covid-data-challenge

Task 1: Clinical data imputation

Method

We use KNN-3/KNN-6/KNN-9 and SoftImpute with fancyimpute, and a voting method to choose the best imputation method. The results are organized in folder below.

```
csv
├─feature_importance
├─filled_testSet
├─filled_trainSet
└─prediction_testSet
```

For a imputation method [method] in [filled_knn_3/filled_knn_6/filled_knn_9/filled_softimpute/] :

- ./csv/filled_testSet/[method]_testSet.txt
  - Filled **NaN** data using [method] based on **concatenation** of trainSet.txt & testSet.txt.
  - There are columns filled with **NaN** so we need **concatenation** of trainSet.txt & testSet.txt, otherwise nothing meaningful imputation can be obtained.
- ./csv/filled_trainSet/[method]_trainSet.txt
  - Filled **NaN** data using [method] based on trainSet.txt. (or the **concatenation** if needed (Question 1))
- ./csv/feature_importance/[method].csv
  - **Feature importance** of an ensemble methods consist of **random forest**, **gradient boost**, **adaboost** and **xgboost** predicting the **severity** with **filled training dataset** with **5-fold cross-validation**.
- ./csv/prediction_testSet/[method]_pred.txt
  - Prediction of a **voting** method using [method] filled testSet.txt with only clinical data.
  - TODO

Clinical data prediction

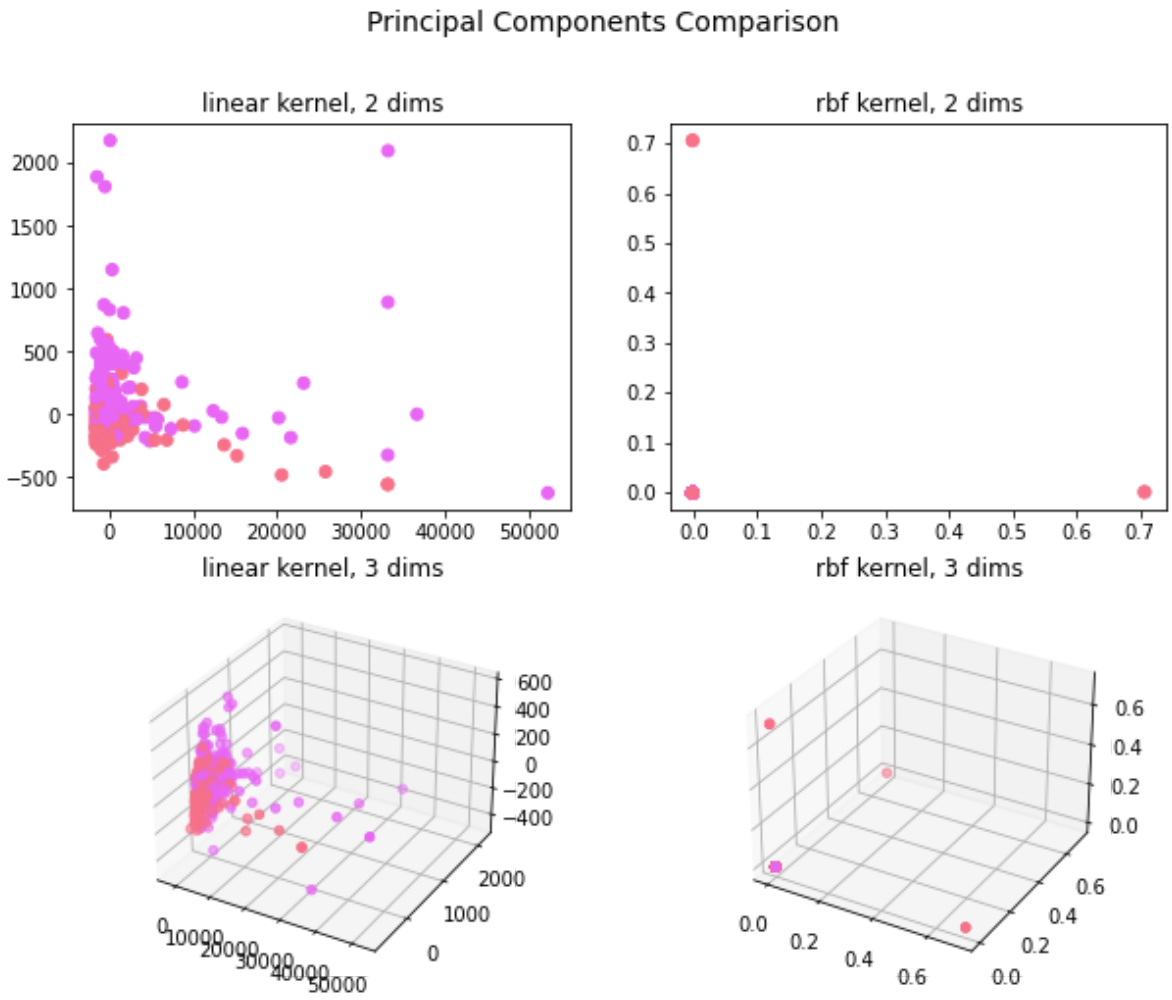
We verify the imputation by predicting the **severity** with **filled** training clinical data only with **5-fold cross-validation**.

