ICU Patients Data Analysis

Ahmet Ensar Köprülü
Enes Gülce

1. Objective

In this project we are working on the dataset ICU patients which contains the clinical and demographic information of intensive care unit patients. The issue that we are going to work on that is predicting the outcome of the discharge of the patient. Why is this issue important is because discharging patients quicker has the potential to increase the frequency of ICU readmissions but also quick discharges can lead to death on the other hand slow discharges may not be effective in terms of cost. Since we are predicting the outcome of discharge it is going to be a classification problem and the dataset we have contains the labels “death” and “readmission” so it a supervised classification problem.

2. Constraints

- **Heterogeneity and Imbalance of Data:**
  Many algorithms like neural networks and support vector machines like their feature vectors to be homogeneous numeric and normalized. The algorithms that employ distance metrics are very sensitive to this, and hence if the data is heterogeneous, these methods should be the afterthought. Decision Trees can handle heterogeneous data very easily. Moreover, reflects an unequal distribution of classes within a dataset. Most machine learning algorithms work best when the number of samples in each class are about equal. This is because most algorithms are designed to maximize accuracy and reduce error. Therefore, before training our models we standardize our numeric variables in order to scale for the specified models. Also, applied one hot encoding to the categorical variables since most of the algorithms we use works with numerical data. Furthermore, we used ADASYN(Adaptive Synthetic Sampling Method) to balance mostly negative death and readmitted targets.

- **Redundancy of Data/ Curse of Dimensionality:**
  If the data contains redundant information, i.e. contain highly correlated values, then it’s useless to use distance based methods because of numerical instability. In this case, some sort of Regularization can be employed to the data to prevent this Situation. Moreover, if the data scientist can manually remove irrelevant features from the input data, this is likely to improve the accuracy of the learned function. In addition, there are many algorithms for feature selection that seek to identify the relevant features and discard the irrelevant ones. Therefore, to compute relations between features and targets we mainly compare metrics such as pearson correlation score and p-values. Also, Lasso (L2) regularization will down effectiveness of unrelated features to zero and eliminate it.

- **Dependent Features:**
  If there is some dependence between the feature vectors, then algorithms that monitor complex interactions like Neural Networks and Decision Trees fare better than other algorithms.

3.Dataset:

In ICU Patients dataset there are 51 features and we will focus on death and readmitted variables. Top 5 correlated features for death is like:

Age	0.177991
LOS	0.073823
previous_LOS	0.058586
previous_ICU_stays	-0.007155
Charlson_index	0.002592

Statistics about top 5 correlated features of death is like:

	coef	std err	t	P>|t|	[0.025	0.975]
Age	0.0019	0.000	12.828	0.000	0.002	0.002
LOS	-2.893e-06	1.27e-05	-0.228	0.819	-2.77e-05	2.2e-05
previous_LOS	0.0002	3.02e-05	6.037	0.000	0.000	0.000
previous_ICU_stays	-0.0081	0.005	-1.588	0.112	-0.018	0.002
Charlson_index	0.0026	0.008	0.327	0.744	-0.013	0.018

Top 5 correlated features for readmitted like:

Age	0.041844
LOS	-0.011790
previous_LOS	0.062318
previous_ICU_stays	0.040352
Charlson_index	0.005481

Statistics about top 5 correlated features of readmitted is like:

	coef	std err	t	P>|t|	[0.025	0.975]
Age	0.0011	0.000	5.903	0.000	0.001	0.001
LOS	-8.48e-06	1.58e-05	-0.537	0.591	-3.94e-05	2.25e-05
previous_LOS	0.0001	3.76e-05	3.902	0.000	7.31e-05	0.000
previous_ICU_stays	0.0193	0.006	3.021	0.003	0.007	0.032
Charlson_index	-0.0056	0.010	-0.564	0.573	-0.025	0.014

By looking at the statistics above, even though a feature is good enough correlated it still may not be statistically significant. Therefore, one of them may work for the model and one of them may not. Thats why rather than computing relations between variables manually, we used recursive feature elimination methods to achieve that.

