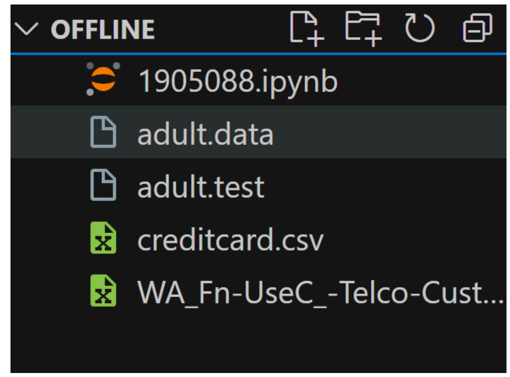
CSE472 (Machine Learning Sessional)

Assignment- 2: Logistic Regression with Bagging and Stacking

Mubasshira Musarrat
ID : 1905088

Process to run the code:



The .ipynb file along with all the unzipped csv files need to be kept in the same directory.

Inside the .ipynb file under ‘Workflow’ section, by commenting or uncommenting the lines one can get the results for different datasets. After selecting the desired dataset, ‘run all’ instances gives the necessary evaluations.

Work flow

```
X_train, X_test, Y_train, Y_test = telcecomInput()
# X_train, X_test, Y_train, Y_test = adultInput()
# X_train, X_test, Y_train, Y_test = creditCardInput()
```

[40] ✓ 57.0s Python

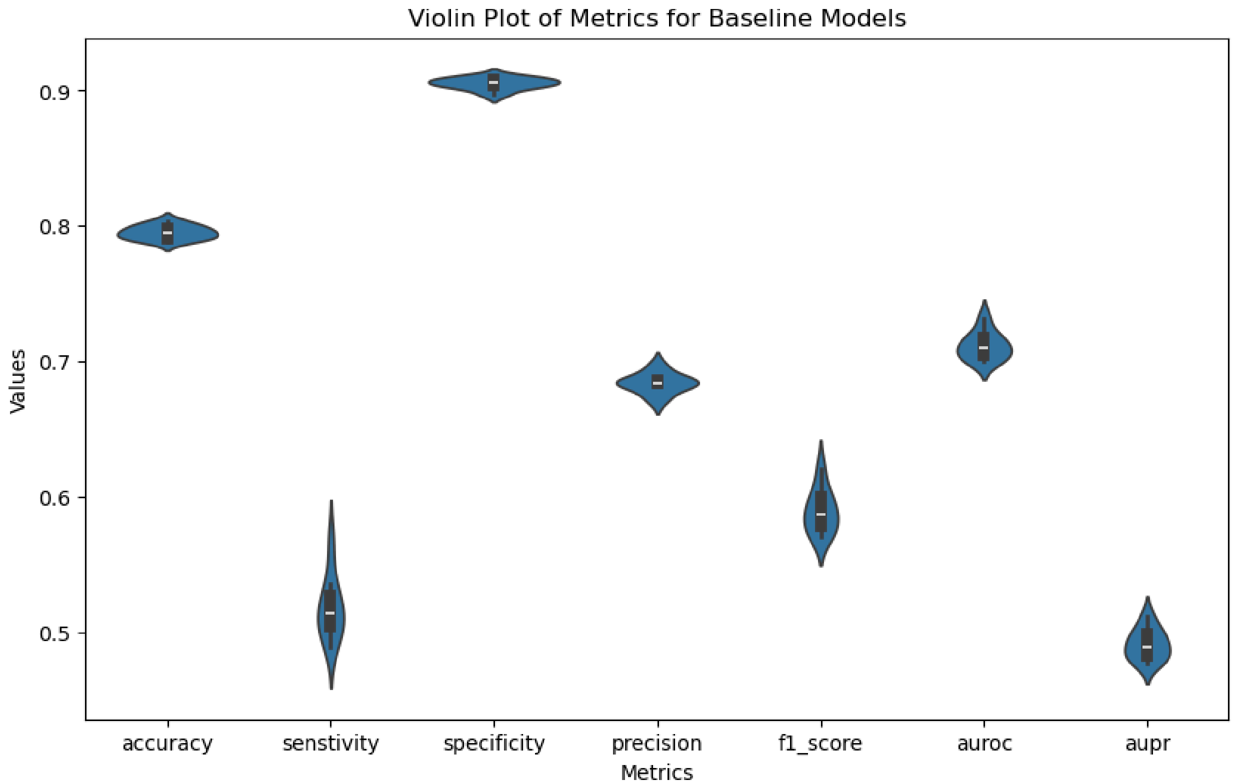
Performance Evaluation:

