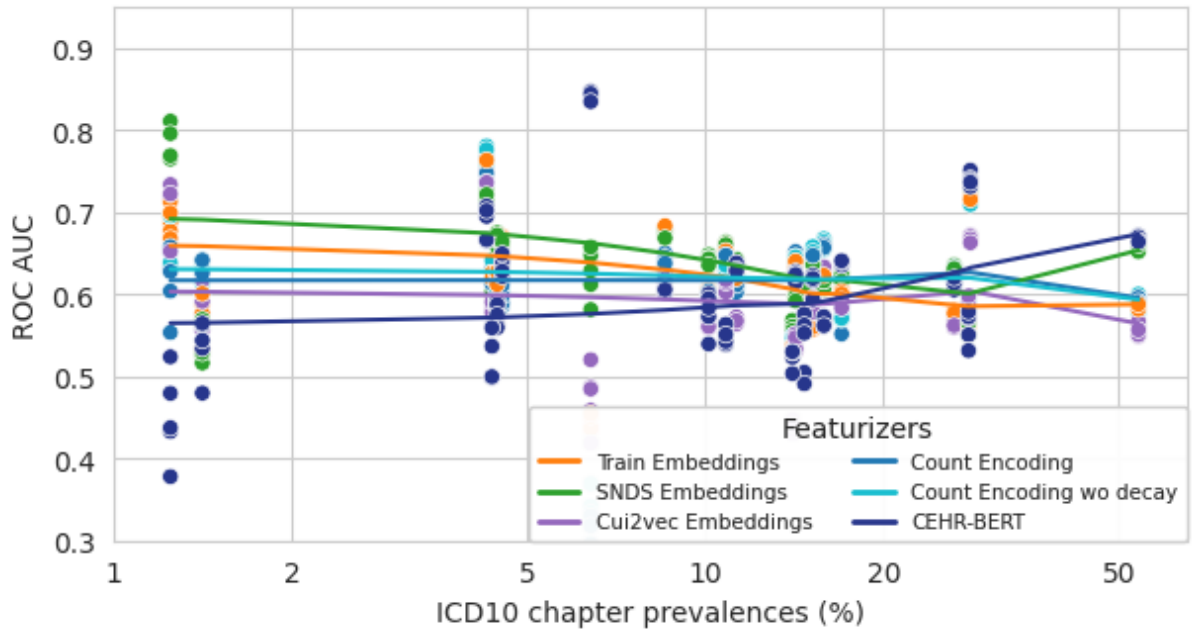


Figure 80: Except for CEHR-BERT, the performances of all featurizers chained with random forest estimators are independant from the chapter prevalence. Below 15% of prevalences, random forets manage to extract information from pretrained concept embedding methods and outperform CEHR-BERT.

