

## **Offline Assignment 2 Report**

CSE 472 Sessional

Name: Md. Zulkar Naim

Student ID: 1905016

## How to run script:

We just need to comment out dataset specific parts from **Dataset selection** code block (second code block in jupyter notebook).

### Dataset selection

```
# For first dataset
dataframe = pd.read_csv("WA_Fn-UseC_-Telco-Customer-Churn.csv")
dataframe = dataframe.drop("customerID",axis=1)
dataframe['TotalCharges'] = pd.to_numeric(dataframe['TotalCharges'], errors='coerce')
Features,Labels = preprocessing(dataframe,'Churn',feature_count=30)
X,X_test,y,y_test = train_test_split(Features, Labels, test_size=0.2, random_state=96)
X_train,X_val,y_train,y_val = train_test_split(X, y, test_size=0.2, random_state=16)

## For second dataset
# dataframe = pd.read_csv("adult.data", header=None)
# dataf = pd.read_csv("adult.test",skiprows=1,header=None)
# dataf[14] = dataf[14].replace(' <=50K.',' <=50K')
# dataf[14] = dataf[14].replace(' >50K.',' >50K')
# index = dataframe.shape[0]
# dataframe = pd.concat([dataframe,dataf], ignore_index=True)
# Features,Labels = preprocessing(dataframe,14,feature_count=30)
# X_test = Features[index:]
# y_test = Labels[index:]
# X = Features[:index]
# y = Labels[:index]
# X_train,X_val,y_train,y_val = train_test_split(X, y, test_size=0.2, random_state=16)

## For third dataset
# dataframe = pd.read_csv("creditcard.csv")
# dataframe = pd.concat([dataframe[dataframe['Class']==1], dataframe[dataframe['Class']==0].sample(n=20000, random_state=16)], ignore_index=True)
# Features,Labels = preprocessing(dataframe,'Class')
# X,X_test,y,y_test = train_test_split(Features, Labels, test_size=0.2, random_state=96)
# X_train,X_val,y_train,y_val = train_test_split(X, y, test_size=0.2, random_state=16)
```

- For first dataset (<https://www.kaggle.com/blatchar/telco-customer-churn>): Keep the first part of this code and comment the rest of this block.
- For second dataset (<https://archive.ics.uci.edu/ml/datasets/adult>): Keep the second part of this code and comment the rest of this block.
- For third dataset (<https://www.kaggle.com/mlg-ulb/creditcardfraud>): Keep the third part of this code and comment the rest of this block.

