Bangladesh University of Engineering and Technology

CSE 472: Machine Learning Sessional

Name: Faria Binta Awal

Student Id: 1905012

Level: 4

Term: II

Section: A

Date: 20 September, 2024

Task Overview:

In this offline, we have to implement a Logistic Regression Model along with two Ensemble Learning methods – Bagging and Stacking. We are given 3 datasets with various impurities from Kaggle and UCI to run our implemented model on those datasets after applying necessary preprocessing steps which already have been done in Offline-1. We have to measure different performance metrics to get an overview of how the model worked well on the datasets.

Pipeline Overview:

As it has been said to modularize the code, several code snippets which occur frequently at different steps for all the 3 datasets have been written in functions. The first cells in the 1905012.ipynb file contain those functions.

After that, preprocessing steps have been done for all the 3 models and data prepared as input for the models have also been stored in a csv file for further use.

The features have been extracted and written in Dataset_X_1.csv, Dataset_X_2.csv and Dataset_X_3.csv respectively. Similarly, the labels are also extracted and written in Dataset_Y_1.csv, Dataset_Y_2.csv and Dataset_Y_3.csv respectively.

To get the performance metrics shown in the following tables, just press the 'Run All' option after opening the 1905012.ipynb Jupyter notebook in VsCode.

Logistic Regression:

The function 'logistic_regression' basically takes inputs as training set and validation set. We can define various parameters to run the model which do not get updated during the update steps. These parameters are called Hyperparameters.

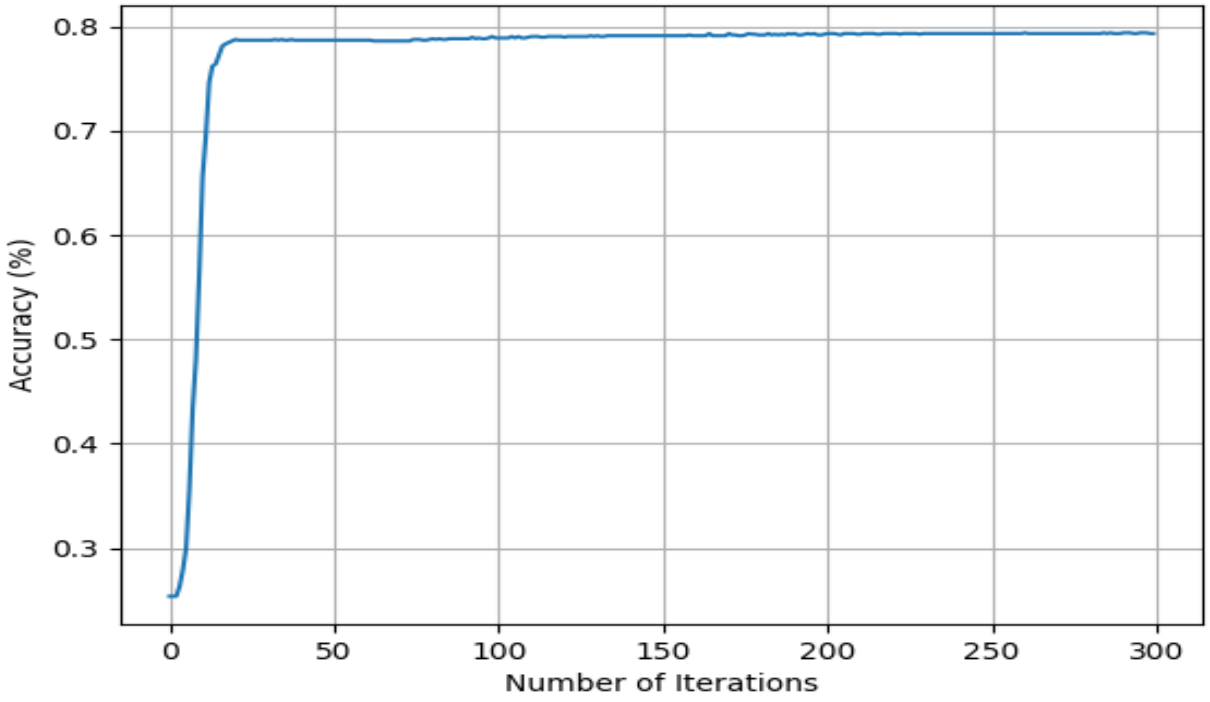
Possible hyperparameters are:

1. Alpha (Learning Rate)

2. Batch Size (for Minibatch Stochastic Process)

3. Number of Iterations

4. Threshold Value (The cutoff value of probability where we take decisions for prediction as positive or negative)



To tune the hyperparameters, cross-validation method is applied holding out 20% data from the training set as validation set. Due to time limitations, only two hyperparameters have been optimized to minimize the loss between true labels and predicted probabilities of validation set. These two important hyperparameters are – alpha and number of iterations.

