Diabetes 130 US Hospitals for Years 1999-2008: a Machine Learning Approach to Predict Readmission

Introduction

In this project, we aim to improve the prediction of whether a patient will be readmitted to the hospital within 30 days or not by using a diabetes dataset. Previously the diabetes readmission problem was partially investigated by simple mean comparisons (Ostiling et al., 2017) or multivariate logistic regression (Strack et al., 2014), which had a hidden assumption of no multicollinearity among the independent variables, and therefore requires rigorous feature selections. For the current project, we incorporated 10 machine learning models following feature exploration and data preprocessing. After hyper tuning, we were able better to predict the possibility of a 30-day readmission rate and to determine the driving features. Critically, we systematically compared undersampling versus undersampling methods to address the problem of an imbalanced class. Our results suggest 1) Advanced algorithms performed better than the simple method in predicting the 30-day diabetes readmission rates. Among them Cat Boosting Classifier had the best scores, outperforming the existing models using a similar approach. 2) Oversampling and undersampling do not produce the same effects for all models. Our results show that oversampling improves the performance of Gradient Boosting Classifier, but is harmful to tree-based models like Random Forest. 3) Hypertuning improves the performances of all learners except for the fully-connected neural network. 4) The novel features we engineered became the dominant features for prediction. These results suggest that advanced machine learning classifiers, relative to simple linear models, are more useful for predicting the 30-day readmission rates in diabetic patients based on their profiles. Importantly, we should use proper feature engineering and preprocessing skills, as well as appropriate sampling methods for different models.

Problem Statement

Several challenges are worth mentioning. First, some patients occurred more than once in the dataset, leading to a concern about the i.i.d. assumption. We quantitatively tried to address why we decided to keep the repeats. Second, some features contain many missing values, so we evaluated each of them to decide whether we would create a new category or drop them. Some important features contain up to hundreds of distinct values, and to avoid too many binary features when using one-hot

encoding, we either used domain knowledge to reduce the categories or engineered some features that would incorporate more information. Finally, we tried to tackle the issues regarding the imbalance between the two classes. Besides different sampling methods, it is also critical to find good metrics for model evaluation other than accuracy. As we have more information on the negative group and more negative group samples in the testing set, it is likely that the models would produce lots of negative predictions and few positive predictions, so high accuracy doesn't indicate the model is learning. Therefore, we chose the AUC score to capture the trade-off between specificity and sensitivity. However, it is still possible that the AUC is high due to low positive predictions in general, so we further included the Harmonic Mean F1 score to get a measure of the trade-off between precision and recall, where precision is calculated by the F1 score of the probability of being in the negative group. The recall is calculated by the probability of being in the positive group.

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Methods

We first did feature exploration to remove redundant features and deal with missing data(**Table 1**). More than 75% of the patients have only one encounter record (**Figure 1**). We decided not to remove the repeated records, as we are unsure how the repeated visits were sampled. The feature "Medical_specialty" has 50% of the missing data, but we did not remove it because this feature is very important in predicting the readmission rate. We removed non-related and redundant features with too much missing data (**Table 2**).

We then did feature engineerings such as integer encoding and one-hot encoding. First, we regrouped some features with too many distinct values into several categories, such as the primary, secondary, and third diagnoses (Strack et al., 2014). Second, we introduced a novel method of reducing the 3x9 one-hot features by creating only the binary feature per disease group and encoding the values based on whether this disease has occurred in any of the diagnoses (**Table 3**). Finally, we split our data into a training set (70%), a validation set (15%), and a test set (15%), followed by rescaling some numerical features and combining them with binary features. We also preprocessed the dataset before training the models. As the target class is unbalanced (about 15% of our training data was readmitted within 30 days), we thus experimented with both oversampling and undersampling to balance our dataset. We then trained 10 different machine learning algorithms with both oversampling and undersampling to preprocess the dataset and hyperparameter tuning to find the best hyper tuned model and dominant features to predict the 30-day readmission rate: Neural Network with "lbfgs" optimizer, Neural Network with "adam" in Pytorch, Stochastic Gradient Descent,

Random Forest, Gradient Boosting, AdaBoosted, CatBoosted, Naïve Bayes, Logistic Regression and Decision Tree.