## Performance Evaluation:

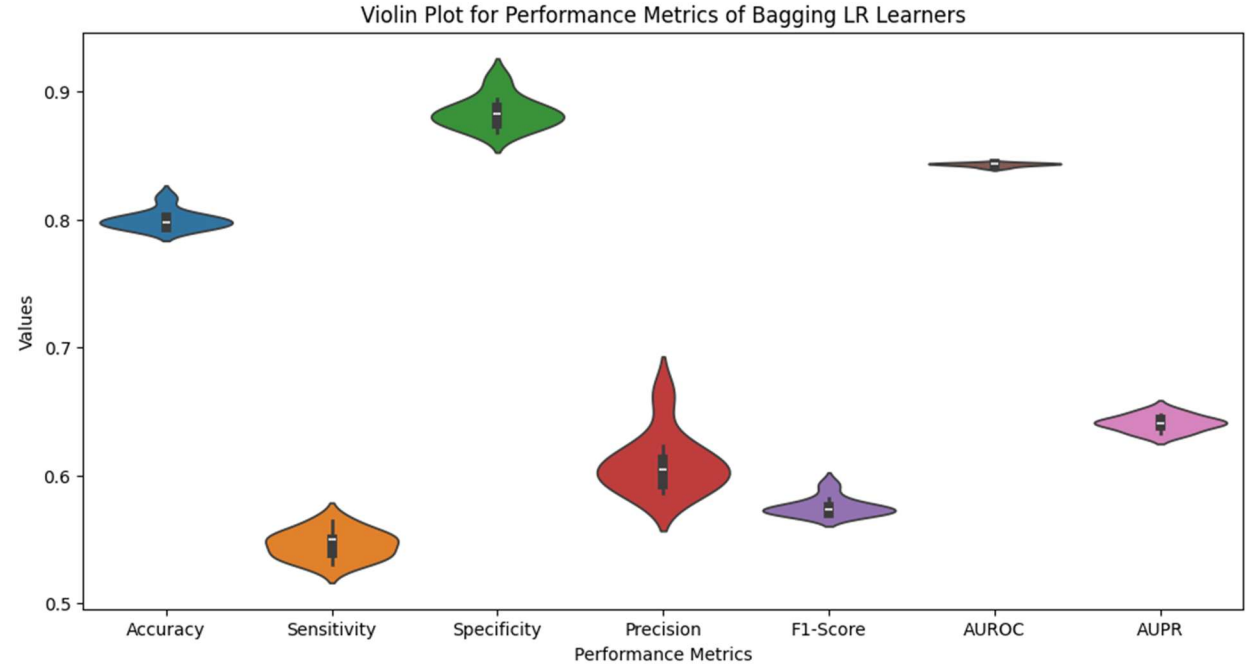
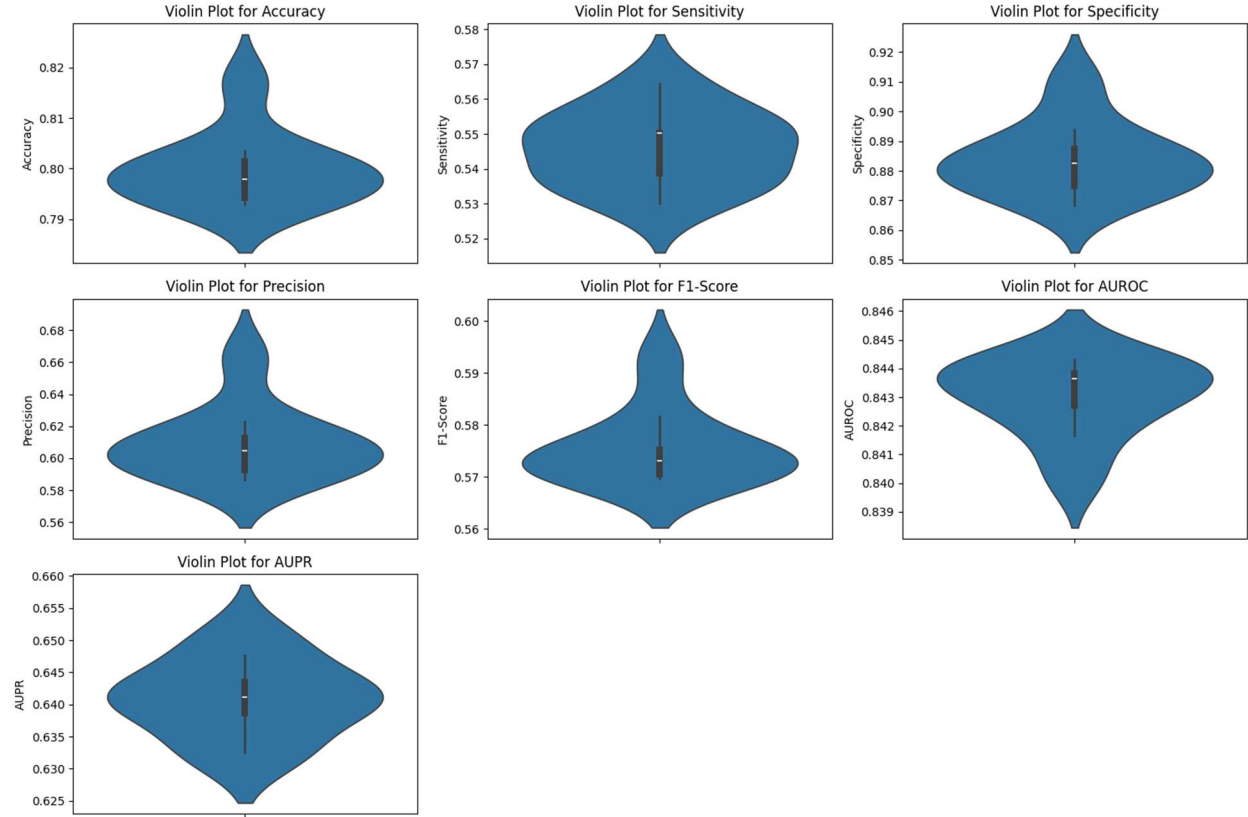
For first dataset (<https://www.kaggle.com/blastchar/telco-customer-churn>):