Functions Implemented:

1. **hypothesis**: sigmoid function
2. **labeling**: to label encode and one-hot encode
3. **scaling**: to min-max and standard scaling of data
4. **get_accuracy_results**: to get the predictions, binary predictions and accuracy
5. **get_loss**: to calculate L2 loss for cross-validation

6. **logistic_regression**: base function which implements LR

7. **cross_validation**: to cross validate hyperparameters alpha and iterations.

8. **dataset_splitting**: it splits data into two disjoint subsets based on seed and ratio

9. **get_parameters**: it returns different performance parameters

10. **get_mean_values**: to calculate mean values of metrics for Bagging

11. **get_stdev_values**: to calculate standard deviation values of metrics for Bagging

12. **plot_violins**: for plotting violin plots to get the idea of distributions of metrics

Performance Metrics:

Dataset-1 (Telco Customer Churn)

	Accuracy	Sensitivity	Specificity	Precision	F1 Score	AUROC	AUPR
LR	0.801994	0.558989	0.884542	0.621875	0.588757	0.852153	0.642709
Bagging	0.812678	0.544715	0.908213	0.679054	0.604511	0.85031	0.677377
Mean	0.81157	0.537188	0.909393	0.67929	0.599507	0.848951	0.675765
STD	0.002418	0.024269	0.00789	0.010287	0.012522	0.003841	0.00643
Stacking	0.813522	0.523605	0.912281	0.67033	0.587952	0.844728	0.648318

Dataset-2 (Adult Income Dataset)

	Accuracy	Sensitivity	Specificity	Precision	F1 Score	AUROC	AUPR
LR	0.827321	0.499368	0.932819	0.705113	0.584668	0.871624	0.675534
Bagging	0.82365	0.50641	0.923709	0.676756	0.57932	0.871487	0.677026
Mean	0.824697	0.500522	0.926944	0.684883	0.577675	0.870432	0.674542
STD	0.002167	0.023107	0.009326	0.020515	0.009773	0.004071	0.008891
Stacking	0.834398	0.511832	0.937869	0.725466	0.600206	0.878997	0.700598