KNN-3 imputation

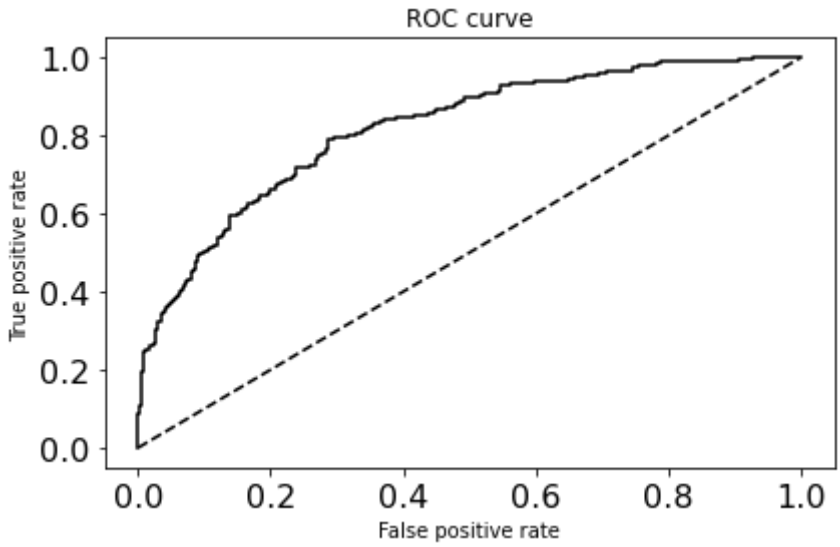
5-fold cross validation summary

	roc	acc	recall	cm	sensitivity	specificity
svm	0.582933	0.570104	0.684706	[[201 237] [134 291]]	0.684706	0.458904
knn	0.723202	0.653534	0.524706	[[341 97] [202 223]]	0.524706	0.778539
naive bayes	0.804169	0.70336	0.534118	[[380 58] [198 227]]	0.534118	0.86758
mlp	0.643368	0.644264	0.647059	[[281 157] [150 275]]	0.647059	0.641553
random forest	0.816409	0.73117	0.724706	[[323 115] [117 308]]	0.724706	0.737443
gradient boost	0.815257	0.747393	0.734118	[[333 105] [113 312]]	0.734118	0.760274
logistic	0.811571	0.728853	0.703529	[[330 108] [126 299]]	0.703529	0.753425
adaboost	0.787518	0.723059	0.701176	[[326 112] [127 298]]	0.701176	0.744292
xgboost	0.801144	0.707995	0.687059	[[319 119] [133 292]]	0.687059	0.728311
voting	**0.817846**	0.732329	0.670588	[[347 91] [140 285]]	0.670588	0.792237