1. Teleco-customer-churn dataset

Performance on Test set

	Accuracy	Sensitivity	Specificity	Precision	F_1-score	AUROC	AUPR
LR	0.8032 ± 0.0024	0.5294 ± 0.0140	0.8947 ± 0.0056	0.6271 ± 0.0081	0.5739 ± 0.0072	0.712 ± 0.005	0.4498 ± 0.005
Voting ensemble	0.8071	0.5341	0.8984	0.6373	0.5811	0.7162	0.4571
Stacking ensemble	0.8036	0.5483	0.8889	0.6226	0.5831	0.7186	0.4545

The highest performance was observed for standard scaling, no feature selection, decaying learning rate with small initial learning rate & L2 regularization.

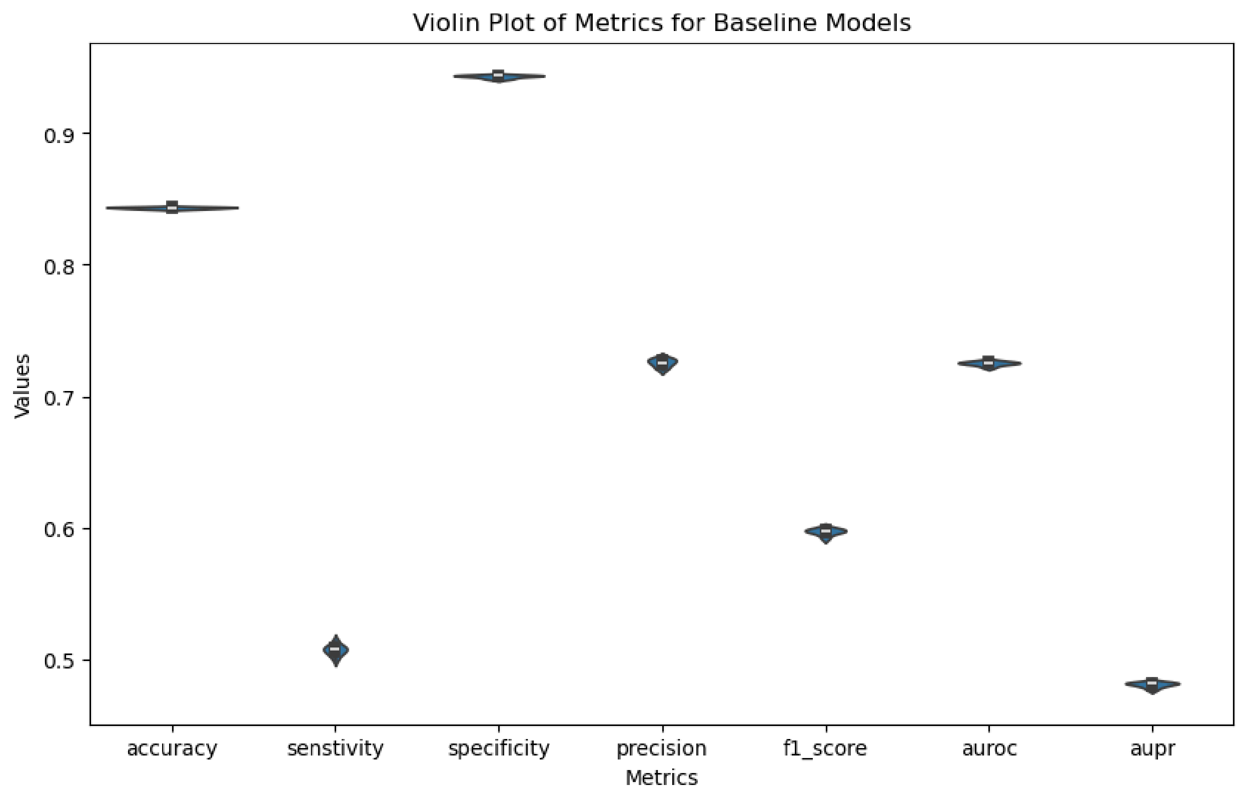


2. Adult dataset

Performance on Test set

	Accuracy	Sensitivity	Specificity	Precision	F_1-score	AUROC	AUPR
LR	0.8287 ± 0.0009	0.4908 ± 0.0032	0.9391 ± 0.0013	0.7246 ± 0.0037	0.5852 ± 0.0023	0.7150 ± 0.0014	0.481 ± 0.0021
Voting ensemble	0.8284	0.4913	0.9386	0.7230	0.5850	0.7149	0.4804
Stacking ensemble	0.8282	0.4900	0.9387	0.7230	0.5841	0.7143	0.4798

The highest prediction was observed for standard scaling, no feature selection, decaying learning rate with small initial learning rate & L2 regularization.