Experiments

Experimental setup

We conducted experiments for feature selection, preprocessing, different learning algorithms, and hyperparameter tuning for each learning algorithm. Given the imbalanced dataset, we implemented both undersampling and oversampling for the training data using all the models. We have 88 features in total after one-hot encoding, feature selection, and feature engineering. After initial model exploration, hyperparameter tuning was conducted for the best six models (Stochastic Gradient Descent, Random Forest, Gradient Boosting, Adaboost Classifier, Cat boosting Classifier, and Neural Network with “lbfgs” optimizer) using their default setting.

Experimental results

We first compared undersampling and oversampling results using each model. We identified that the tree-based models, especially the random forest, performed very poorly in the validation data when using the oversampling method, and hyperparameter tuning couldn’t improve its performance. This is because oversampling the positive group data (readmitted <30 days) led to overfitting in the tree models (**left bottom panel in Figure 2**, f1 harmonics score closes to 1.0 in the training data but closes to 0.0 in the validation data). When using undersampling, the tree-based models showed tremendous improvement, increasing the h1 harmonics score to around 0.4 (**right bottom panel in Figure 2**). The undersampling did not indicate lower performances in other models by comparing the AUC and the harmonic f1 score with the oversampling, even though 90% of the data were dropped (**Figure 2**). The Cat Boosting classifier performed the best in the validation data, and the neural network using “lbfgs” optimizer, logistic regression, and Random Forest classifier fell closely behind (**right panel in Figure 2**).

We then conducted hyperparameter tuning for the six best models, as it was too computationally expensive to run all ten models. All of the models showed improvements after hyperparameter tuning, except for the Neural Network with “lbfgs” optimizer, where the scores stayed the same. The Cat Boosting Classifier performed the best among the four models (**Figure 3**). **Figure 4** shows the learning curve of the Cat Boosting Classifier, where the training score continued to decrease, and the validation score continued to increase with more samples. This indicated that the Cat Boosting Classifier is robust and did not encounter overfitting problems.

Figure 5 shows the features sorted by dominance used to predict the 30-day readmission rate in the Cat Boosting Classifier. The top five dominant features are the

number of inpatients, discharge disposition id, discharge disposition id, age, and number of diagnostics, which provided more information compared to previous studies without considering all of the features (Strack et al., 2014). Finally, we evaluated the final model fits among the best four hyper-tuned models Catboosting, Gradient Boosting Classifier, Random Forest, and Neural Network with “lbfgs” optimizer using the test datasets (**Figure 6**). The ROC curve and the performance score (i.e., AUC and f1 harmonics score) confirmed that the Cat Boosting Classifier performed best in the testing sets, even though these four models did not show very obvious differences.

Discussion

In this project, our best performance model trained to predict the patient's 30-day readmission rate was the Cat Boosting Classifier. We were also able to determine the driving features used to achieve the performance. We can conclude that feature exploration and data preprocessing are very important before applying machine learning algorithms. However, we did not compare our model with a reduced model with fewer essential features, as we included all the features we considered to be important. In future work, we could conduct more experiments after feature selection, such as re-train our model and comparing the results using the group with and without the feature “medical_specialty”, which has over 50% missing value. Moreover, as we already identified the value of novel feature engineering on the 9 disease groups, we should compare models using the raw features (27 one-hot features) versus models only using the 9 features. If the results are promising, it could shed new light on clinical practice, as well as healthcare data mining.

References

Strack B, DeShazo J, et al. Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records. *BioMed Research International.* 2014. doi: 10.1155/2014/781670

Ostling S, Wyckoff J, Ciarkowski SL, et al. The relationship between diabetes mellitus and 30-day readmission rates. *Clin Diabetes Endocrinol*. 2017;3:3. Published 2017 Mar 22. doi:10.1186/s40842-016-0040-x

Tables:

Table 1: Missing Values

Feature	Action	Feature	Action
Weight, Payer_code	Remove whole features (96%, 40% missing)	Race	Group Nan into a new category (2%)
Diag_1, Diag_2, Diag_3	Remove Rows with Nan (Less than 1%)	Medical_Speciality	Group Nan into a new category (50%)

Table2: Remove Uninformative Features¶

Feature	Discarding Reasons
Encounter_id, Patient_nbr	Not Related to our target (readmitted)
Glipizide-metformin, Glimepiride-pioglitazone, Metformin-rosiglitazone, Metformin-pioglitazone, Acetohexamide, Tolbutamide, Troglitazone, Tolazamide	Few sample (less than 20 patients) use these drugs
Repaglinide,Nateglinide, Chlorpropamide, Acarbose, Miglitol, Glyburide-metformin	Over 98% percent of patients didn't use these drugs

