Predicting Sepsis on the PredictEM Masked Blutkulturen Dataset - June

Bachelor Thesis by Yannick Müller

What I did

1. More Feature engineering: minimum, maximum, multiplication and division did not improve the model
2. ROC_AUC Curve
3. Finally got same score with very simple Dense Neural Network. Everything more advanced made it worse.
4. Some measurements only done at one point in time: Combined all timesteps of patient. Made ROC_AUC score better in the second decimal after comma.

Predicting blood culture positive

Average Score 8-Fold Validation Set	F1 Score	AUC_ROC
Linear Regression	0.45	0.72
Ridge Classifier	0.45	0.72
Ridge Regression	0.45	0.72
Dense NN	0.45	0.72

A coin with 20 % chance of being true has a F1-Score of 0.29

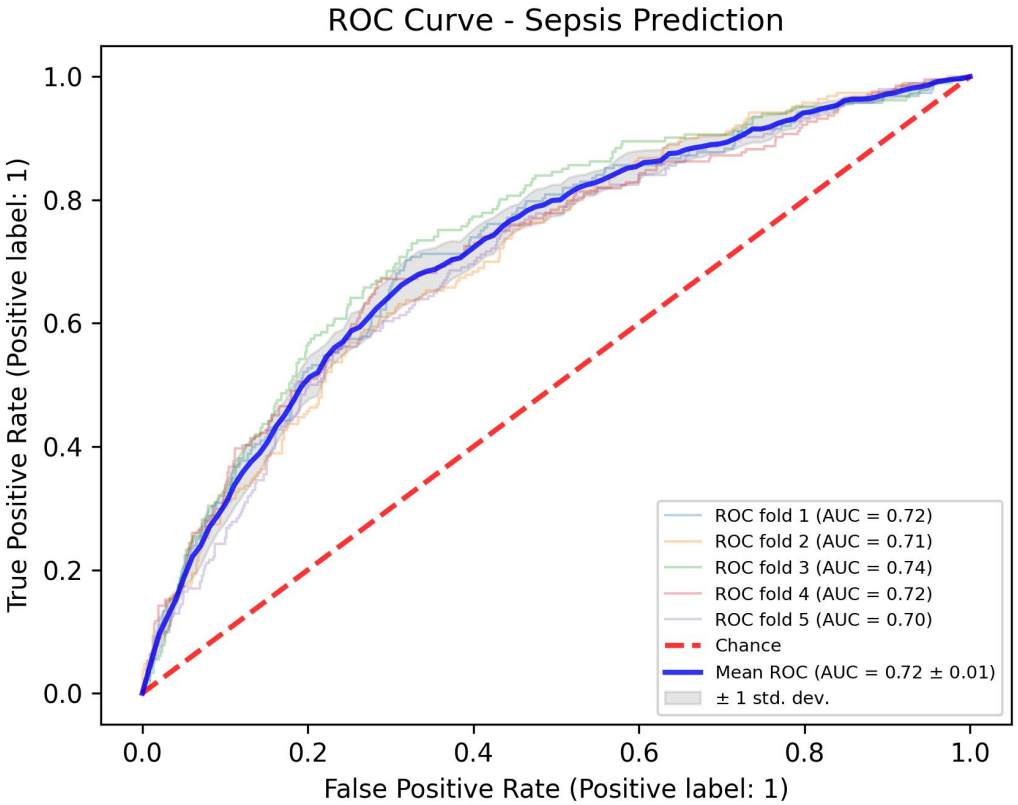
Preprocessing Example

```
from sklearn.model_selection import KFold
from sklearn.linear_model import Ridge
from sklearn.metrics import roc_auc_score, f1_score
import pandas as pd
import numpy as np

minmaxint = lambda x : int(round(min(1,max(0,x)),0))

cols = ["age_admission_unz", "triage", "referral_unz",
"diastolic_bp_first", "gcs_inf_15_first", "lvl_consc_alert",
"spo2_first", "THZ (Thrombozyten)", "EOS =eosinophile",
"CRP", "KA = Kalium", "Hb", "LACT =lactat", "GGT", "BIC_st",
"leuk_sup_10", "leuk_inf_4", "leuk_inf_0_5",
"LACT =lactatinf7", "INRiHinf5", "THZ (Thrombozyten)inf3",
"ASATinf3", "UREA = Harnstoffinf3",
"ort_vor_aufnahme_spitalinf2", "EOS =eosinophileinf6",
"respiratory_rate_firstinf3", "systolic_bp_firstinf4",
'Inselklinik', 'temperature_lowestinf9', 'EOS =eosinophileinf8']

df = pd.read_csv("sepsis_prediction5.csv")

X = df[cols]
y = df["blood_culture_positive"]
```