Dataset-3 (Credit Card Fraud Detection) - Full Dataset

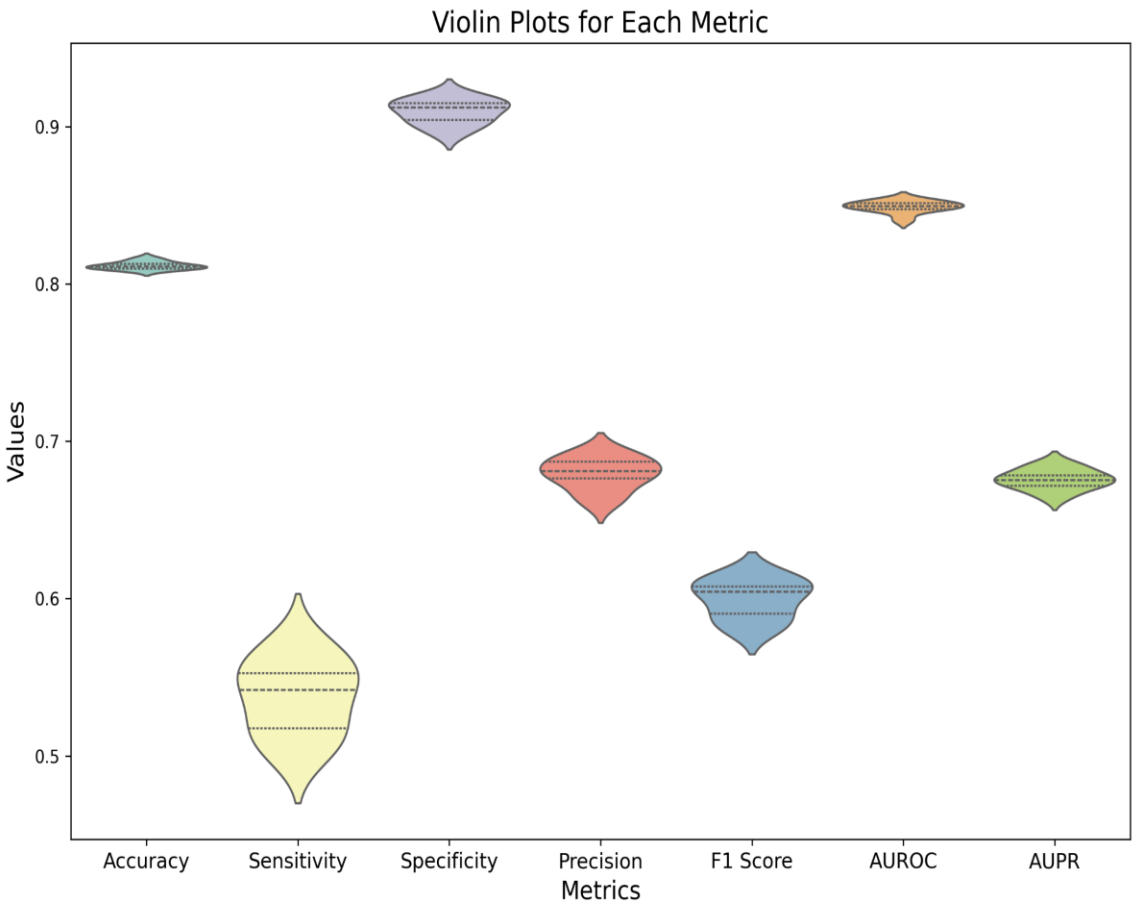
	Accuracy	Sensitivity	Specificity	Precision	F1 Score	AUROC	AUPR
LR	0.998202	0	1	0	0	0.935107	0.612974
Bagging	0.99852	0	1	0	0	0.958442	0.666749
Mean	0.99852	0	1	0	0	0.953551	0.657436
STD	0	0	0	0	0	0.014534	0.020549
Stacking	0.998376	0	1	0	0	0.928387	0.642943