Also, the dataset is unbalanced(overwhelmingly negative) we oversampled the normal dataset by using ADASYN and create over sampled dataset. In total there are three datasets total like normal, oversampled and feature eliminated.

4. Results:

In total eight models are trained, tested and analyzed with various datasets. This part includes training and test results and performance measures in training order.

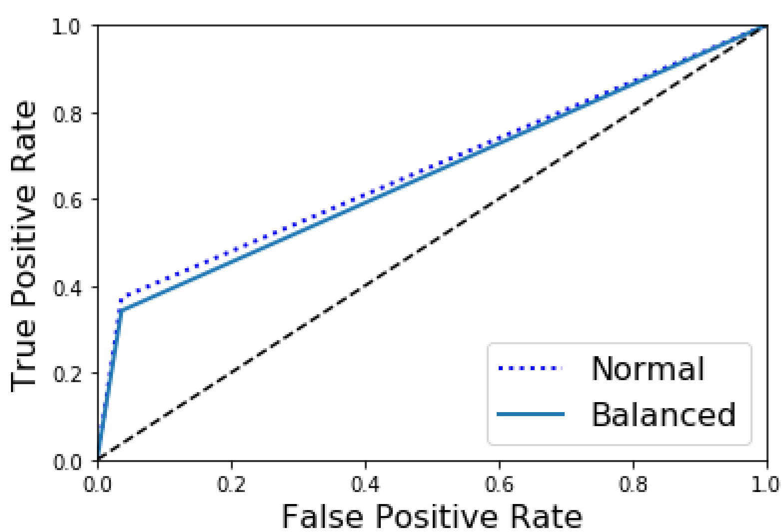
Logistic Regression

First model we tried to train was the logistic regression since it is simple and fast model and is a good start to training model.

Death Target

Training Results:
Training accuracy: 0.8471242425022683
Training accuracy for balanced samples: 0.7265990829264475

Test Results:
Test accuracy: 0.8446450809464509
F1 score: 0.49133537206931704
Test accuracy for balanced samples: 0.6500771604938271
F1 score: 0.4975069252077563
ROC Curve that includes both normal and balanced datasets



Readmitted Target
Training results:
Training accuracy: 0.8471238184095615

Test results:
Test accuracy: 0.8396637608966376

Since we are convinced that our logistic regression model is not enough successful to go with it we stop optimize it.

KNN

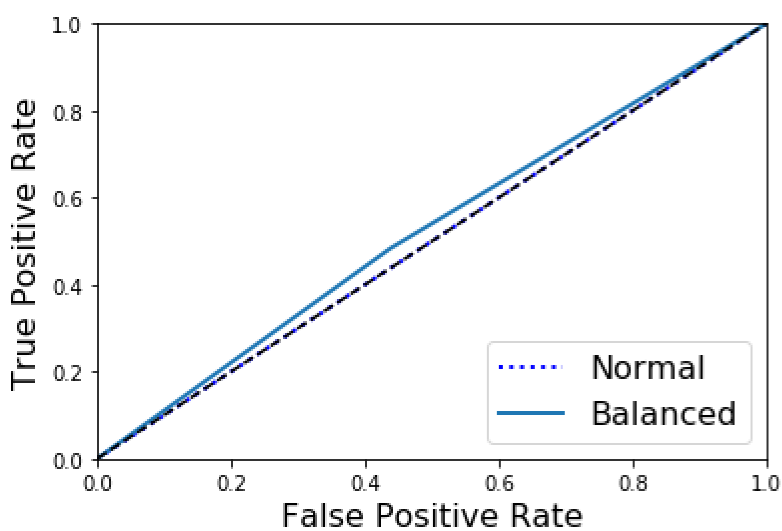
Second model we trained was KNN. Before training has been started, we fine tuned a knn model with grid search with 5 fold cross validation.