Learning rate = 0.0001, no regularization

### Performance on Test set

|                              | Accuracy         | Sensitivity      | Specificity      | Precision        | F1-Score         | AUROC            | AUPR             |
|------------------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| <b>LR</b>                    | 0.800 ±<br>0.007 | 0.546 ±<br>0.010 | 0.884 ±<br>0.012 | 0.610 ±<br>0.022 | 0.576 ±<br>0.007 | 0.843 ±<br>0.001 | 0.641 ±<br>0.006 |
| <b>Voting<br/>ensemble</b>   | 0.79573          | 0.538682         | 0.880682         | 0.598726         | 0.567119         | 0.844396         | 0.644381         |
| <b>Stacking<br/>ensemble</b> | 0.808541         | 0.515759         | 0.905303         | 0.642857         | 0.572337         | 0.748987         | 0.50329          |

Violin plots for each performance metric for the 9 bagging LR learners



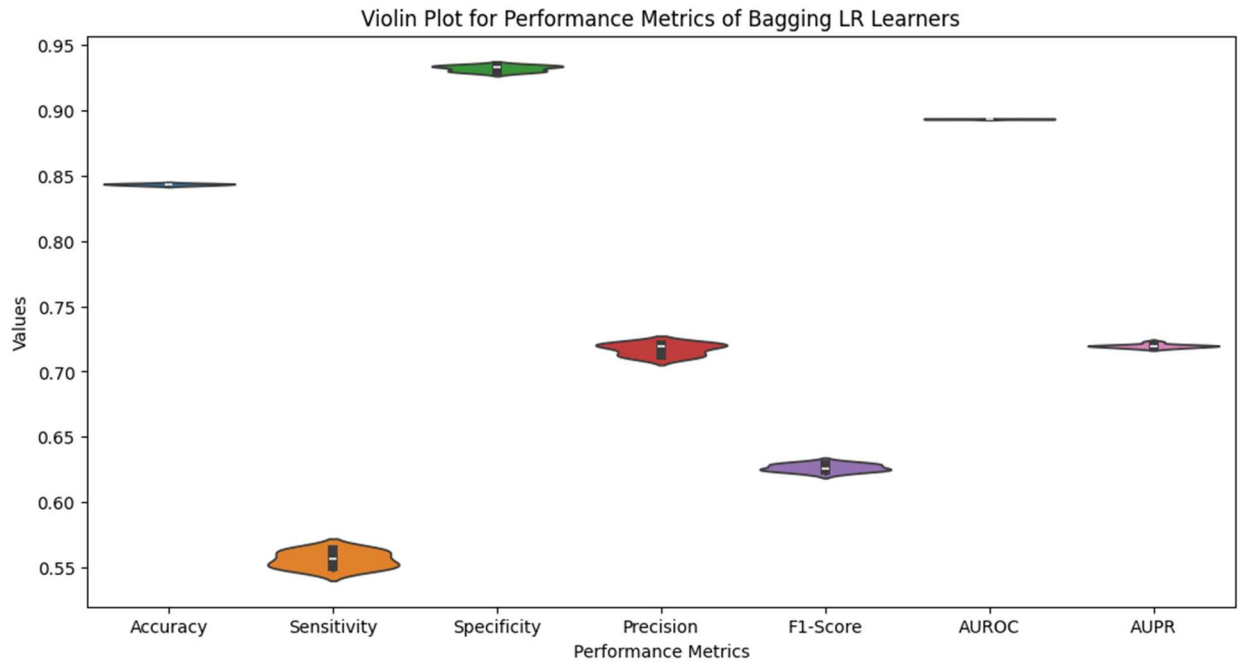
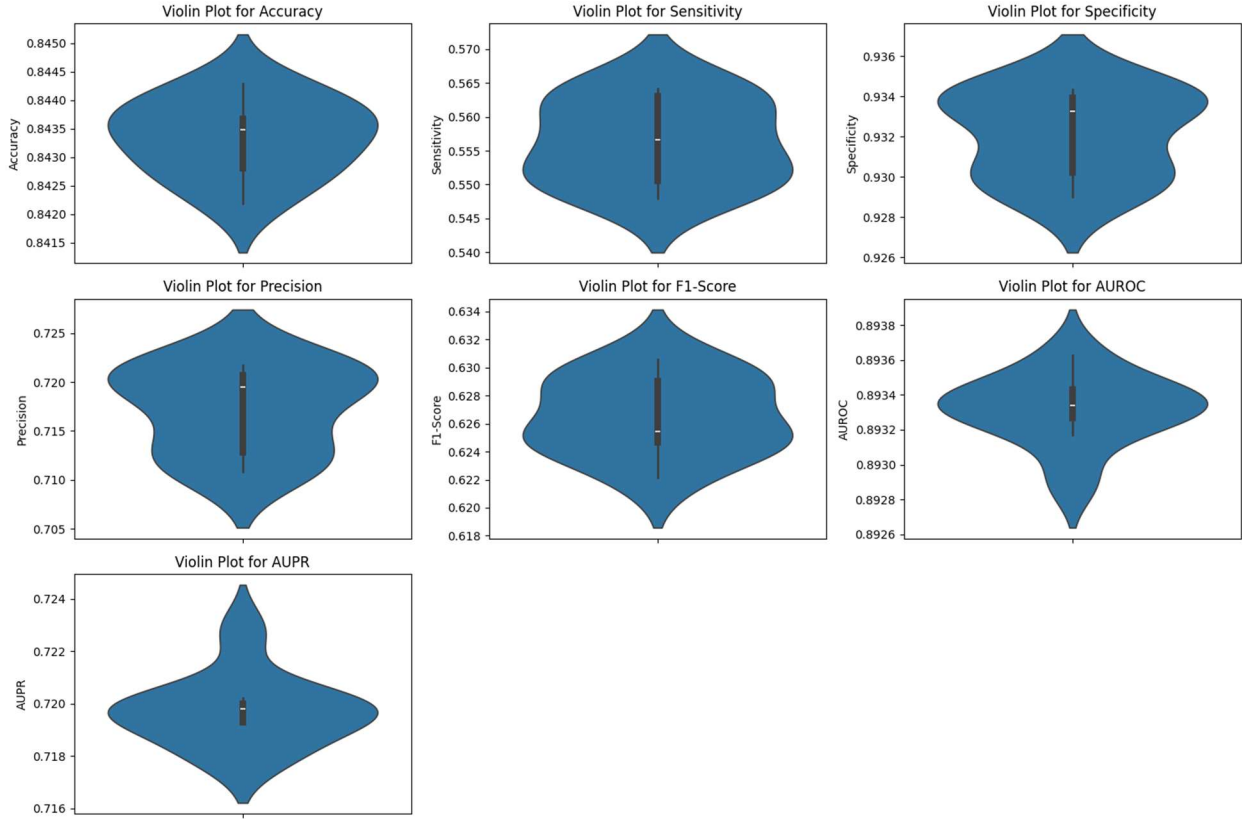
For second dataset (<https://archive.ics.uci.edu/ml/datasets/adult>):