Dataset-3 (Credit Card Fraud Detection) - Sampling 20K Dataset

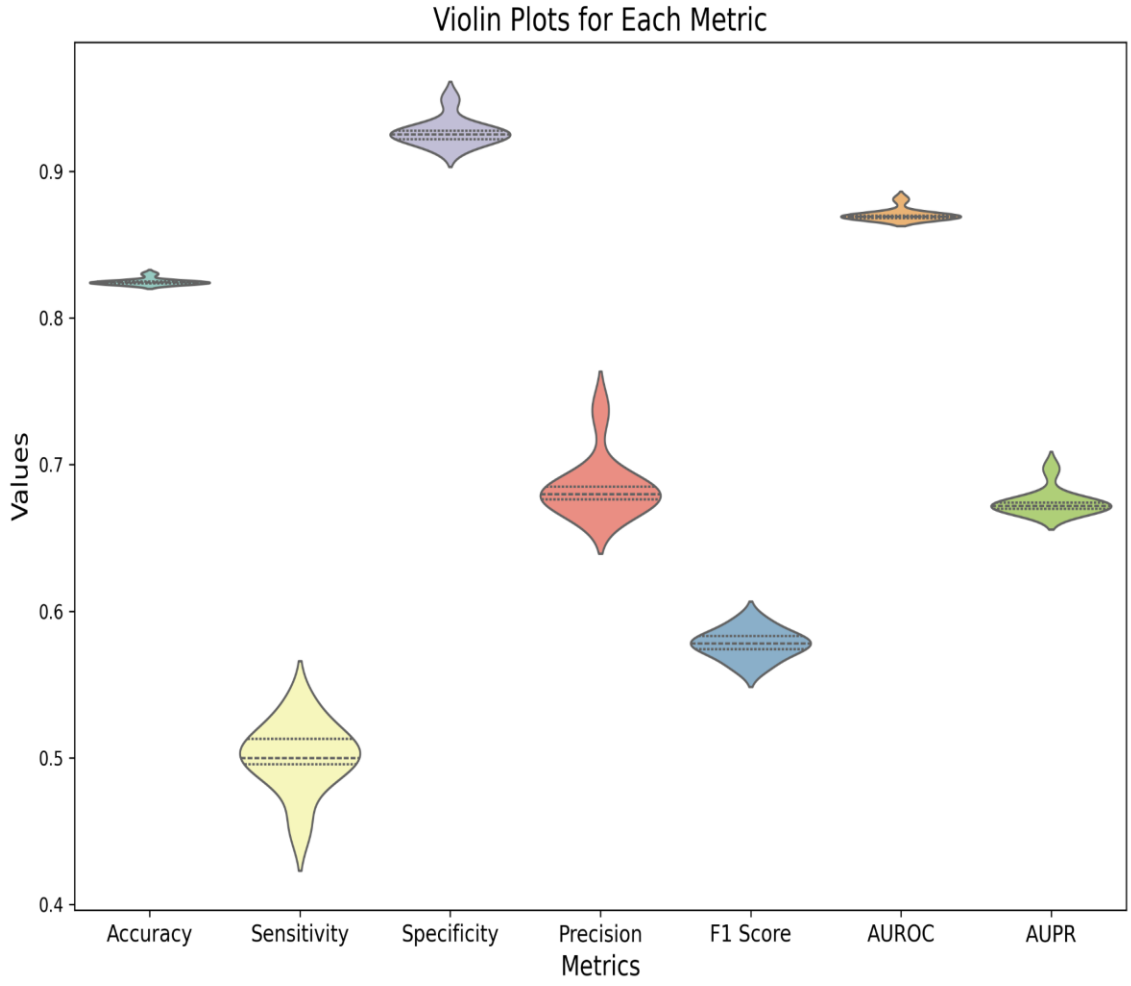
	Accuracy	Sensitivity	Specificity	Precision	F1 Score	AUROC	AUPR
LR	0.988598	0.53211	1	1	0.694611	0.971443	0.884554
Mean	0.988648	0.537207	0.999924	0.994565	0.697461	0.971024	0.882689
STD	0.000217	0.013064	0.000115	0.008153	0.00908	0.002165	0.004409
Stacking	0.991265	0.644068	0.999223	0.95	0.767677	0.977131	0.895534

Violin Plots:

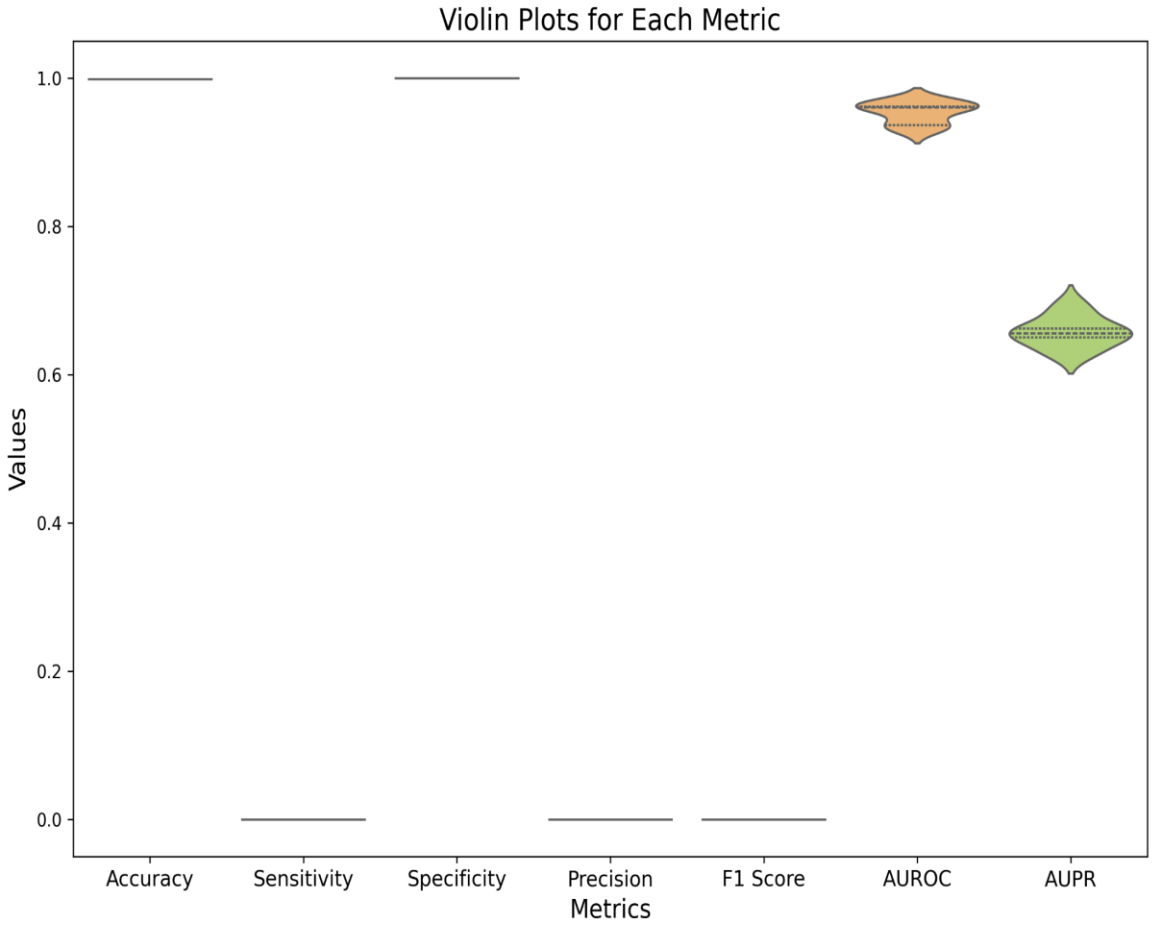
Dataset-1:



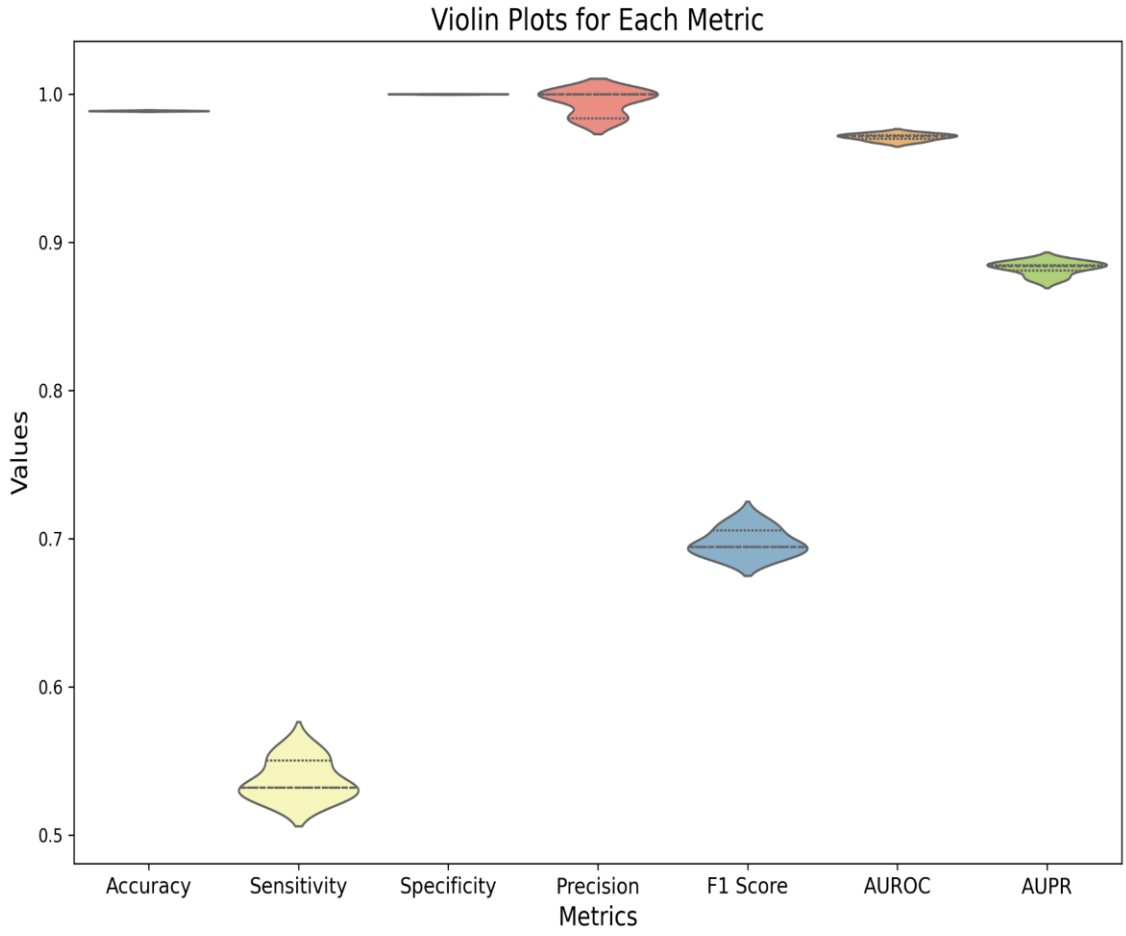
Dataset-2:


Violin Plots for Each Metric
Values
0.9
0.8
0.7
0.6
0.5
0.4
Accuracy
Sensitivity
Specificity
Precision
F1 Score
AUROC
AUPR
Metrics

Dataset-3: (Full dataset)



Dataset-3: (Sampled 20K)



Metrics Evaluation for Different Datasets:

1. Accuracy:

Accuracy = (True Positives + True Negatives)/ Total Instances

Accuracy measures the overall correctness of model. For dataset-1, it is around 81%. For dataset-2, it is around 83%. For dataset-3, it is around 99%. *As dataset-3 is highly imbalanced, it actually gives no significant insights from the metric.*

2. Sensitivity/Recall/True Positive Rate:

Recall = (True Positives)/ (True Positives + False Negatives)

Recall measures how well the model identifies positive instances. High sensitivity means model is good at identifying positive classes but it doesn't account for false positives. For dataset-1, it is around 54%. For dataset-2, it is around 50%. For dataset-3, it is 0%. As dataset-3 is highly imbalanced and positive class data are very rare, the model can't predict any positive classes. However, we get 53% recall after sampling 20K samples and it improves to 64% after stacking. So, after sampling and applying ensemble learning (stacking), the model's predictive power towards positive classes increased.

3. Specificity/True Negative Rate:

Specificity = (True Negatives) / (True Negatives + False Positives)

Specificity measures how well the model identifies negative instances. High specificity means model is good at identifying negative classes but it may miss positive instances. For dataset-1, it is around 90%. For dataset-2, it is around 93%. For dataset-3, it is 100%. As dataset-3 is highly imbalanced and positive class data are very rare, the model is biased towards negative classes. However, we get 99.99% specificity after sampling 20K samples and it reaches to 99.92% after stacking. So, after sampling and applying ensemble learning (stacking), the model's predictive power towards negative classes slightly decreased.

4.Precision:

Precision= (True Positives) / (True Positives+ False Positives)

Precision measures the accuracy of positive predictions. Precision is important when false positives are costly. If precision is low, then model predicts too many positive classes which are actually wrong. For dataset-1, it is around 68%. For dataset-2, it is around 69% but after stacking it improves to 72%. For dataset-3, it is 0%. As dataset-3 is highly imbalanced and positive class data are very rare, the model is biased towards negative classes and it may not predict positives classes at all. However, we get 95% precision after sampling 20K but still it is too much unbalanced and hence, we can't rely on the parameter.

5. F1-Score:

*F1_score = (2 * (Precision * Recall)) / (Precision + Recall)*

F1 score is high only when precision and recall is high as it penalizes extremes of either. For dataset-1, it is around 60%. For datset-2, it is around 58%. For dataset-3, f1 score is 0 as precision is 0. However, after sampling, we get precision of 69% for bagging and 76% for stacking. As we can't rely on precision too much for the highly imbalanced dataset, we can't rely on f1 score also.

6. AUROC (Area Under Receiving Operating Characteristic Curve):

AUROC = 1 means perfect classification.

AUROC = 0.5 means random guessing.

For dataset-1, AUROC is 85%. For dataset-2, we get AUROC 87% for bagging and it improves to 88% after stacking.

For the third dataset, AUROC is near 95% but it does not give much significance as dataset if highly imbalanced.

7. AUPR (Area Under Precision Recall Curve):

AUPR is particularly useful when positive class is rare (like detecting rare diseases or frauds). For the third dataset, it is 65% but after sampling 20K data, it raises to around 89%. This suggests that sampling procedure helped model to balance between precision and recall.