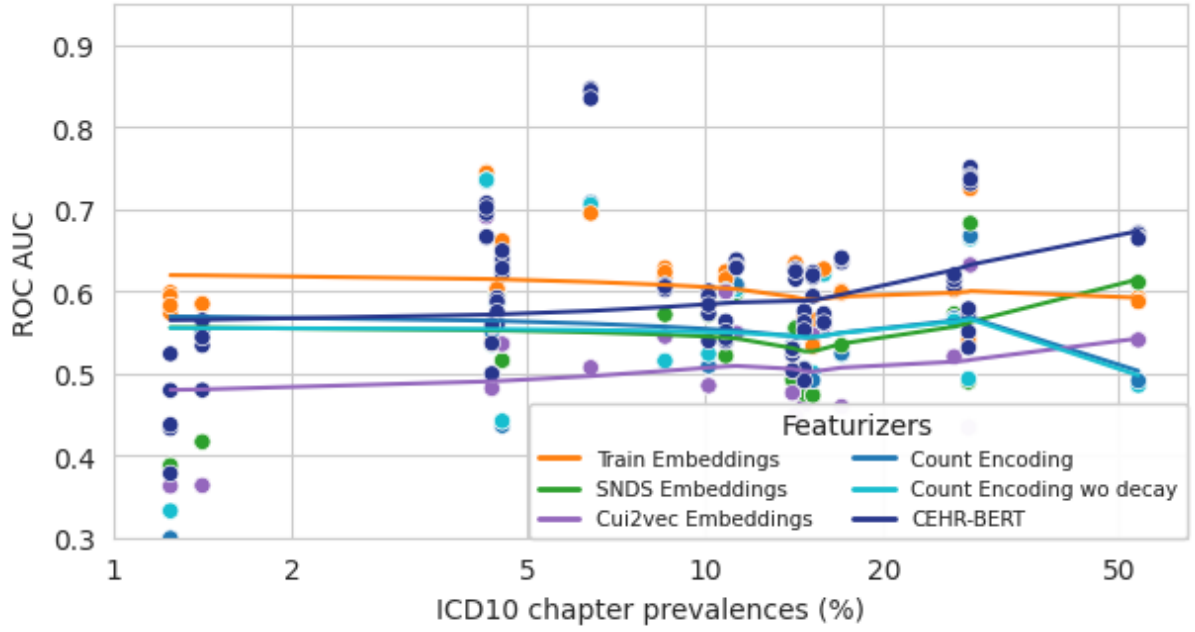


Figure 81: The performances of all featurizers chained with logistic regression benefit from higher chapter prevalences. The benefit are higher for CEHR-BERT.

1.4 Task 3: Predicting MACE complication in incident patients

1.4.1 Task definition

The cohort are the patients with a hospitalization (0 days and more ie. incomplete and complete) visits and at least 2 visits and at most 7 visits. The index visit is defined either as the first, last or radom visit respecting the following inclusion criteria: being during the study period: 01-01-2017/01-06-2022, having at least 360 days of followup (before 01-06-2022), no in-hospital mortality, aged over 18 at admission, with at least one billing code.

The task definition adds one exclusion criteria: No MACE during the first visit (incident MACE in the database). Then it searches for a MACE code during the 360 days of followup after the end of the index visit.

I found 20308 / 429860 = 4.72% prevalence of MACE in this cohort when studying the 01-01-2017/01-06-2022 period and the first visit as index. However this ratio drops to less than 1% when choosing the last visit as index.

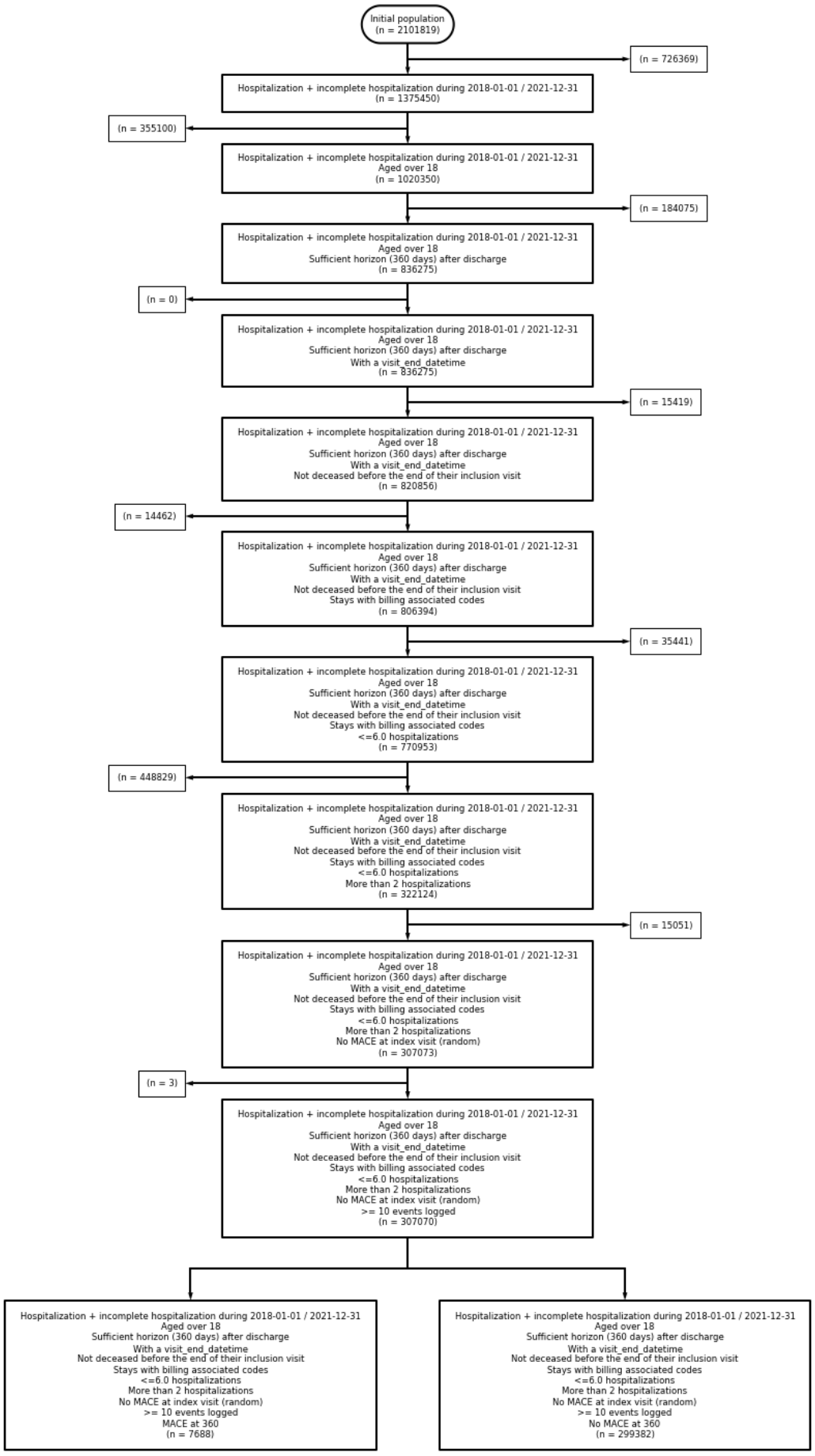
After looking at the occurences of MACE codes during the whole study period, I saw distributional shift with respect to time, ie. a drop in MACE codes after the end of 2021 and before 2018.