t-SNE

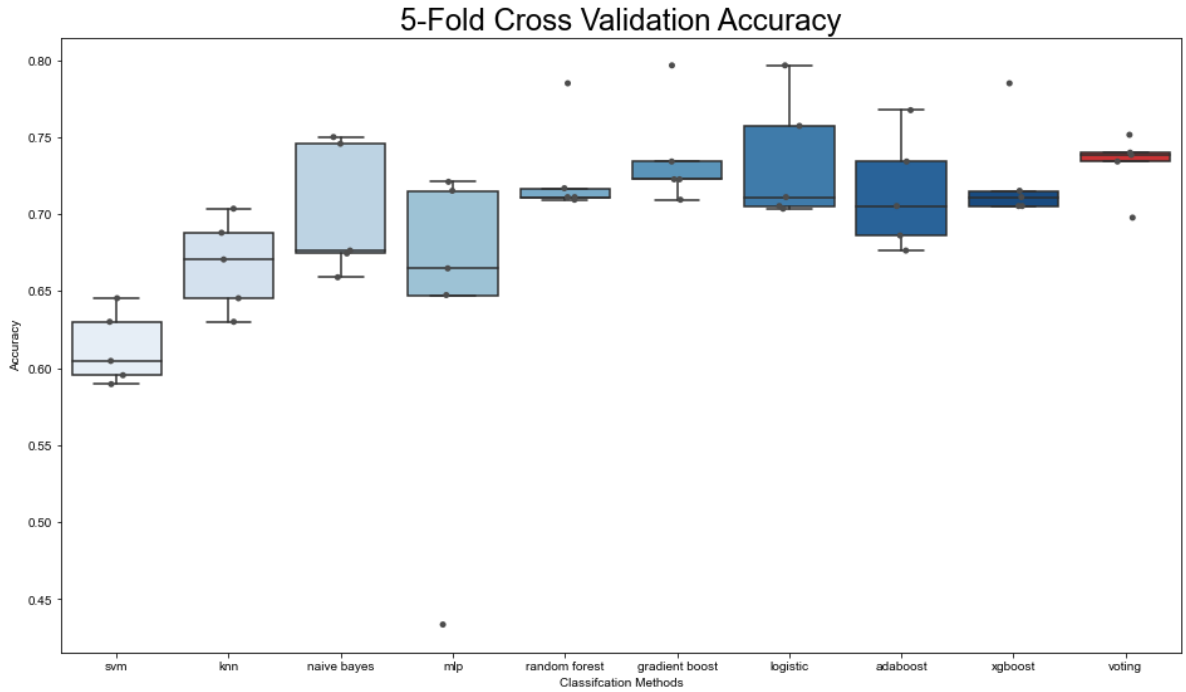


ROC curve

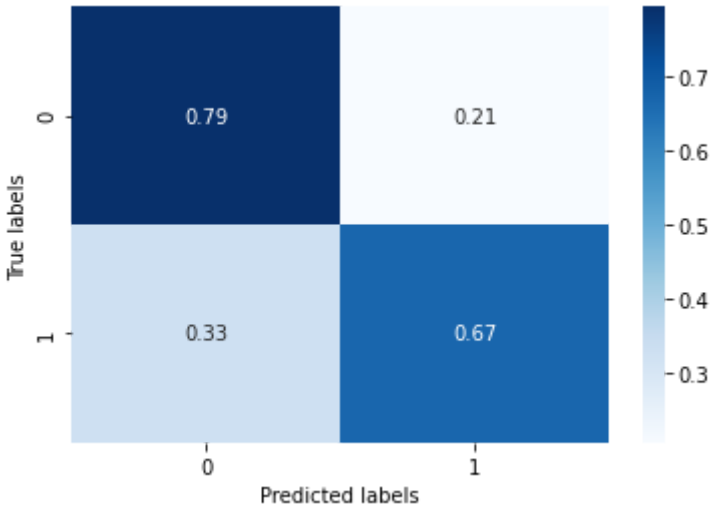


ROC AUC score of voting methods: **0.8178458232608112**

Accuracy@0.5



Confusion matrix@0.5

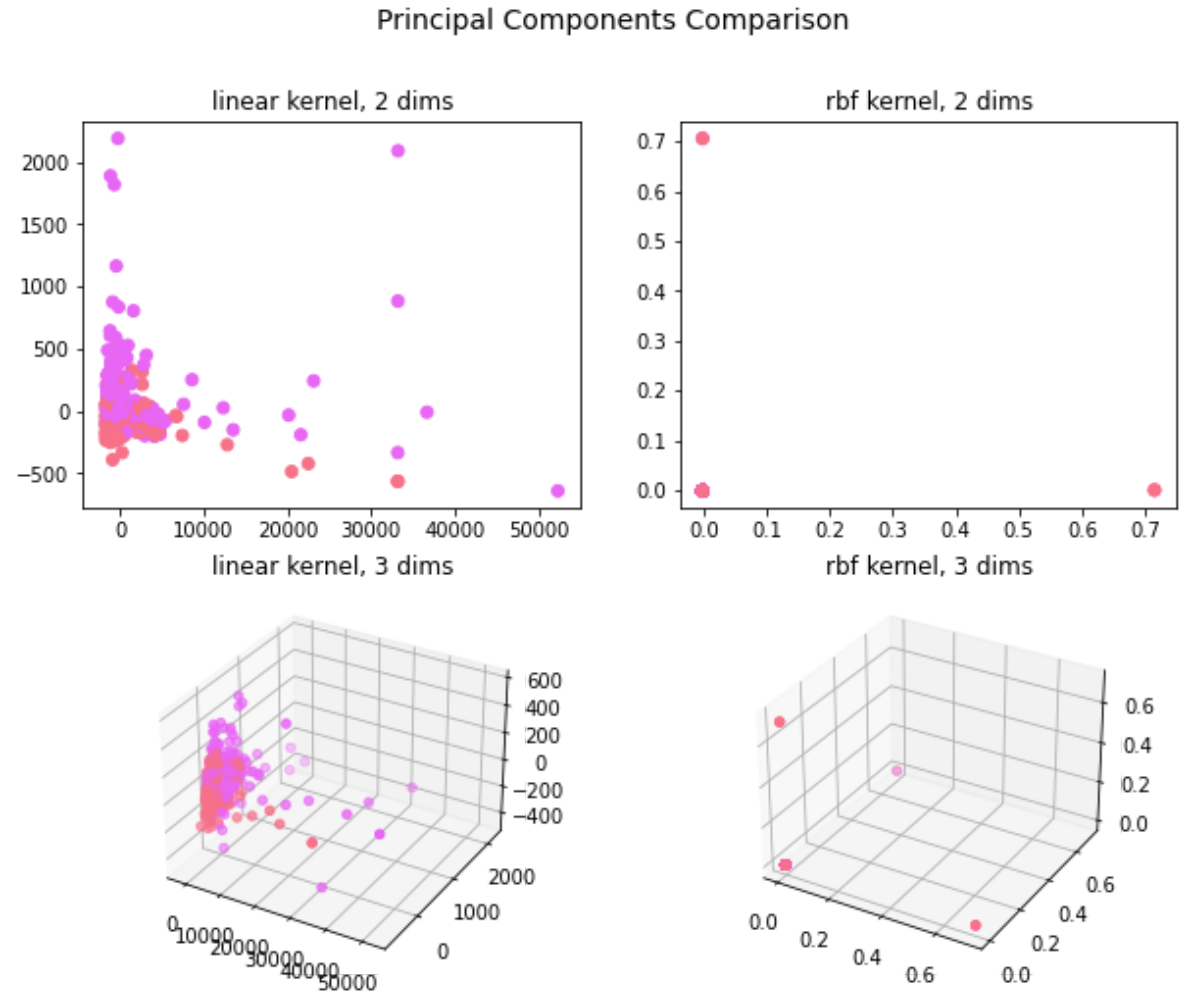


KNN-6 imputation

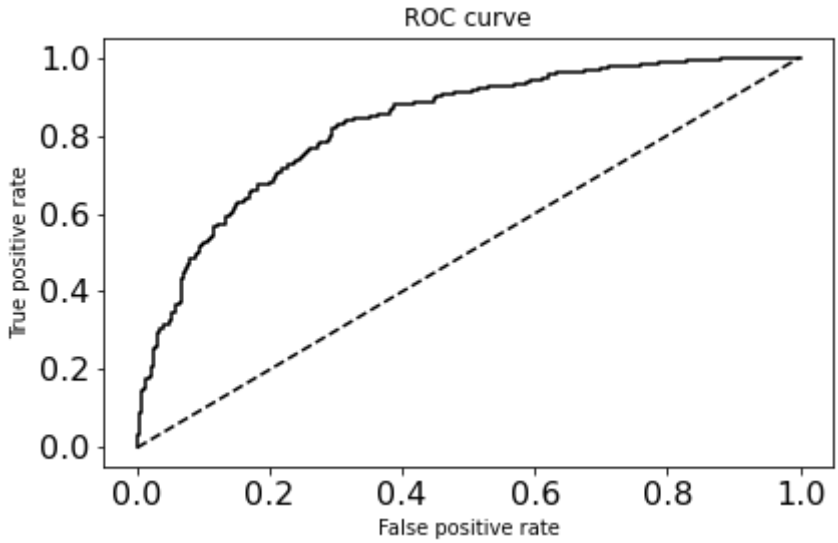
5-fold cross validation summary

	roc	acc	recall	cm	sensitivity	specificity
svm	0.616151	0.625724	0.804706	[[198 240] [83 342]]	0.804706	0.452055
knn	0.75198	0.681344	0.550588	[[354 84] [191 234]]	0.550588	0.808219
naive bayes	0.809213	0.717265	0.552941	[[384 54] [190 235]]	0.552941	0.876712
mlp	0.537749	0.559676	0.576471	[[238 200] [180 245]]	0.576471	0.543379
random forest	0.817913	0.752028	0.727059	[[340 98] [116 309]]	0.727059	0.776256
gradient boost	0.809981	0.73117	0.708235	[[330 108] [124 301]]	0.708235	0.753425
logistic	0.818141	0.741599	0.710588	[[338 100] [123 302]]	0.710588	0.771689
adaboost	0.799484	0.728853	0.705882	[[329 109] [125 300]]	0.705882	0.751142
xgboost	0.804346	0.732329	0.705882	[[332 106] [125 300]]	0.705882	0.757991
voting	**0.822756**	0.738123	0.672941	[[351 87] [139 286]]	0.672941	0.801370