Learning rate = 0.0001, no regularization

### Performance on Test set

|                              | Accuracy         | Sensitivity      | Specificity      | Precision        | F1-Score         | AUROC            | AUPR             |
|------------------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| <b>LR</b>                    | 0.843 ±<br>0.001 | 0.556 ±<br>0.006 | 0.932 ±<br>0.002 | 0.717 ±<br>0.004 | 0.627 ±<br>0.003 | 0.893 ±<br>0.000 | 0.720 ±<br>0.001 |
| <b>Voting<br/>ensemble</b>   | 0.843552         | 0.555295         | 0.932715         | 0.718529         | 0.626453         | 0.893589         | 0.720353         |
| <b>Stacking<br/>ensemble</b> | 0.843552         | 0.556077         | 0.932473         | 0.718087         | 0.626782         | 0.755387         | 0.522309         |

**Violin plots for each performance metric for the 9 bagging LR learners**



For third dataset (<https://www.kaggle.com/mlg-ulb/creditcardfraud>):

Learning rate = 0.01, L2 regularization, Regularization strength = 0.1

**Performance on Test set**

|                              | Accuracy         | Sensitivity      | Specificity      | Precision        | F1-Score         | AUROC            | AUPR             |
|------------------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| <b>LR</b>                    | 0.993 ±<br>0.001 | 0.766 ±<br>0.009 | 0.999 ±<br>0.001 | 0.967 ±<br>0.027 | 0.855 ±<br>0.008 | 0.978 ±<br>0.001 | 0.872 ±<br>0.002 |
| <b>Voting<br/>ensemble</b>   | 0.993405         | 0.767857         | 0.999749         | 0.988506         | 0.864322         | 0.978656         | 0.872587         |
| <b>Stacking<br/>ensemble</b> | 0.992672         | 0.776786         | 0.998744         | 0.945652         | 0.852941         | 0.888031         | 0.773377         |

# Violin plots for each performance metric for the 9 bagging LR learners