To stay on a stable regime, I restricted the study period to 2018-2021. For random index viit, it gives 5550 / 221455 = 2.51% prevalence of MACE in the cohort. For last index visit, it is only 1.05%. For first index visit, it is XX%.

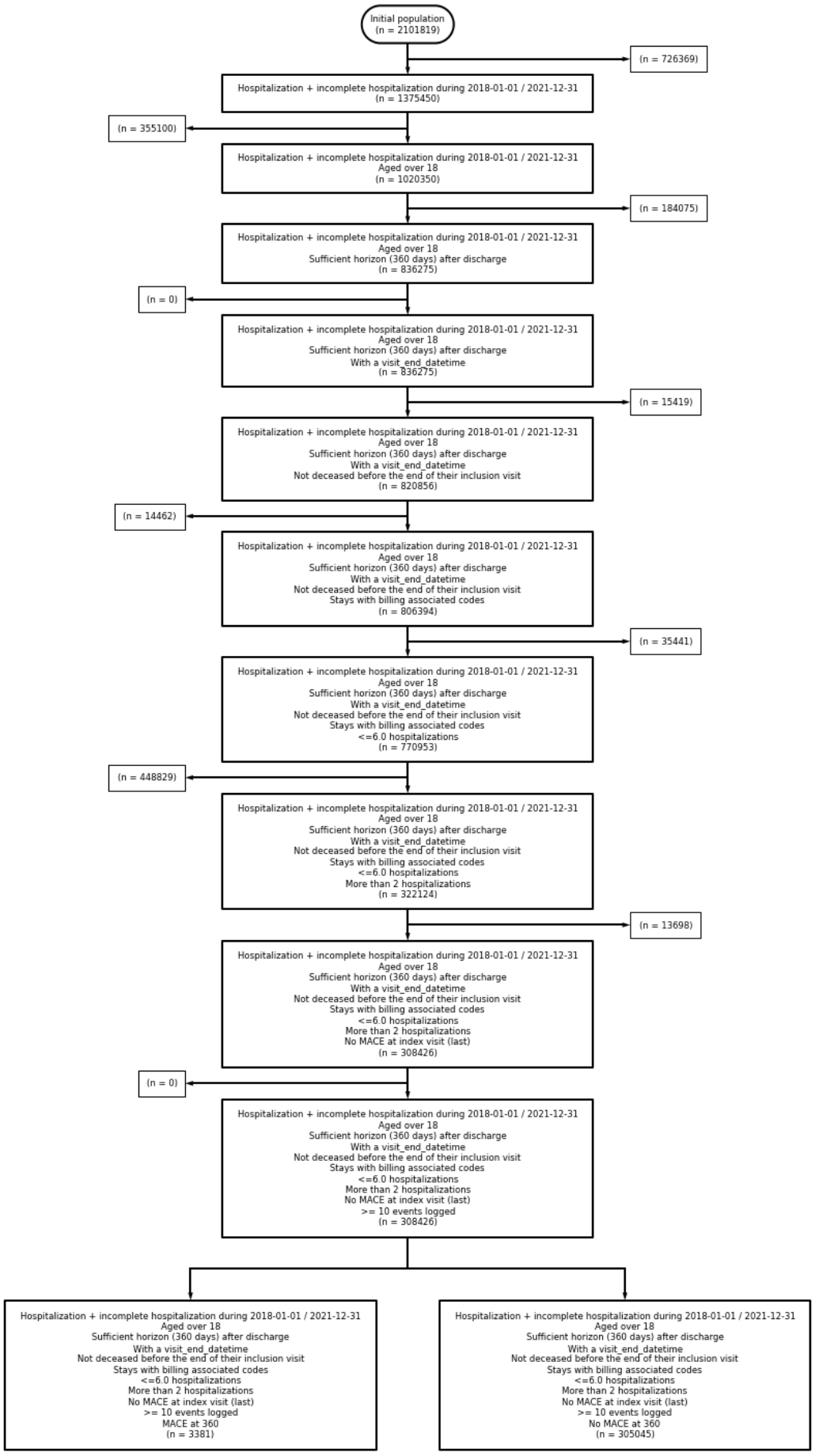
Enhancement:
- what to do with dead patients ? Target not considered yet during followup.
- Add a blank period (how much). What to do with patient meeting the criteria during the blank period ?
- Our patients are not perfectly incidents since we only select the MACE after the followup period. It might make more sense to remove patients having MACE before the start of the followup period (defined as the end of the index visit).