Death Target
Optimized parameters weight for uniform and distance and n neighbors for various numbers.
As a result of grid search we obtain those results:

Best training accuracy: 0.8102280861453457
Best parameters: {'n_neighbors': 23, 'weights': 'uniform'}

Test results:
Test accuracy: 0.798879202988792
F1 score: 0.0
Test accuracy for balanced samples: 0.5223765432098766
F1 score: 0.505591054313099

ROC Curve that includes both normal and balanced datasets



Readmitted Target
Optimized parameters weight for uniform and distance and n neighbors for various numbers.
As a result of grid search we obtain those results:

Best training Accuracy: 0.84727952101758
Best parameters: {'n_neighbors': 19, 'weights': 'uniform'}

Test results:
Test accuracy: 0.839975093399751
F1 score: 0.0

We also stop optimization of knn model since it does not seem promising to us.

Decision Tree

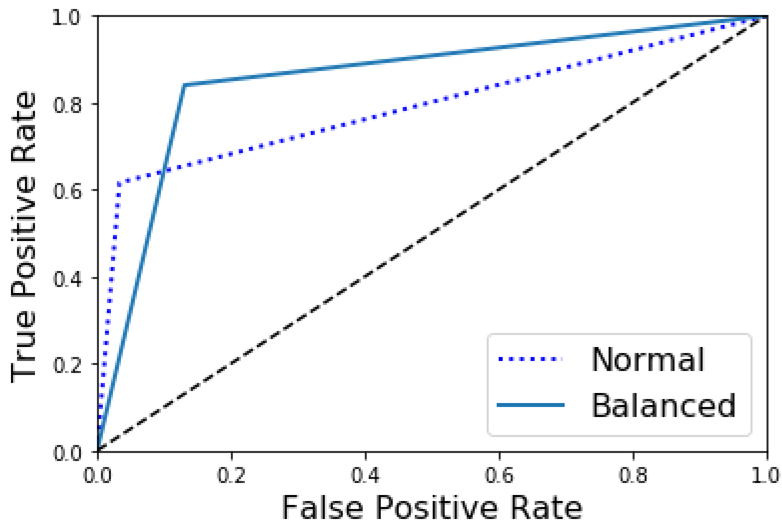
Third model we trained was Decision Tree. Before training has been started, we fine tuned a Decision Tree model with grid search with 5 fold cross validation.

Death Target
Optimized parameters max depth for 2, 4 and 10 and min sample split for 2, 3, 4. As a result of grid search we obtain those results:

Best training accuracy: 0.9048026984413079
Best parameters: {'max_depth': 4, 'min_samples_split': 3}

Test results:
Test accuracy: 0.8960149439601495
F1 score: 0.7044247787610619
Test accuracy for balanced samples: 0.8547453703703703
F1 score: 0.8538715311469047

ROC Curve that includes both normal and balanced datasets



Readmitted Target

Optimized parameters max depth for 2, 4 and 10 and min sample split for 2, 3, 4. As a result of grid search we obtain those results:

Best training Accuracy: 0.8493813850564876

Best parameters: {'max_depth': 4, 'min_samples_split': 4}

Test Results:

Test Accuracy: 0.8405977584059776 F1 score: 0.13220338983050847

Even though decision tree is better than others below for death target f1 score too low.

Bagging

Fourth model we trained was bagging. Before training has been started, we fine tuned a bagging model with grid search with 5 fold cross validation.