t-SNE

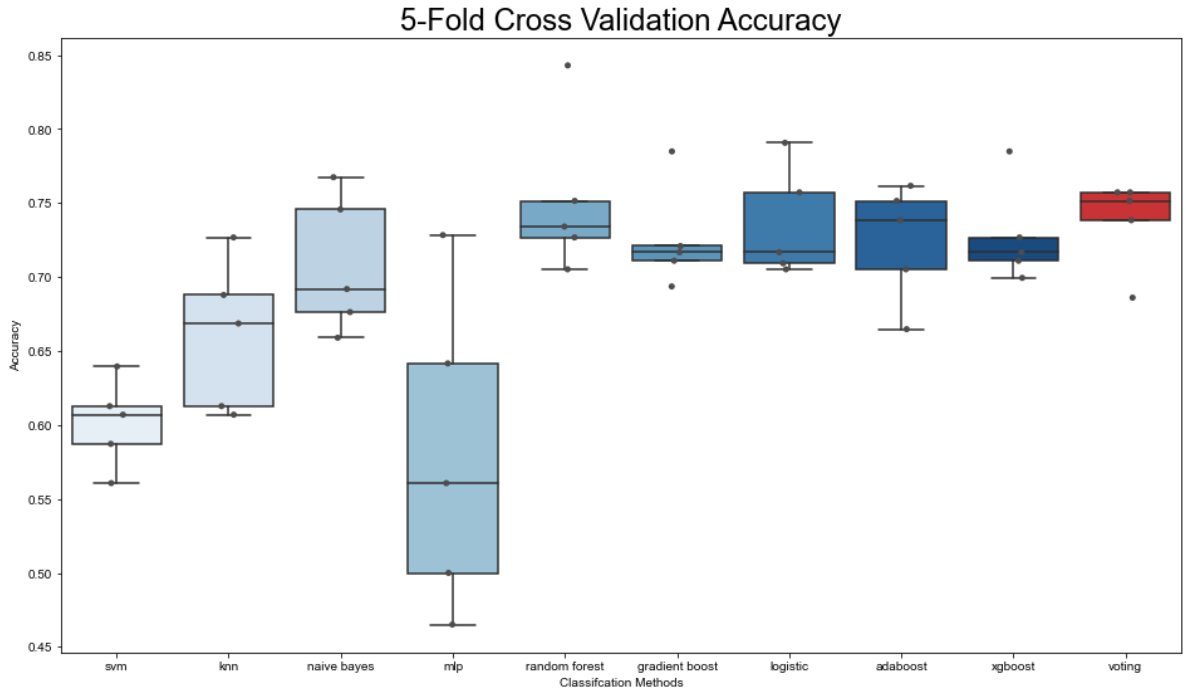


ROC curve

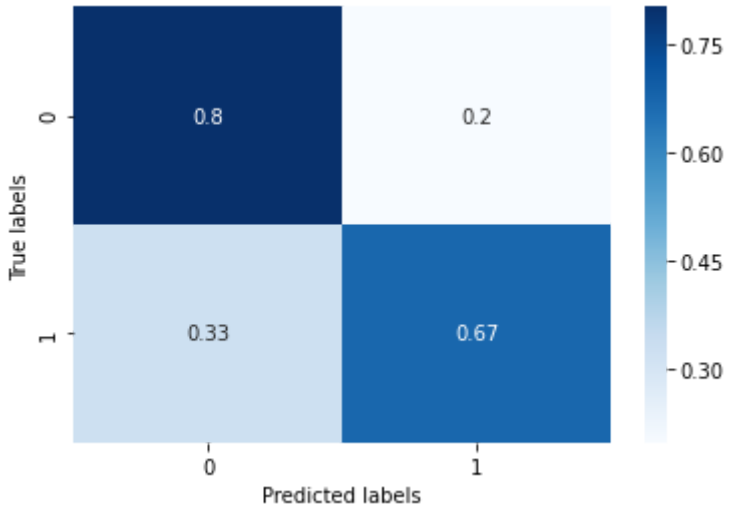


ROC AUC score of voting methods: **0.8227558420628526**

Accuracy@0.5



Confusion matrix@0.5

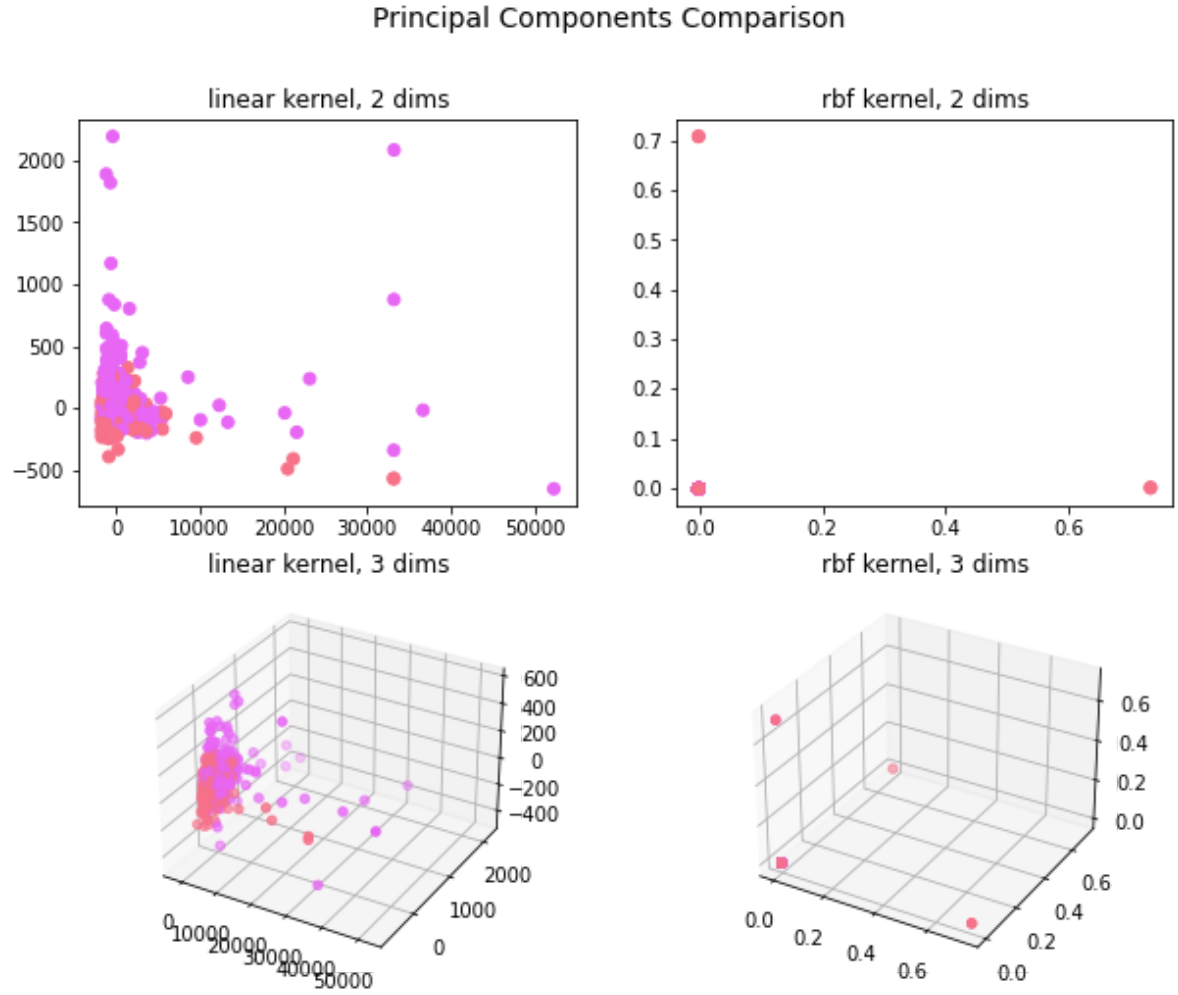


KNN-9 imputation

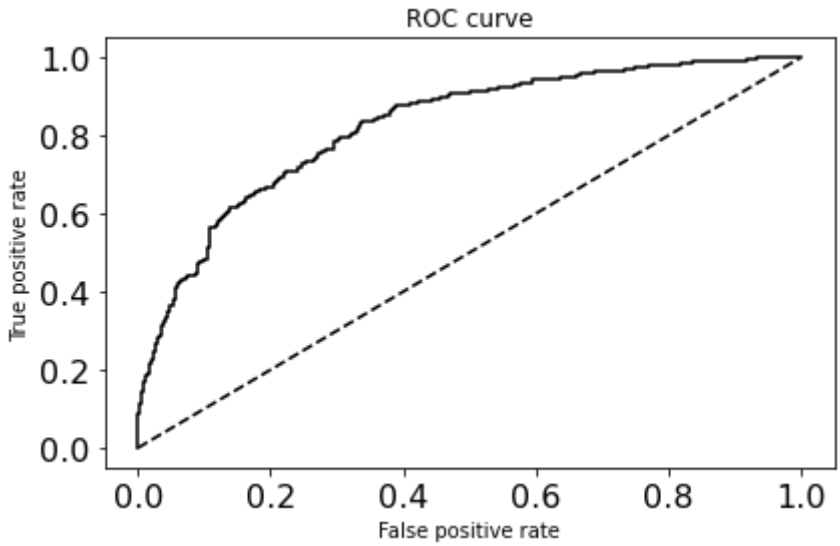
5-fold cross validation summary

	roc	acc	recall	cm	sensitivity	specificity
svm	0.620013	0.586327	0.651765	[[229 209] [148 277]]	0.651765	0.522831
knn	0.73618	0.668598	0.545882	[[345 93] [193 232]]	0.545882	0.787671
naive bayes	0.811026	0.71263	0.541176	[[385 53] [195 230]]	0.541176	0.878995
mlp	0.596433	0.604867	0.64	[[250 188] [153 272]]	0.64	0.570776
random forest	0.818383	0.739282	0.715294	[[334 104] [121 304]]	0.715294	0.762557
gradient boost	0.816095	0.733488	0.715294	[[329 109] [121 304]]	0.715294	0.751142
logistic	0.819645	0.734647	0.708235	[[333 105] [124 301]]	0.708235	0.760274
adaboost	0.801749	0.733488	0.696471	[[337 101] [129 296]]	0.696471	0.769406
xgboost	0.81134	0.730012	0.722353	[[323 115] [118 307]]	0.722353	0.737443
voting	**0.824072**	0.735805	0.675294	[[348 90] [138 287]]	0.675294	0.794520