I moved away from the 200k omop samples to the 2M samples from the diabetes project. The reason are “MACE” is a common complication of diabetes so it is good application in addition the diabetics foot and it has sufficient sample size compared to the 200K samples where I found only: 500/10534 patients = 4.74% of prevalence, so not enough sample for reliable testing of the model.

1.4.1.1 Flowchart for random index visit and 2018-2021 period


Initial population
(n = 2101819)
(n = 726369)
Hospitalization + incomplete hospitalization during 2018-01-01 / 2021-12-31
(n = 1375450)
(n = 355100)
Hospitalization + incomplete hospitalization during 2018-01-01 / 2021-12-31
Aged over 18
(n = 1020350)
(n = 184075)
Hospitalization + incomplete hospitalization during 2018-01-01 / 2021-12-31
Aged over 18
Sufficient horizon (360 days) after discharge
(n = 836275)
(n = 0)
Hospitalization + incomplete hospitalization during 2018-01-01 / 2021-12-31
Aged over 18
Sufficient horizon (360 days) after discharge
With a visit_end_datetime
(n = 836275)
(n = 15419)
Hospitalization + incomplete hospitalization during 2018-01-01 / 2021-12-31
Aged over 18
Sufficient horizon (360 days) after discharge
With a visit_end_datetime
Not deceased before the end of their inclusion visit
(n = 820856)
(n = 14462)
Hospitalization + incomplete hospitalization during 2018-01-01 / 2021-12-31
Aged over 18
Sufficient horizon (360 days) after discharge
With a visit_end_datetime
Not deceased before the end of their inclusion visit
Stays with billing associated codes
(n = 806394)
(n = 35441)
Hospitalization + incomplete hospitalization during 2018-01-01 / 2021-12-31
Aged over 18
Sufficient horizon (360 days) after discharge
With a visit_end_datetime
Not deceased before the end of their inclusion visit
Stays with billing associated codes
<=6.0 hospitalizations
(n = 770953)
(n = 448829)
Hospitalization + incomplete hospitalization during 2018-01-01 / 2021-12-31
Aged over 18
Sufficient horizon (360 days) after discharge
With a visit_end_datetime
Not deceased before the end of their inclusion visit
Stays with billing associated codes
<=6.0 hospitalizations
More than 2 hospitalizations
(n = 322124)
(n = 15051)
Hospitalization + incomplete hospitalization during 2018-01-01 / 2021-12-31
Aged over 18
Sufficient horizon (360 days) after discharge
With a visit_end_datetime
Not deceased before the end of their inclusion visit
Stays with billing associated codes
<=6.0 hospitalizations
More than 2 hospitalizations
No MACE at index visit (random)
(n = 307073)
(n = 3)
Hospitalization + incomplete hospitalization during 2018-01-01 / 2021-12-31
Aged over 18
Sufficient horizon (360 days) after discharge
With a visit_end_datetime
Not deceased before the end of their inclusion visit
Stays with billing associated codes
<=6.0 hospitalizations
More than 2 hospitalizations
No MACE at index visit (random)
>= 10 events logged
(n = 307070)
Hospitalization + incomplete hospitalization during 2018-01-01 / 2021-12-31
Aged over 18
Sufficient horizon (360 days) after discharge
With a visit_end_datetime
Not deceased before the end of their inclusion visit
Stays with billing associated codes
<=6.0 hospitalizations
More than 2 hospitalizations
No MACE at index visit (random)
>= 10 events logged
MACE at 360
(n = 7688)
Hospitalization + incomplete hospitalization during 2018-01-01 / 2021-12-31
Aged over 18
Sufficient horizon (360 days) after discharge
With a visit_end_datetime
Not deceased before the end of their inclusion visit
Stays with billing associated codes
<=6.0 hospitalizations
More than 2 hospitalizations
No MACE at index visit (random)
>= 10 events logged
No MACE at 360
(n = 299382)