Death Target

Optimized parameters base estimator for logistic regression and decision tree and bootstrap features for true and false and n estimators for 10, 20, 30. As a result of grid search we obtain those results:

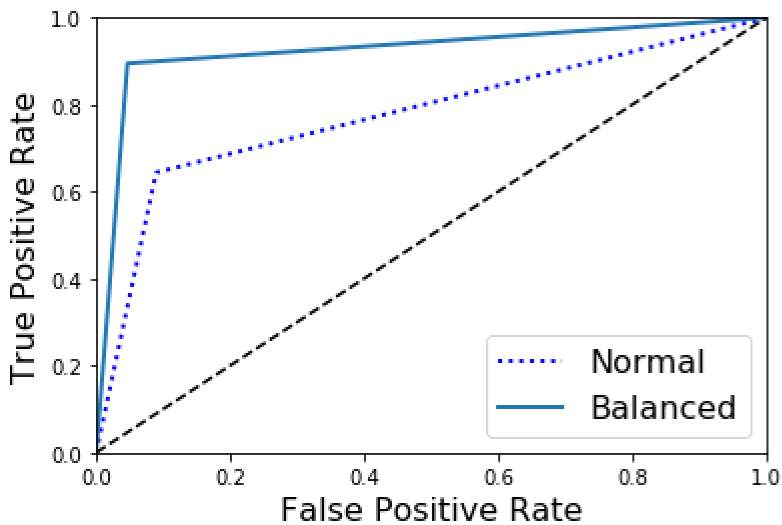
Best training accuracy: 0.9152328647613797

Best parameters: {'base_estimator': DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',

max_depth=None, max_features=None, max_leaf_nodes=None,
min_impurity_decrease=0.0, min_impurity_split=None,
min_samples_leaf=1, min_samples_split=2,
min_weight_fraction_leaf=0.0, presort='deprecated',
random_state=0, splitter='best'), 'bootstrap_features': True, 'n_estimators': 30}

Test results:
Test accuracy: 0.8567870485678705
F1 score: 0.6439628482972136
Test accuracy for balanced samples: 0.923804012345679
F1 score: 0.9222593977563472

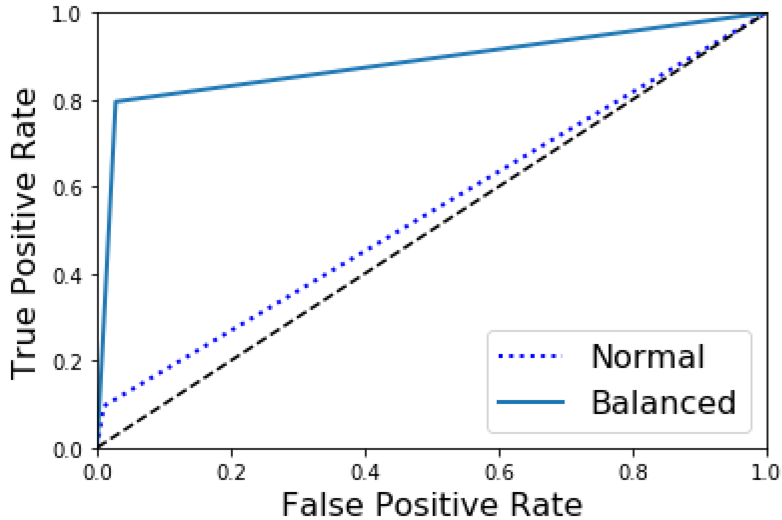
ROC Curve that includes both normal and balanced datasets



Readmitted Target
Optimized parameters base estimator for logistic regression and decision tree and bootstrap features for true and false and n estimators for 10, 20, 30. As a result of grid search we obtain those results:
Best training accuracy: 0.8503152371965654
Best parameters: {'base_estimator': DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
max_depth=None, max_features=None, max_leaf_nodes=None,
min_impurity_decrease=0.0, min_impurity_split=None,
min_samples_leaf=1, min_samples_split=2,
min_weight_fraction_leaf=0.0, presort='deprecated',
random_state=0, splitter='best'), 'bootstrap_features': True, 'n_estimators': 30}

Test results:
Test accuracy: 0.8468244084682441
F1 score: 0.16610169491525423
Test accuracy for balanced samples: 0.8840230098348487
F1 score: 0.8726309353984104