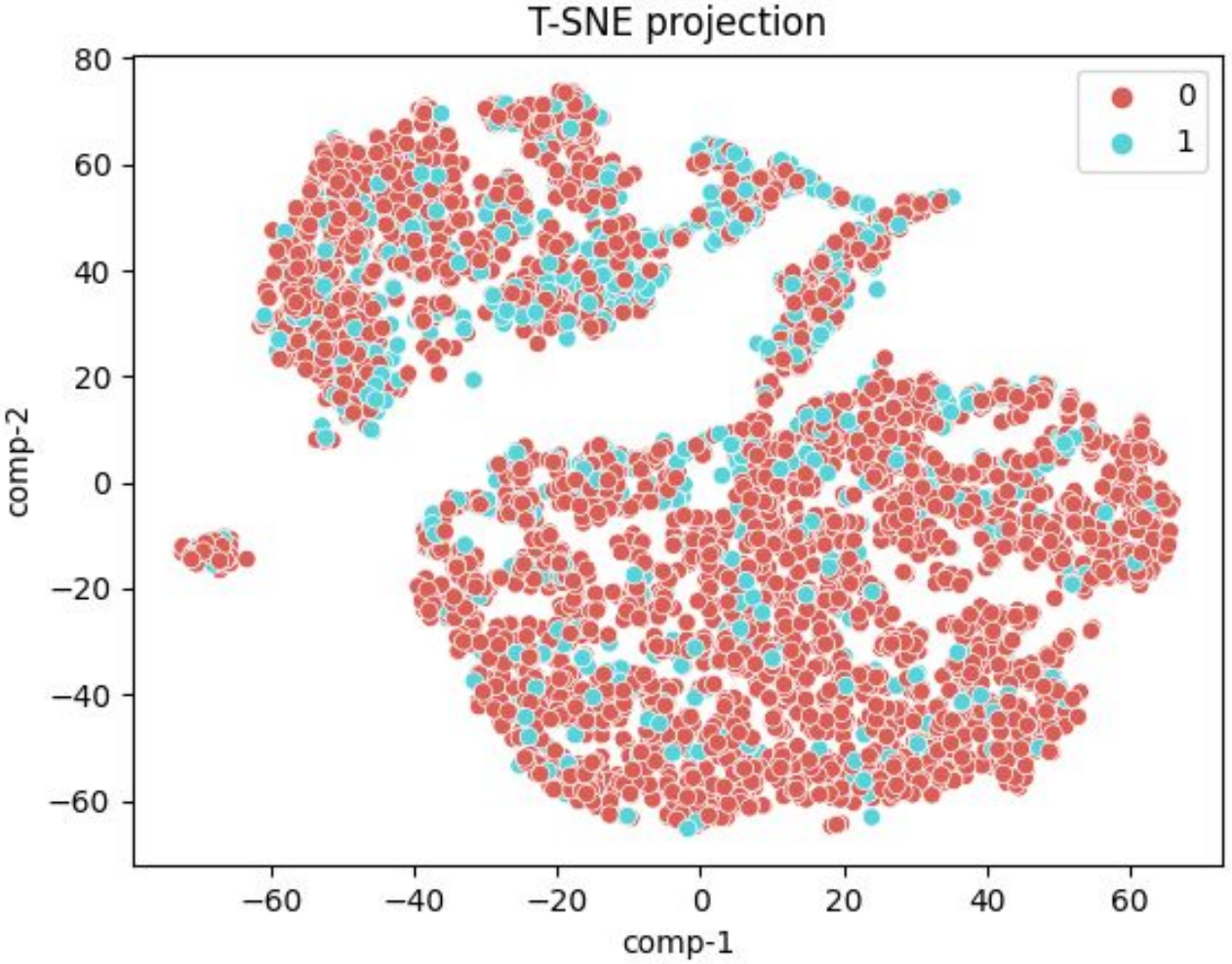
Example Ridge Classifier

```
25 kf = KFold(n_splits=5)
26 roc_auc_list = []
27 roc_auc_list_int = []
28 f1_list = []
29
30 for train_index, test_index in kf.split(X, y):
31         X_train, X_test = X.iloc[train_index], X.iloc[test_index]
32         y_train, y_test = y.iloc[train_index], y.iloc[test_index]
33         y_flatten = y_train.values.flatten()
34         bin_count = np.bincount(y_flatten)
35         class_weight = bin_count[0]/bin_count[1] - 1
36         clf = Ridge()
37         clf.fit(X_train, y_train, sample_weight=class_weight*y_flatten+1)
38         y_pred = clf.predict(X_test)
39         y_pred_int = list(map(minmaxint, y_pred))
40         roc_auc_list.append(roc_auc_score(y_test, y_pred))
41         roc_auc_list_int.append(roc_auc_score(y_test, y_pred_int))
42         f1_list.append(f1_score(y_test, y_pred_int))
43
44 print("This model:  ", np.mean(roc_auc_list), np.mean(roc_auc_list_int), np.mean(f1_list))
45 print("Current Best: 0.7183366311037686 0.6764051644236325 0.45046285449129")
```

AUC_ROC_Curve



T-SNE


T-SNE projection
0
1
comp-2
comp-1
80
60
40
20
0
−20
−40
−60
−60
−40
−20
0
20
40
60

Questions

1. As combining the data of different time samples of the same patient only improved model a bit, dismiss it?
2. Is 0.72 ROC_AUC score high enough to be useful?
3. How can I visualize Linear Regression with more than 2 Dimensions? PCA and then run the model again?
4. Papers which perform better have more than twice as many patients, and more features, specifically to predict sepsis. Can we enrich this dataset?
5. In papers so many things seem to work, like Ensemble learning. Am I doing something wrong or is it the dataset?

Up Next

1. Predict Mortality rate
2. Predict what kind of medicine is needed. Not high enough accuracy to do that?
3. Concentrate more on the writing part