Table3: Feature Engineering

Feature	Changes	Feature	Changes
Gender	Remove 3 rows with Invalid gender, convert to 1/0 for male and female	Age	Convert each range to the average (like [50,60) to 55)
Race	One-hot Encoding	Change, DiabetesMed	0/1 Encoding
Max_glu_serum	'None' => 0, 'Norm' => 1, '>200' => 2, '>300' => 3	A1Cresult	'None' => 0, 'Norm' => 1, '>7' => 2, '>8' => 3
Metformin, Glimepiride, Glipizide, Glyburide, Pioglitazone, Rosiglitazone, Insulin	'No' => 0, 'Down' => 1, 'Steady' => 2, 'Up' => 3	Diag_1, Diag_2, Diag_3	Group 800 distinct values to 9 diagnosis and then One-hot Encoding (Circulatory, Respiratory, Digestive, Diabetes, Injury, Genitourinary, Neoplasms, Musculoskeletal, Others
Ever diagnosed	Create a new feature to check whether a patient is diagnosed with any specific diagnosis during diag_1, diag_2 or diag_3	Discharge_disposition_id	Removing all encounters discharge to a hospice or patient death. Decreasing distinct values from 28 to 8 by grouping similar type, then one-hot encoding
Admission_type_id	Just pick 3 more frequent types and group others. Then one-hot encoding	Admission_source_id	Decreasing distinct values from 17 to 6 by grouping similar type, then one-hot
Readmitted	'No' & '>30' => 0, '<30' => 1	Medical_specialty	Reduce 20 distinct values to 7 by grouping low frequency values, then one-hot encoding