ROC Curve that includes both normal and balanced datasets



Random Forest

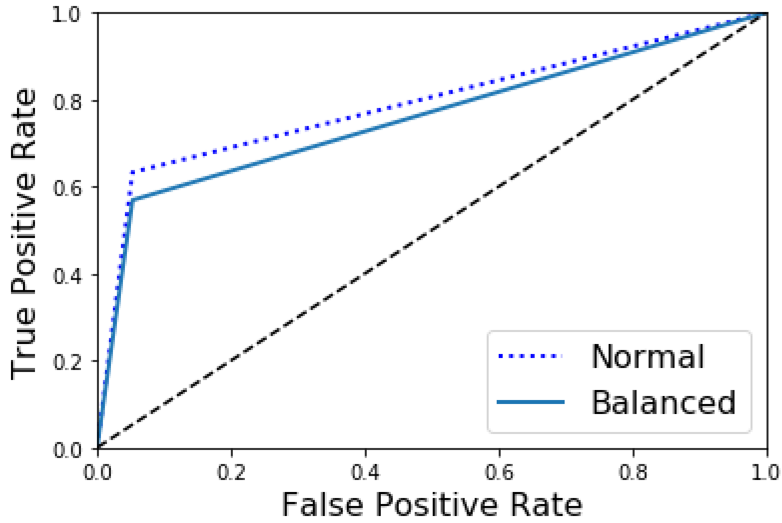
Fifth model we trained was random forest. Before training has been started, we fine tuned a random forest model with grid search with 5 fold cross validation.

Death Target

Optimized parameters n estimators for 10, 20, 30 max depth for None, 1, 5, 7 and max features for 10, 20, 30 and max leaf nodes for 2, 4 and min samples split for 2, 4. As a result of grid search we obtain those results:
Best training accuracy: 0.899976917239823
Best parameters: {'max_depth': None, 'max_features': 30, 'max_leaf_nodes': 4, 'min_samples_split': 2, 'n_estimators': 20}

Test results:
Test accuracy: 0.8838729763387297
F1 score: 0.686817800167926
Test accuracy for balanced samples: 0.7561728395061729
F1 score: 0.7021677662582471

ROC Curve that includes both normal and balanced datasets



Readmitted Target

Optimized parameters n estimators for 10, 20, 30 max depth for None, 1, 5, 7 and max features for 10, 20, 30 and max leaf nodes for 2, 4 and min samples split for 2, 4. As a result of grid search we obtain those results:

Best training accuracy: 0.8473573723215895

Best parameters: {'max_depth': None, 'max_features': 10, 'max_leaf_nodes': 2, 'min_samples_split': 2, 'n_estimators': 10}

Test results:

Test Accuracy: 0.839975093399751

F1 score: 0.0

XGBoost

Sixth model we trained was xgboost. Before training has been started, we fine tuned a xgboost model with grid search with 5 fold cross validation.

Death Target

Optimized parameters eta for 0.3 and max depth for 3, 6, 9 and objective for multi:softprob and num_class for 3, 6 and steps for 20. As a result of grid search we obtain those results:

Best training accuracy: 0.918657716291067

Best parameters: {'eta': 0.3, 'max_depth': 9, 'num_class': 3, 'objective': 'multi:softprob', 'steps': 20}

Test results:

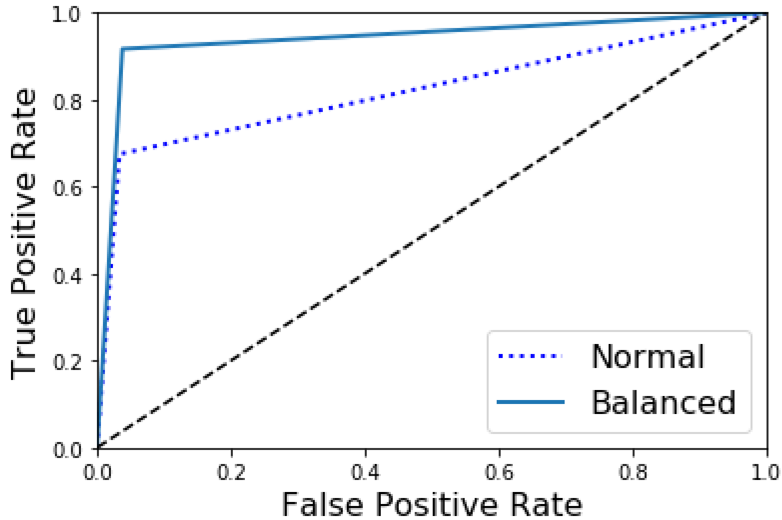
Test accuracy: 0.9078455790784558

F1 scores: 0.7465753424657534

Test accuracy for balanced samples: 0.9392361111111112

F1 scores: 0.93841642228739

ROC Curve that includes both normal and balanced datasets

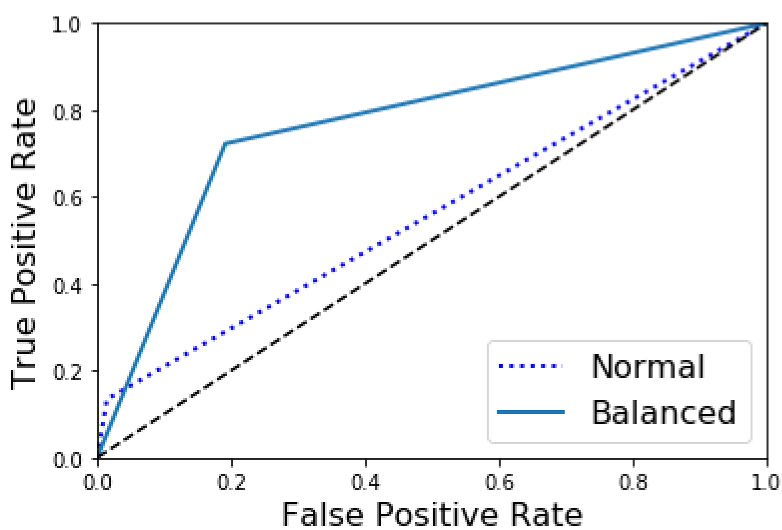


Readmitted Target

Optimized parameters eta for 0.3 and max depth for 3, 6, 9 and objective for multi:softprob and num_class for 3, 6 and steps for 20. As a result of grid search we obtain those results:
Best training accuracy: 0.8530395784518495
Best parameters: {'eta': 0.3, 'max_depth': 6, 'num_class': 3, 'objective': 'multi:softprob', 'steps': 20}

Test results:
Test accuracy: 0.8493150684931506
F1 score: 0.22186495176848872
Test accuracy for balanced samples: 0.7656336982742624
F1 scores: 0.7548048922539313

ROC Curve that includes both normal and balanced datasets



LDA

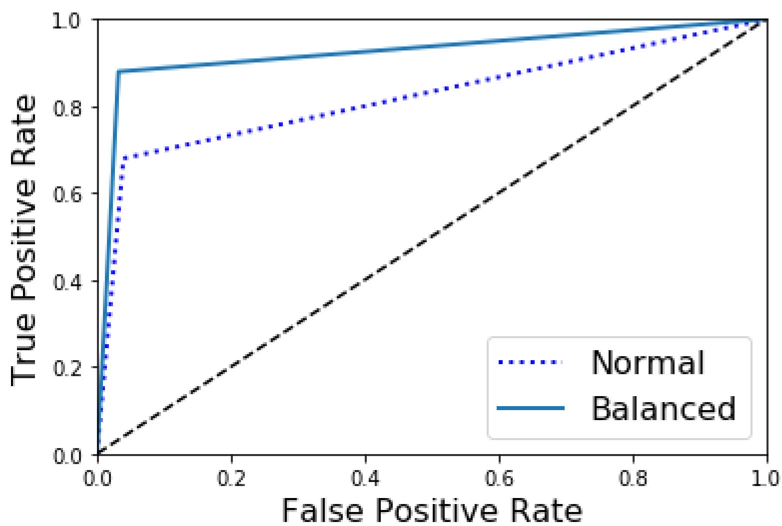
Seventh model we trained was lda. Before training has been started, we fine tuned a ldamodel with grid search with 5 fold cross validation.