1.4.1.2 Flowchart for last index visit and 2018-2021 period


Initial population
(n = 2101819)
(n = 726369)
Hospitalization + incomplete hospitalization during 2018-01-01 / 2021-12-31
(n = 1375450)
(n = 355100)
Hospitalization + incomplete hospitalization during 2018-01-01 / 2021-12-31
Aged over 18
(n = 1020350)
(n = 184075)
Hospitalization + incomplete hospitalization during 2018-01-01 / 2021-12-31
Aged over 18
Sufficient horizon (360 days) after discharge
(n = 836275)
(n = 0)
Hospitalization + incomplete hospitalization during 2018-01-01 / 2021-12-31
Aged over 18
Sufficient horizon (360 days) after discharge
With a visit_end_datetime
(n = 836275)
(n = 15419)
Hospitalization + incomplete hospitalization during 2018-01-01 / 2021-12-31
Aged over 18
Sufficient horizon (360 days) after discharge
With a visit_end_datetime
Not deceased before the end of their inclusion visit
(n = 820856)
(n = 14462)
Hospitalization + incomplete hospitalization during 2018-01-01 / 2021-12-31
Aged over 18
Sufficient horizon (360 days) after discharge
With a visit_end_datetime
Not deceased before the end of their inclusion visit
Stays with billing associated codes
(n = 806394)
(n = 35441)
Hospitalization + incomplete hospitalization during 2018-01-01 / 2021-12-31
Aged over 18
Sufficient horizon (360 days) after discharge
With a visit_end_datetime
Not deceased before the end of their inclusion visit
Stays with billing associated codes
<=6.0 hospitalizations
(n = 770953)
(n = 448829)
Hospitalization + incomplete hospitalization during 2018-01-01 / 2021-12-31
Aged over 18
Sufficient horizon (360 days) after discharge
With a visit_end_datetime
Not deceased before the end of their inclusion visit
Stays with billing associated codes
<=6.0 hospitalizations
More than 2 hospitalizations
(n = 322124)
(n = 13698)
Hospitalization + incomplete hospitalization during 2018-01-01 / 2021-12-31
Aged over 18
Sufficient horizon (360 days) after discharge
With a visit_end_datetime
Not deceased before the end of their inclusion visit
Stays with billing associated codes
<=6.0 hospitalizations
More than 2 hospitalizations
No MACE at index visit (last)
(n = 308426)
(n = 0)
Hospitalization + incomplete hospitalization during 2018-01-01 / 2021-12-31
Aged over 18
Sufficient horizon (360 days) after discharge
With a visit_end_datetime
Not deceased before the end of their inclusion visit
Stays with billing associated codes
<=6.0 hospitalizations
More than 2 hospitalizations
No MACE at index visit (last)
>= 10 events logged
(n = 308426)
Hospitalization + incomplete hospitalization during 2018-01-01 / 2021-12-31
Aged over 18
Sufficient horizon (360 days) after discharge
With a visit_end_datetime
Not deceased before the end of their inclusion visit
Stays with billing associated codes
<=6.0 hospitalizations
More than 2 hospitalizations
No MACE at index visit (last)
>= 10 events logged
MACE at 360
(n = 3381)
Hospitalization + incomplete hospitalization during 2018-01-01 / 2021-12-31
Aged over 18
Sufficient horizon (360 days) after discharge
With a visit_end_datetime
Not deceased before the end of their inclusion visit
Stays with billing associated codes
<=6.0 hospitalizations
More than 2 hospitalizations
No MACE at index visit (last)
>= 10 events logged
No MACE at 360
(n = 305045)