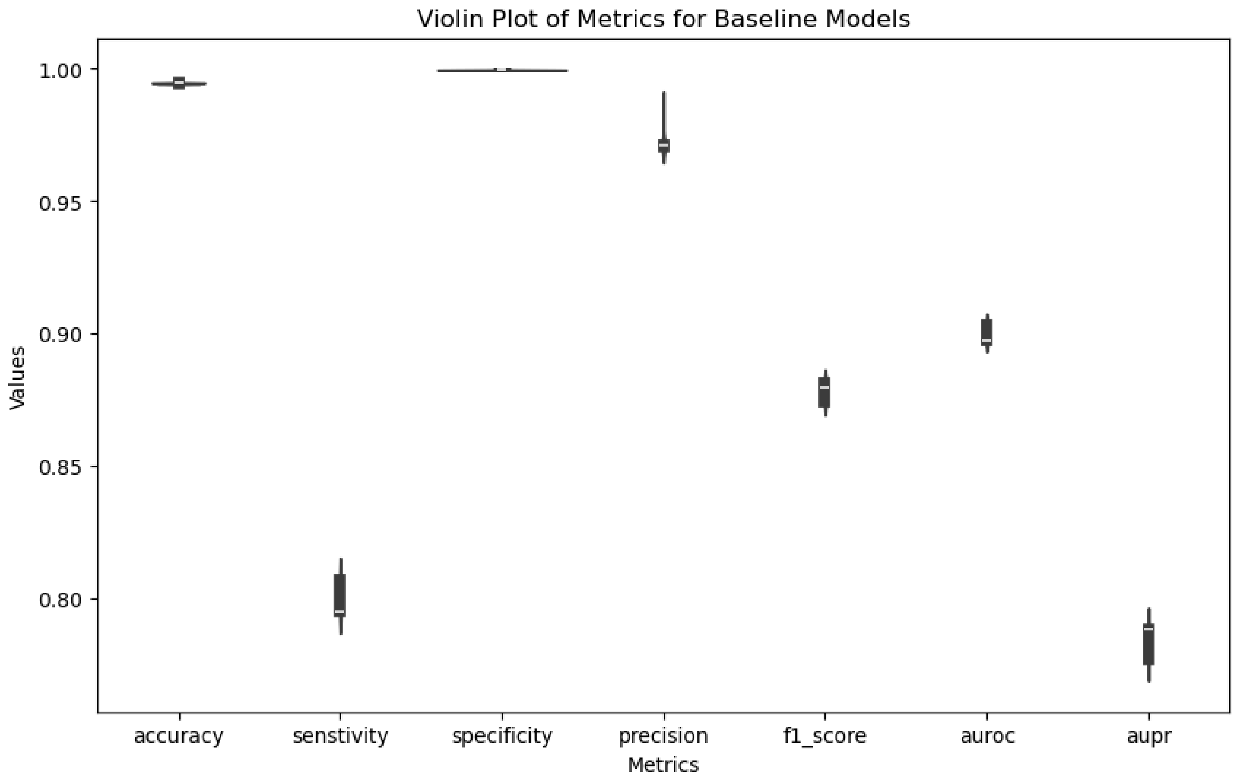


3. Credit-card-fraud dataset

Performance on Test set

	Accuracy	Sensitivity	Specificity	Precision	F_1-score	AUROC	AUPR
LR	0.9972 ± 0.0002	0.8852 ± 0.0055	0.9997 ± 0.0002	0.9851 ± 0.0074	0.9325 ± 0.0038	0.9425 ± 0.0027	0.8746 ± 0.0070
Voting ensemble	0.9973	0.8901	0.9998	0.9878	0.9364	0.9449	0.8817
Stacking ensemble	0.9971	0.8901	0.9995	0.9759	0.9310	0.9448	0.8711

The highest prediction is accrued by standard scaling, Feature selection, decaying learning rate with small initial learning rate & L2 regularization.