Figures

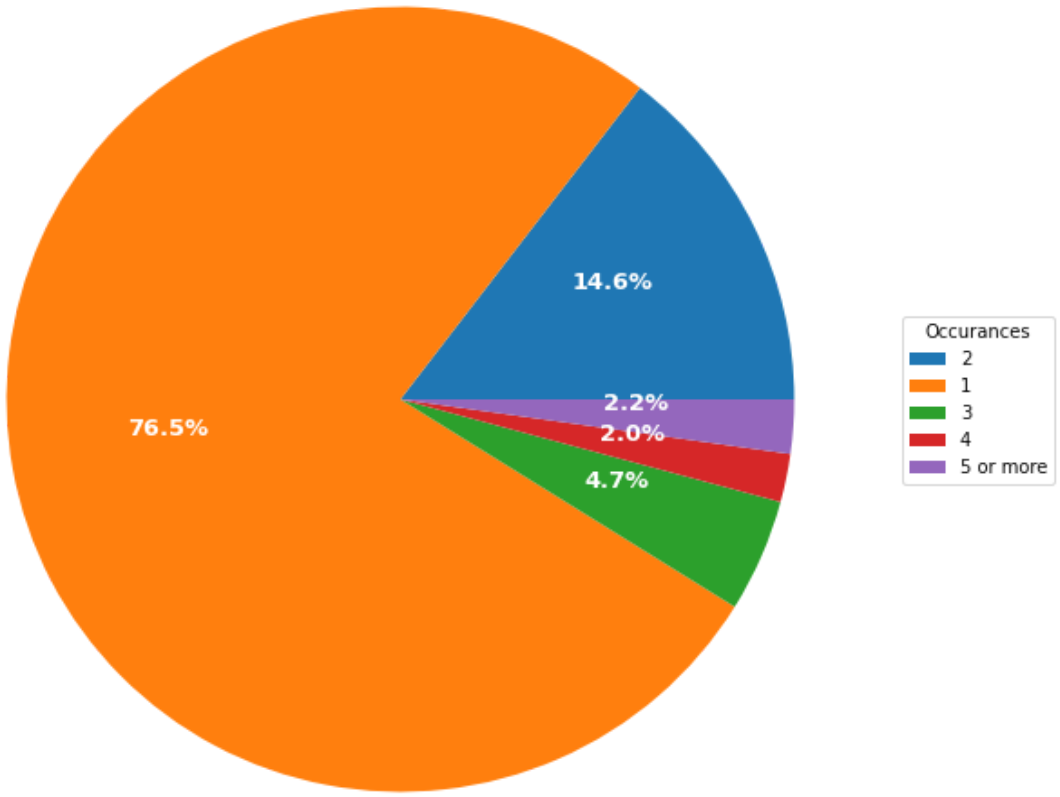

Figure 1. The Percentage of Different Numbers of Occurance pere Patient ID

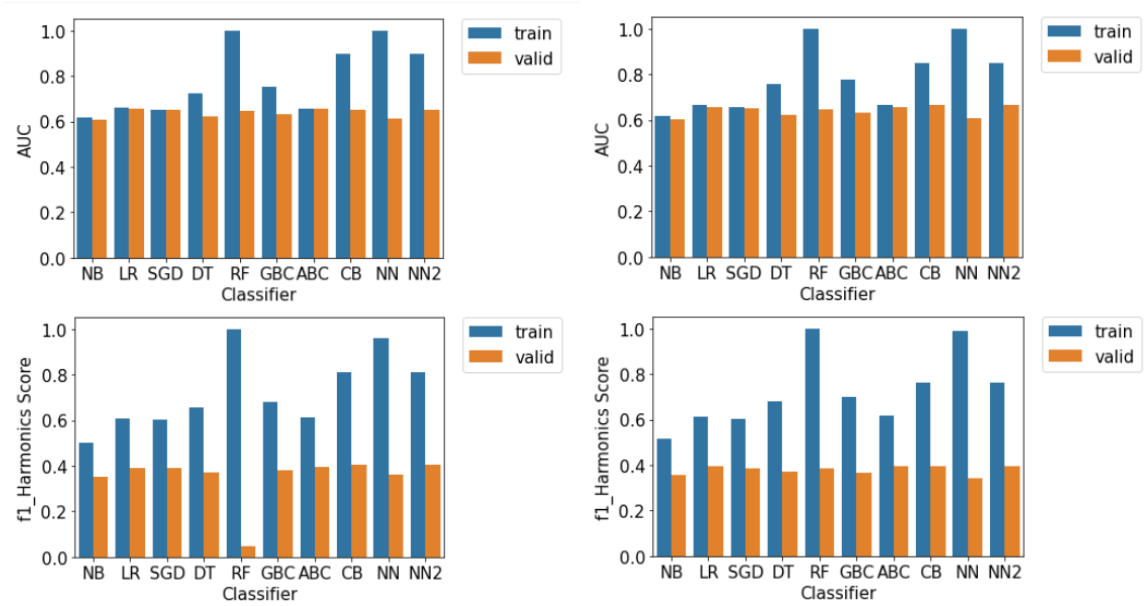

Figure 2. The AUC (top panel) and f1 harmonics score (bottom panel) for each model in the oversampling (left panel) and undersampling (right panel). The blue and orange bar represents the score in the training and validation dataset. NB, LR, SGD, DT, RF, GBC, ABC, CB, NN and NN2 represents Naive Bayes, Logistic Regression, Stochastic Gradient Descent, Decision Tree, Random Forest, Gradient Boosting Classifier, Catboosting Classifier, Neural Network with "adam" in Pytorch and Neural Network with "lbfgs" optimizer, respectively.