1.4.1.3 Flowchart for first index visit and 2018-2021 period

1.4.2 Results for 2017-2022 period

1.4.3 First index visit

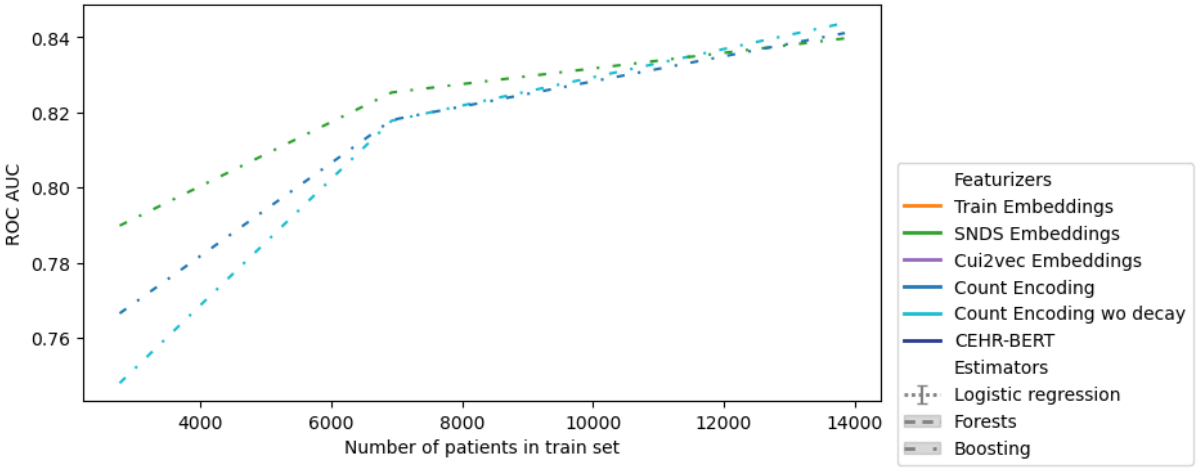


Figure 84: Results for MACE and HGB.

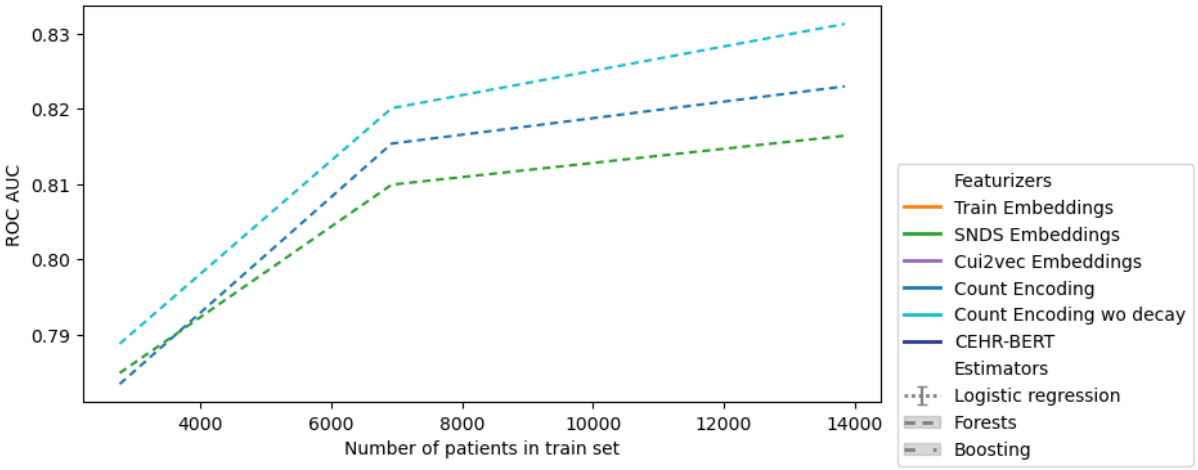


Figure 85: Results for MACE and forest.