Observations:

1. Teleco-customer-churn dataset

Output with min-max scaling:

accuracy	0.7934 ± 0.0019	0.7936	0.7900
senstivity	0.5085 ± 0.0148	0.5057	0.5284
specificity	0.8886 ± 0.0032	0.8898	0.8775
precision	0.6040 ± 0.0031	0.6054	0.5905
f1_score	0.5521 ± 0.0091	0.5511	0.5577
auroc	0.6985 ± 0.0060	0.6978	0.7030
aupr	0.4303 ± 0.0059	0.4300	0.4302

Standard scaling increases accuracy more than min-max scaling.

accuracy	0.7954 ± 0.0030	0.7936	0.7943
senstivity	0.5243 ± 0.0184	0.5199	0.5483
specificity	0.8860 ± 0.0042	0.8851	0.8765
precision	0.6060 ± 0.0059	0.6020	0.5975
f1_score	0.5620 ± 0.0114	0.5579	0.5719
auroc	0.7052 ± 0.0077	0.7025	0.7124
aupr	0.4369 ± 0.0080	0.4332	0.4408

Training with all the columns increases accuracy.

accuracy	0.7974 ± 0.0035	0.7936	0.7979
senstivity	0.5246 ± 0.0187	0.5284	0.5682
specificity	0.8886 ± 0.0069	0.8822	0.8746
precision	0.6118 ± 0.0106	0.6000	0.6024
f1_score	0.5646 ± 0.0103	0.5619	0.5848
auroc	0.7066 ± 0.0071	0.7053	0.7214
aupr	0.4400 ± 0.0075	0.4352	0.4505

Large learning rate worsens the performance

accuracy	0.7949 ± 0.0042	0.7929	0.7964
senstivity	0.5205 ± 0.0208	0.5227	0.5455
specificity	0.8866 ± 0.0075	0.8832	0.8803
precision	0.6056 ± 0.0118	0.5993	0.6038
f1_score	0.5596 ± 0.0121	0.5584	0.5731
auroc	0.7035 ± 0.0082	0.7030	0.7129
aupr	0.4353 ± 0.0088	0.4329	0.4432

Decaying learning rate works better than constant learning rate

accuracy	0.8032 ± 0.0025	0.8071	0.8036
senstivity	0.5297 ± 0.0139	0.5341	0.5483
specificity	0.8947 ± 0.0056	0.8984	0.8889
precision	0.6273 ± 0.0083	0.6373	0.6226
f1_score	0.5742 ± 0.0072	0.5811	0.5831
auroc	0.7122 ± 0.0050	0.7162	0.7186
aupr	0.4500 ± 0.0051	0.4571	0.4545

L1 regularization worsens the performance:

accuracy	0.7975 ± 0.0036	0.8021	0.7943
senstivity	0.4318 ± 0.0232	0.4375	0.4659
specificity	0.9198 ± 0.0052	0.9240	0.9041
precision	0.6429 ± 0.0090	0.6581	0.6189
f1_score	0.5163 ± 0.0167	0.5256	0.5316
auroc	0.6758 ± 0.0097	0.6808	0.6850
aupr	0.4200 ± 0.0101	0.4289	0.4221

No regularization improves the performance than L1.

accuracy	0.8032 ± 0.0025	0.8071	0.8036
senstivity	0.5297 ± 0.0139	0.5341	0.5483
specificity	0.8947 ± 0.0056	0.8984	0.8889
precision	0.6273 ± 0.0083	0.6373	0.6226
f1_score	0.5742 ± 0.0072	0.5811	0.5831
auroc	0.7122 ± 0.0050	0.7162	0.7186
aupr	0.4500 ± 0.0051	0.4571	0.4545

L2 performs the best.

accuracy	0.8032 ± 0.0024	0.8071	0.8036
senstivity	0.5294 ± 0.0140	0.5341	0.5483
specificity	0.8947 ± 0.0056	0.8984	0.8889
precision	0.6271 ± 0.0081	0.6373	0.6226
f1_score	0.5739 ± 0.0072	0.5811	0.5831
auroc	0.7120 ± 0.0050	0.7162	0.7186
aupr	0.4498 ± 0.0050	0.4571	0.4545

The data seems to be gaussian distributed. Majority voting gives the best result. Stacking adds another layer of learning, which can lead to **overfitting** if the meta-model over-learns the patterns of the base models. Majority voting, being simpler, avoids this risk.

2. Adult dataset:

The adult.test seem to have some values missing than adult.data., hence when the columns are one-hot encoded the no of columns mismatch in case of both datasets separately preprocessed. Therefore, I first combine the datasets, then preprocess & split it into a train & a test set.