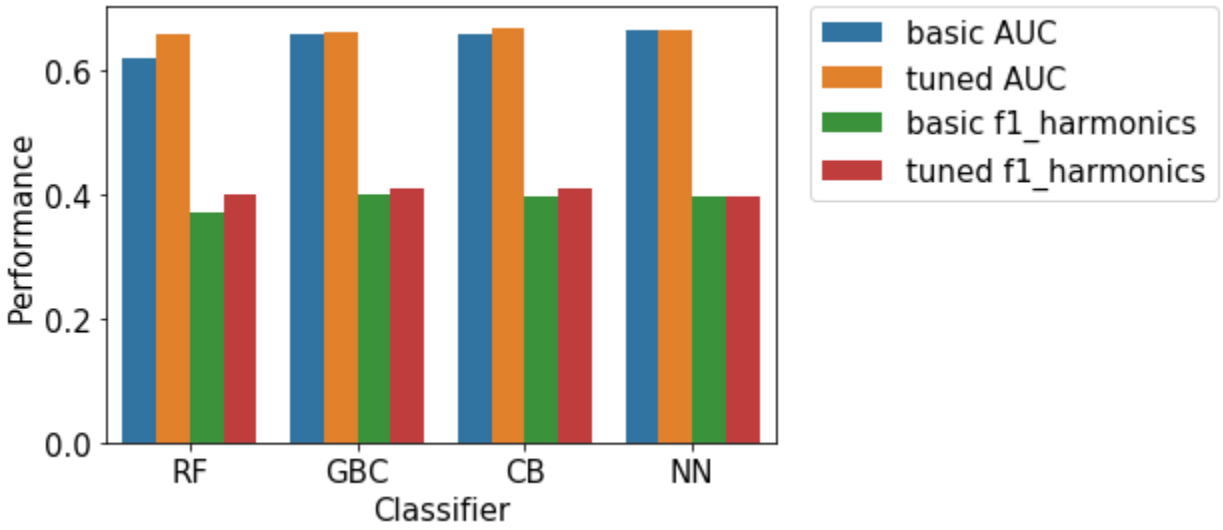

Figure 3. The AUC and f1 harmonics scores in Random Forest, Gradient Boosting Classifier, Catboosting Classifier and Neural Network using "lbfgs" optimizer before and after hyperparameter tuning. The blue, orange, green and red bar represents basic AUC, tuned AUC, basic f1 harmonics and tuned f1 harmonics of the four models, respectively.

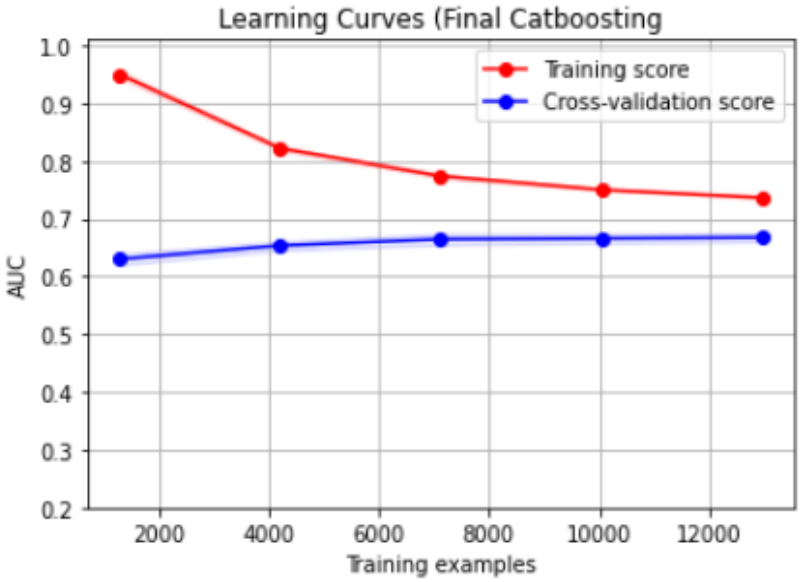

Figure 4. The learning curve of tuned Catboosting classifier. The red and blue line represents the training and cross-validation AUC score with training examples, respectively.