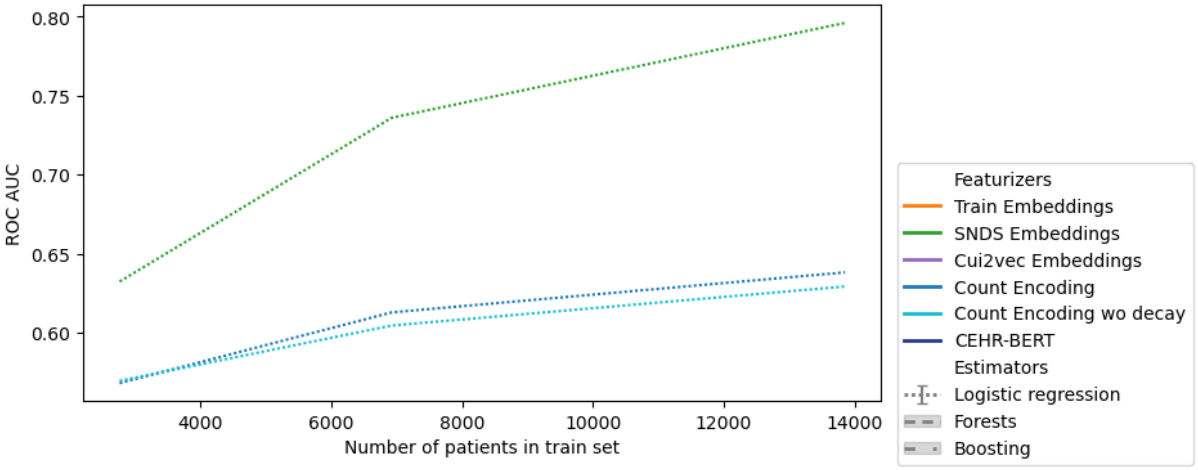


Figure 86: Results for MACE and ridge.

1.4.4 Results for 2018-2021 period (bad end of visits for incomplete hospitalizations)

1.4.4.1 Random index visit

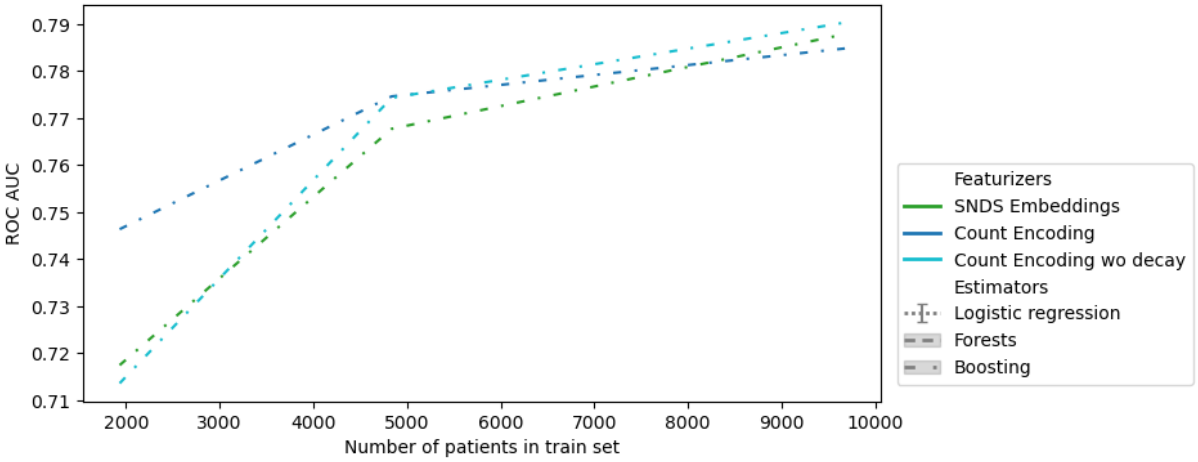


Figure 87: Results for MACE and HGB.