Output with min-max scaling.

```
accuracy       0.7869 ± 0.0013     0.7866                0.7855
senstivity     0.5519 ± 0.0078     0.5515                0.5593
specificity    0.8596 ± 0.0037     0.8594                0.8554
precision      0.5488 ± 0.0036     0.5482                0.5448
f1_score       0.5503 ± 0.0029     0.5498                0.5520
auroc          0.7057 ± 0.0023     0.7054                0.7074
aupr           0.4087 ± 0.0018     0.4083                0.4089
```

Standard-scaling gives better results.

```
accuracy       0.7901 ± 0.0017     0.7898                0.7897
senstivity     0.3258 ± 0.0145     0.3310                0.3310
specificity    0.9417 ± 0.0050     0.9396                0.9395
precision      0.6468 ± 0.0128     0.6416                0.6411
f1_score       0.4330 ± 0.0114     0.4367                0.4366
auroc          0.6338 ± 0.0051     0.6353                0.6352
aupr           0.3766 ± 0.0044     0.3771                0.3769
```

Taking all the input features increases accuracy.

```
accuracy       0.8287 ± 0.0009     0.8284                0.8282
senstivity     0.4908 ± 0.0032     0.4913                0.4900
specificity    0.9391 ± 0.0013     0.9386                0.9387
precision      0.7246 ± 0.0037     0.7230                0.7230
f1_score       0.5852 ± 0.0023     0.5850                0.5841
auroc          0.7150 ± 0.0014     0.7149                0.7143
aupr           0.4810 ± 0.0021     0.4804                0.4798
```

The data seems to be gaussian distributed. Between LR averaging, Majority voting & Stacking LR averaging works the best. LR averaging may perform better as it may not overfit data like stacking and can balance the impact of minority class predictions more effectively than majority voting.

3. Credit-card-fraud dataset

Output of minmax scaling.

accuracy	0.9966 ± 0.0000	0.9966	0.9963
senstivity	0.8571 ± 0.0000	0.8571	0.8571
specificity	0.9998 ± 0.0000	0.9998	0.9995
precision	0.9873 ± 0.0000	0.9873	0.9750
f1_score	0.9176 ± 0.0000	0.9176	0.9123
auroc	0.9284 ± 0.0000	0.9284	0.9283
aupr	0.8495 ± 0.0000	0.8495	0.8389

Standard scaling works better than min-max scaling. Feature selection of top 20 columns increases accuracy. L2 regularization performs the best. The performance metrics are shown in the performance test data table.

L1 regularization gives the worst results.

accuracy	0.9943 ± 0.0007	0.9949	0.9949
senstivity	0.7436 ± 0.0298	0.7692	0.7692
specificity	1.0000 ± 0.0000	1.0000	1.0000
precision	1.0000 ± 0.0000	1.0000	1.0000
f1_score	0.8526 ± 0.0199	0.8696	0.8696
auroc	0.8718 ± 0.0149	0.8846	0.8846
aupr	0.7493 ± 0.0291	0.7744	0.7744

No regularization gives a somewhat better prediction than L1.

accuracy	0.9966 ± 0.0000	0.9966	0.9963
senstivity	0.8571 ± 0.0000	0.8571	0.8571
specificity	0.9998 ± 0.0000	0.9998	0.9995
precision	0.9873 ± 0.0000	0.9873	0.9750
f1_score	0.9176 ± 0.0000	0.9176	0.9123
auroc	0.9284 ± 0.0000	0.9284	0.9283
aupr	0.8495 ± 0.0000	0.8495	0.8389

The performance table above was filled for a short dataset of 20k random negative samples & all positive samples. Taking all instances may increase the accuracy but the dataset gets very skewed. Hence it’s best to work with the shorter dataset.

accuracy	0.9990 ± 0.0000	0.9990	0.9990
senstivity	0.4420 ± 0.0304	0.4333	0.4222
specificity	0.9999 ± 0.0000	0.9999	0.9999
precision	0.8879 ± 0.0066	0.8864	0.8837
f1_score	0.5897 ± 0.0283	0.5821	0.5714
auroc	0.7209 ± 0.0152	0.7166	0.7111
aupr	0.3935 ± 0.0300	0.3850	0.3740

For the larger dataset LR average, Majority voting & Stacking all give the same result, but for the shorter dataset majority voting gives the best result. This may be due to the shorter size of the data, which causes overfitting in case of stacking.