t-SNE

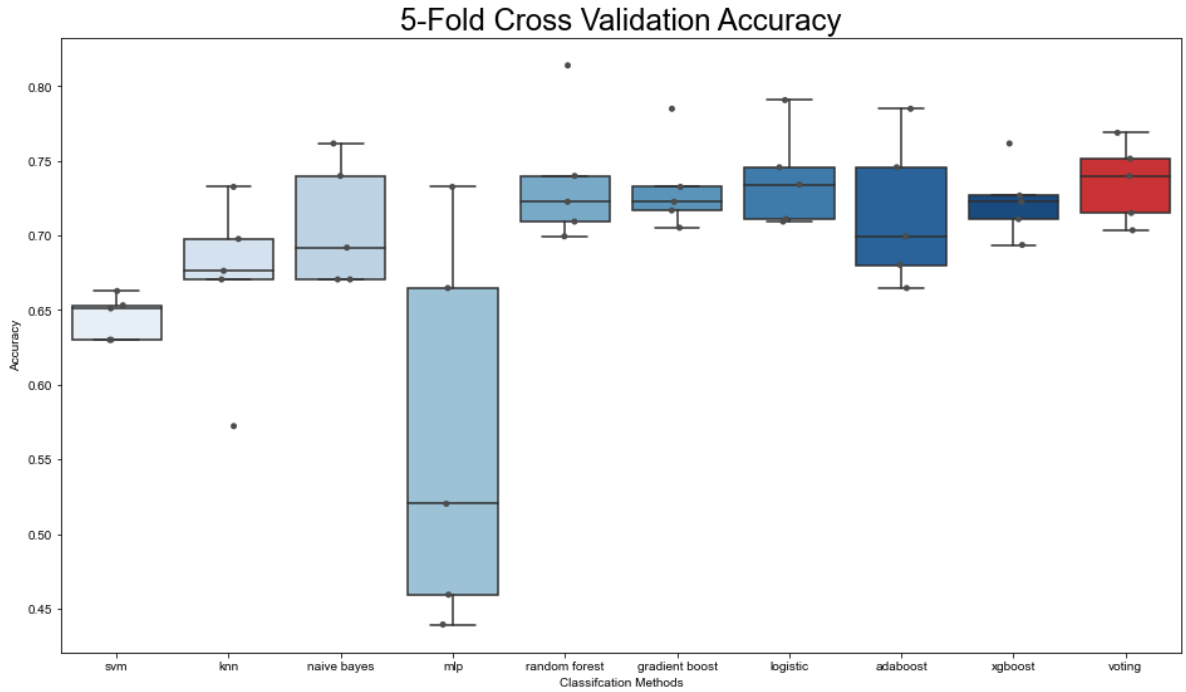


ROC curve

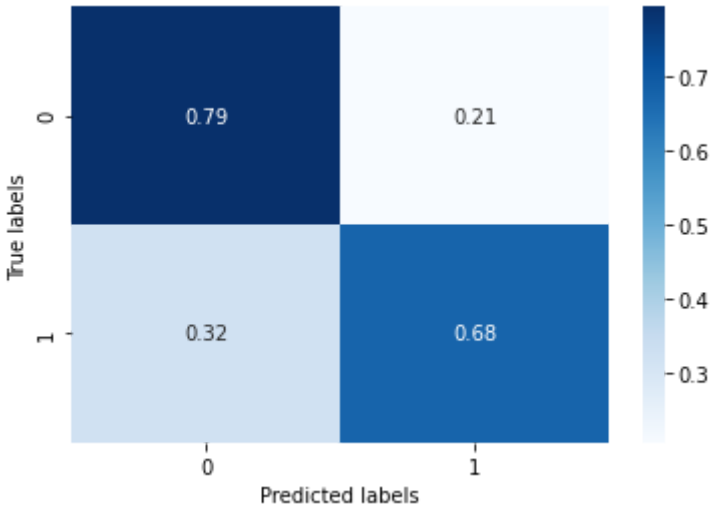


ROC AUC score of voting methods: **0.824071989583669**

Accuracy@0.5



Confusion matrix@0.5

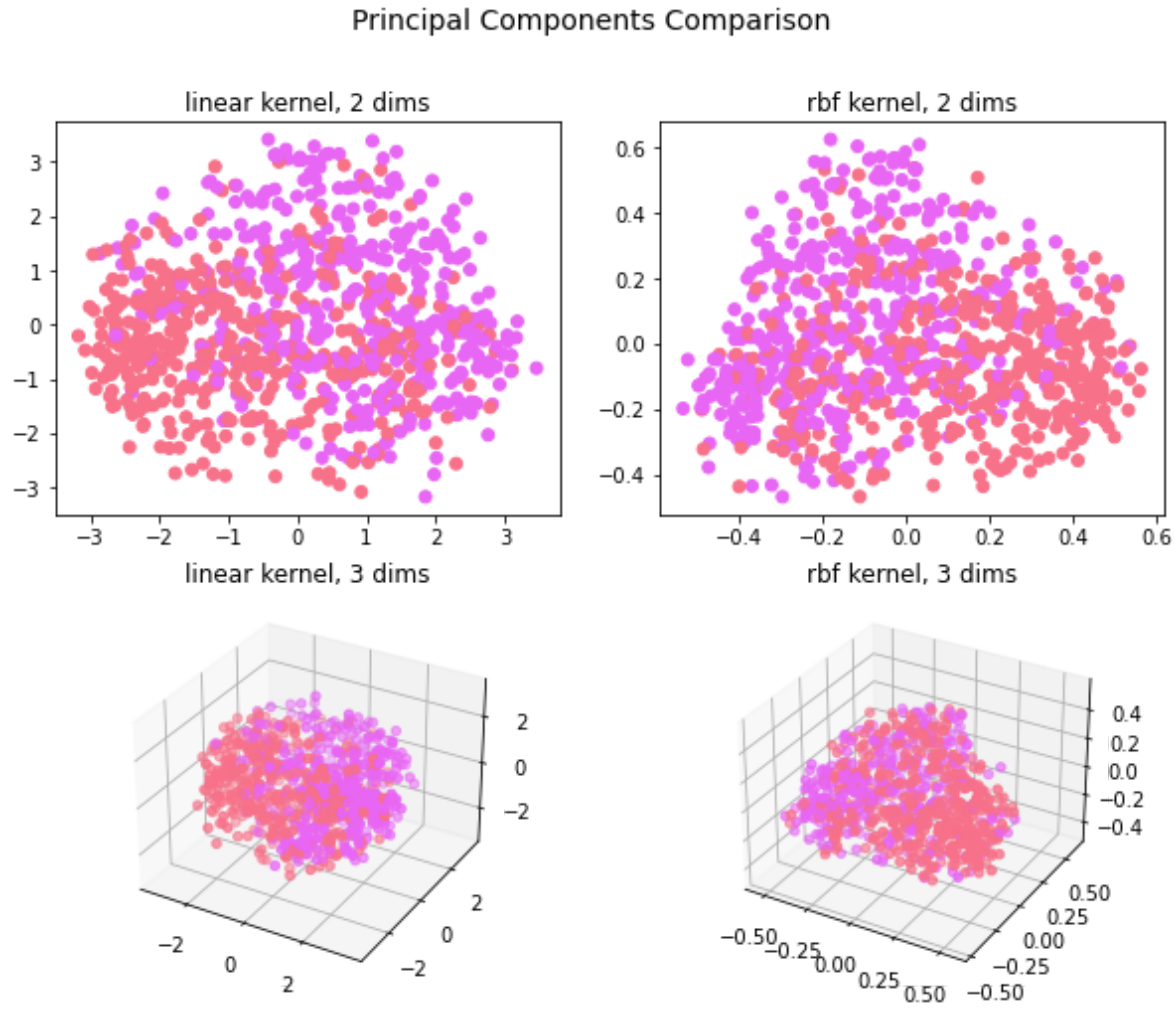


Soft imputation

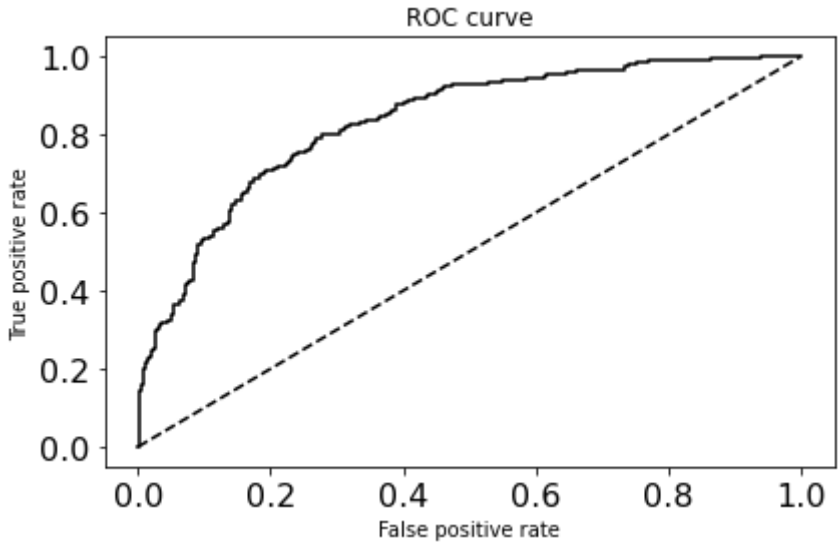
5-fold cross validation summary

	roc	acc	recall	cm	sensitivity	specificity
svm	0.825893	0.747393	0.741176	[[330 108] [110 315]]	0.741176	0.753425
knn	0.805181	0.750869	0.689412	[[355 83] [132 293]]	0.689412	0.810502
naive bayes	0.81899	0.743917	0.729412	[[332 106] [115 310]]	0.729412	0.757991
mlp	0.742673	0.677868	0.672941	[[299 139] [139 286]]	0.672941	0.682648