Death Target

Optimized parameters n components for None, 1, 3, 5 and tol for 0.0001, 0.001, 0.01. As a result of grid search we obtain those results:
Best training accuracy: 0.9145322030252956
Best parameters: {'n_components': None, 'tol': 0.0001}

Test results:
Test accuracy: 0.9041095890410958
F1 score: 0.7403035413153457
Test accuracy for balanced samples: 0.9232253086419753
F1 score: 0.9204318272690925

ROC Curve that includes both normal and balanced datasets

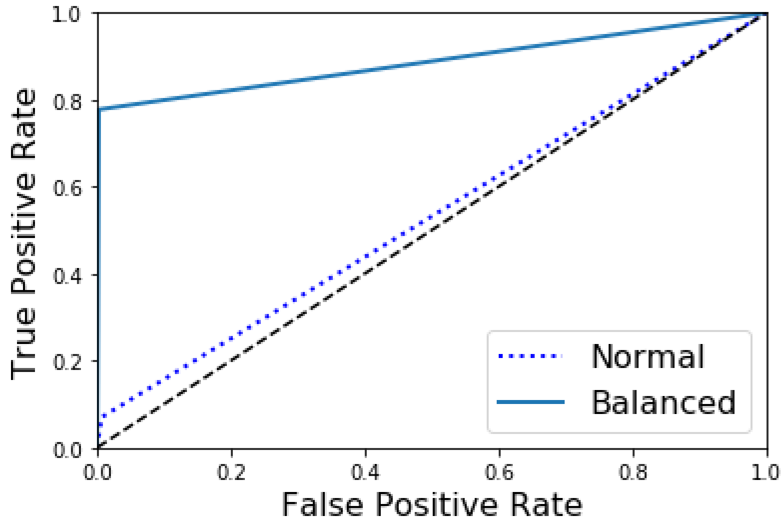


Readmitted Target

Optimized parameters n components for None, 1, 3, 5 and tol for 0.0001, 0.001, 0.01. As a result of grid search we obtain those results:
Best training accuracy: 0.8492257733254774
Best parameters: {'n_components': None, 'tol': 0.0001}

Test results:
Test accuracy: 0.8452677459526775
F1 score: 0.1265377855887522
Test accuracy for balanced samples: 0.8875487103358694
F1 score: 0.8734864300626304

ROC Curve that includes both normal and balanced datasets



5. Conclusion

Generally for all models training and test accuracies are high enough. However, accuracy could be misleading since actual dataset is highly unbalanced. That's why we need some other performance metrics like f1 scores, roc, auc. It can be clearly seen f1 and roc scores are not even close to accuracies for models and even they are 0 for some models. However, some of the models are really promising with oversampled data and even with actual data.

For death target top three models are:

XGBoost(acc: 0.90/ f1: 0.74 - acc: 0.93/ f1: 0.93)
LDA(acc: 0.90/ f1: 0.74 - acc: 0.92/ f1: 0.92)
Bagging(acc: 0.85/ f1: 0.64 - acc: 0.92/ f1: 0.92)

For readmitted target top three models are:

Bagging(acc: 0.84/ f1: 0.16 - acc: 0.88/ f1: 0.87)
LDA(acc: 0.84/ f1: 0.12 - acc: 0.88/ f1: 0.87)
XGBoost(acc: 0.84/ f1: 0.22 - acc: 0.76/ f1: 0.75)