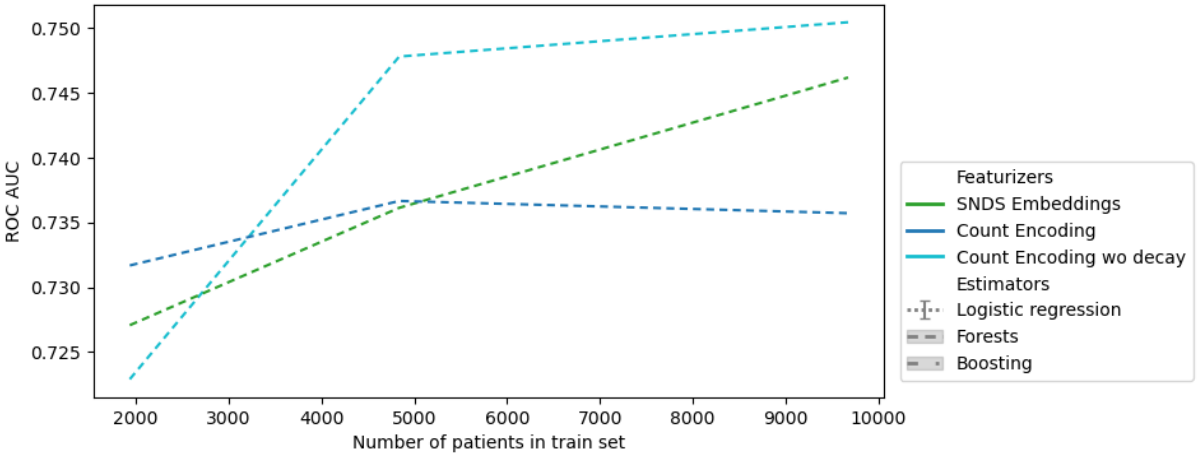


Figure 88: Results for MACE and forest.

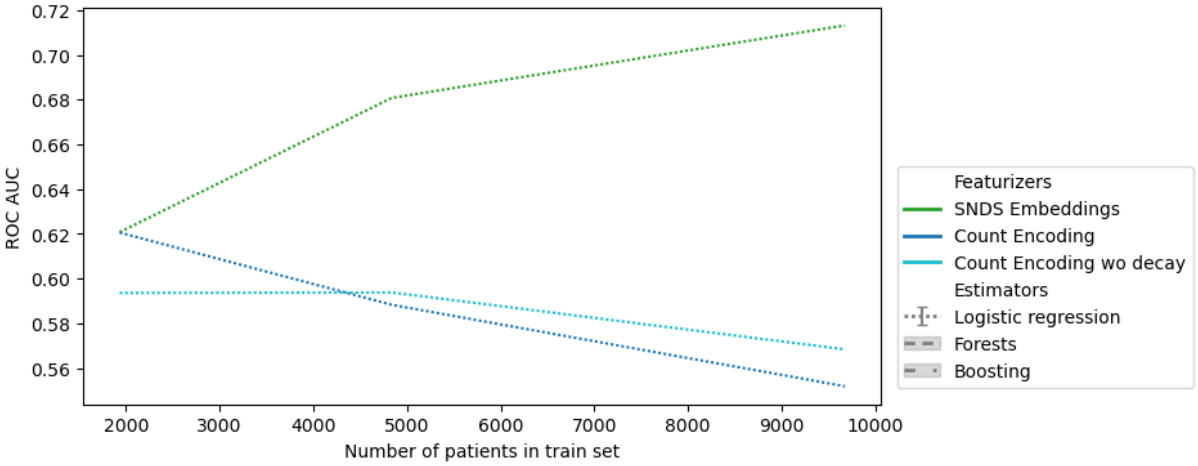


Figure 89: Results for MACE and ridge.

1.4.5 Results for 2018-2021 period (good end of visits for incomplete hospitalizations)

There is a strange drop in prevalence between test set (per hospitals, 1.30%) and train set (3.05%).

This seems to be due to MACE events arriving on the same day as discharge from index hospitalization present in the train but not in the test. I'm having trouble understanding this difference, but it completely rots the algorithm's performance, especially on the average_precision_score.

2 New experiments by time split

2.1 LOS

- 2023-08-10:
  - Testing the los experiment framework: subtrain grid=0.01, grid_decays = [[0], [0, 1], [0, 7], [0, 30], [0, 90]].
    - count featurizer: It takes 2-3 min to run over the 10 iterations of the RS.
    - All featurizers are passing. I prepare a big experience.
  - I ran the full ML models on slurm: train_grid=[0.01, 0.1, 0.25, 0.5, 0.75, 0.9, 1], grid_decays=[[0], [0, 1], [0, 7], [0, 30], [0, 90]], 10 iterations by random search. It fails for the snds2vec because of dimension mismatch. I suspect that it is the fault of the caching. I will remove caching and replace it by pretrained models that have been trained on the train sets (preambule of the prediction script if the embedding do not exist).
- 2023-08-11: Focusing on making ok all ML models (not c-bert).
  - Los is debugged and running
  - prognosis is debugged and running
  - mace is in test phase (30K patients only):
    - passed: subtrain_grid[0.01], models=ridge+forest, all featurizers, n_min_events=100.
    - passed : subtrain_grid[1], models=ridge+forest, with n_min_events=100, count featurizer. I see 7.5GB of memory for RF.
    - passed : subtrain_grid[1], models=ridge+forest, with n_min_events=10, count featurizer. I see 12GB of memory for LR, but passing, RF is fine with 11GB.
  - mace is in launch phase:
    - launched: subtrain_grid[0.1, 1], n_min_events=10, all featurizers