random forest	0.822909	0.738123	0.712941	[[334 104] [122 303]]	0.712941	0.762557
gradient boost	0.828695	0.738123	0.743529	[[321 117] [109 316]]	0.743529	0.732877
logistic	0.824453	0.752028	0.743529	[[333 105] [109 316]]	0.743529	0.760274
adaboost	0.792404	0.7219	0.708235	[[322 116] [124 301]]	0.708235	0.73516
xgboost	0.816288	0.726535	0.708235	[[326 112] [124 301]]	0.708235	0.744292
voting	**0.833059**	0.753186	0.738824	[[336 102] [111 314]]	0.738824	0.767123

t-SNE

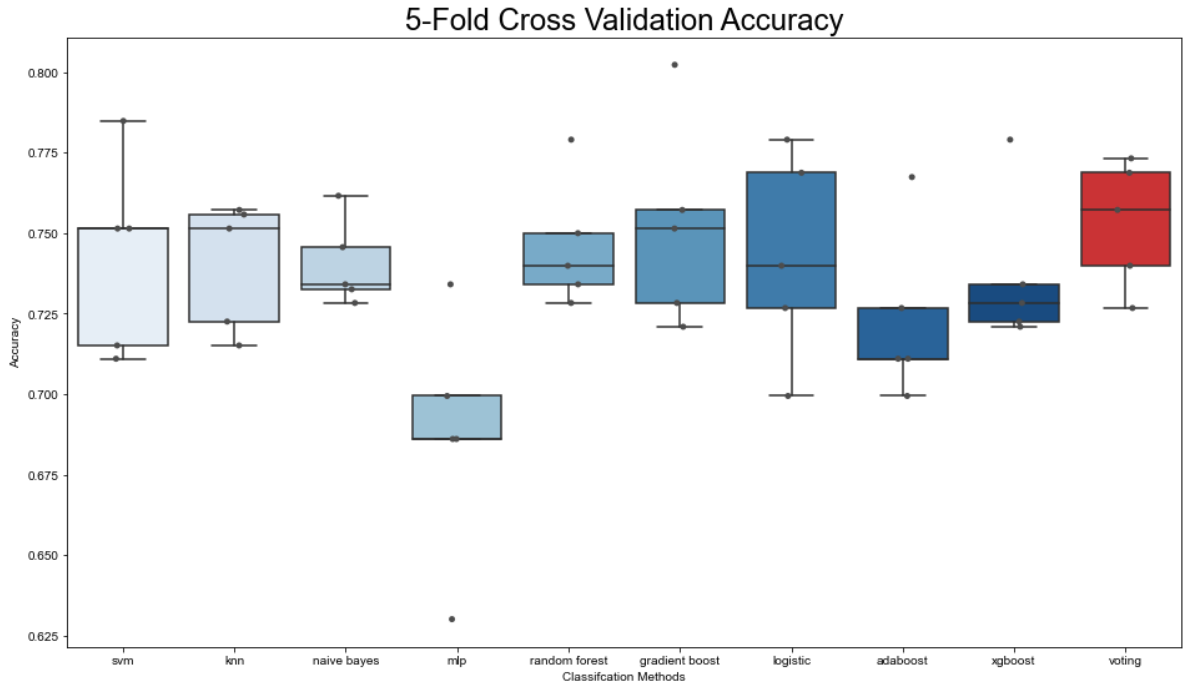


ROC curve



ROC AUC score of voting methods: **0.8330593607305936**

Accuracy@0.5



Confusion matrix@0.5