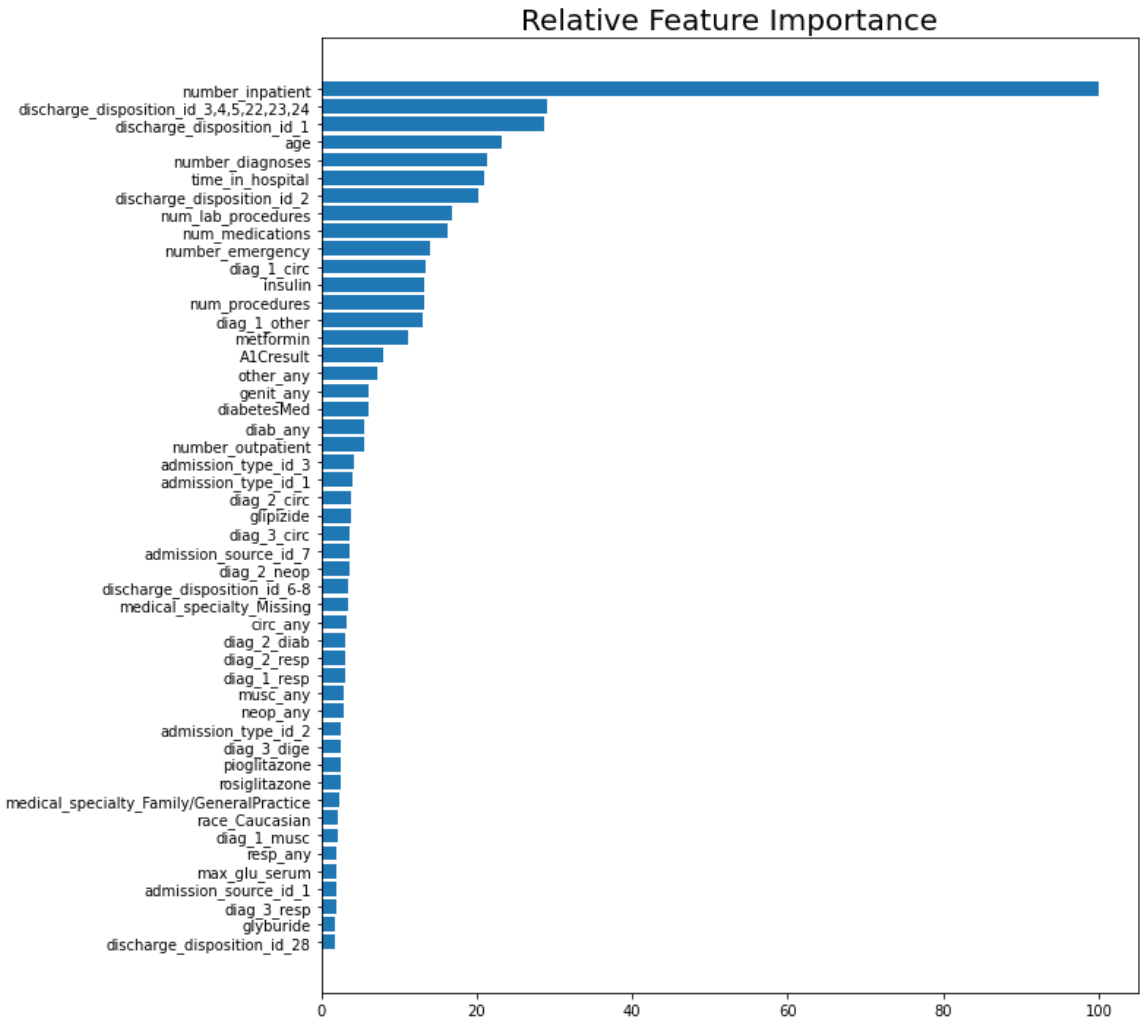

Figure 5. The relative feature importance compared to the most important feature (unit: %).

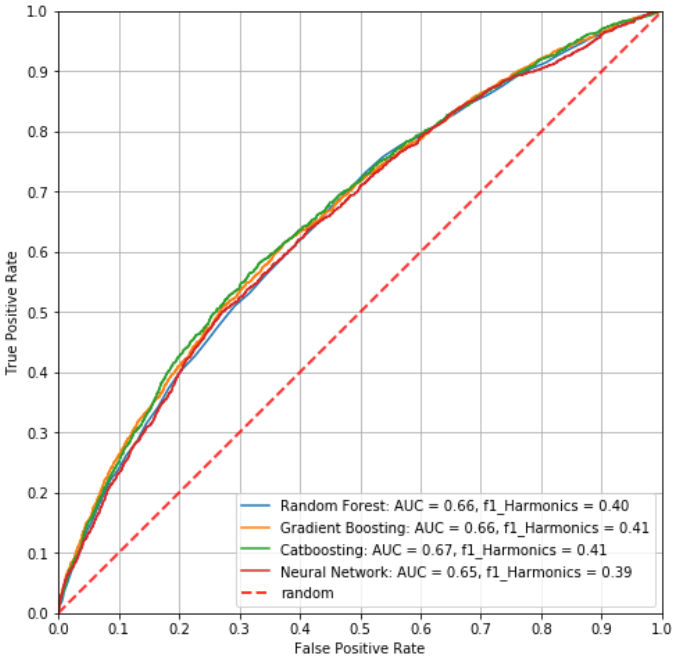

Figure 6. The blue, orange, green and red solid line represents the ROC for tuned Random Forest, Gradient Boosting Classifier, Catboosting Classifier and Neural Network, respectively. The AUC and f1 harmonics score is computed from testing dataset.