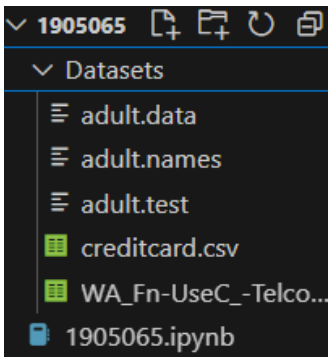
CSE472: Machine Learning

Instruction to train and evaluate:

Step 1:
Download Datasets. Save them in a **Datasets** folder (which is the same directory as 1905065.ipynb). The final directory will look like this**:**



Step 2:
Install Dependencies
pip install scikit-learn, pip install pandas, pip install matplotlib, pip install scipy, pip install seaborn
Step 3:
Run 1905065.ipynb file.
i) Give the short name for the dataset(“telco”, “adult”, “credit”) for which we will run our experiments in the 2nd last line of the first cell.
ii) Give 'Information Gain' or 'correlation' in the top_20_feature_selection_process variable in the last line of the first cell.
iii) By default ‘l1’ regularization is selected in the LogisticRegression class as the default argument. ‘l2’ can also be used here.

```
import pandas as pd
import numpy as np
from sklearn.utils import resample
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.metrics import accuracy_score, roc_auc_score, auc, confusion_matrix, f1_score, precisi
from sklearn.model_selection import train_test_split
from scipy import stats
import seaborn as sns
import matplotlib.pyplot as plt
np.random.seed(42)
#select dataset
dataset= "adult" # "adult" or "credit" or "telco"
top_20_feature_selection_process = 'Information Gain' # 'Information Gain' or 'correlation'
```

Performance Analysis:

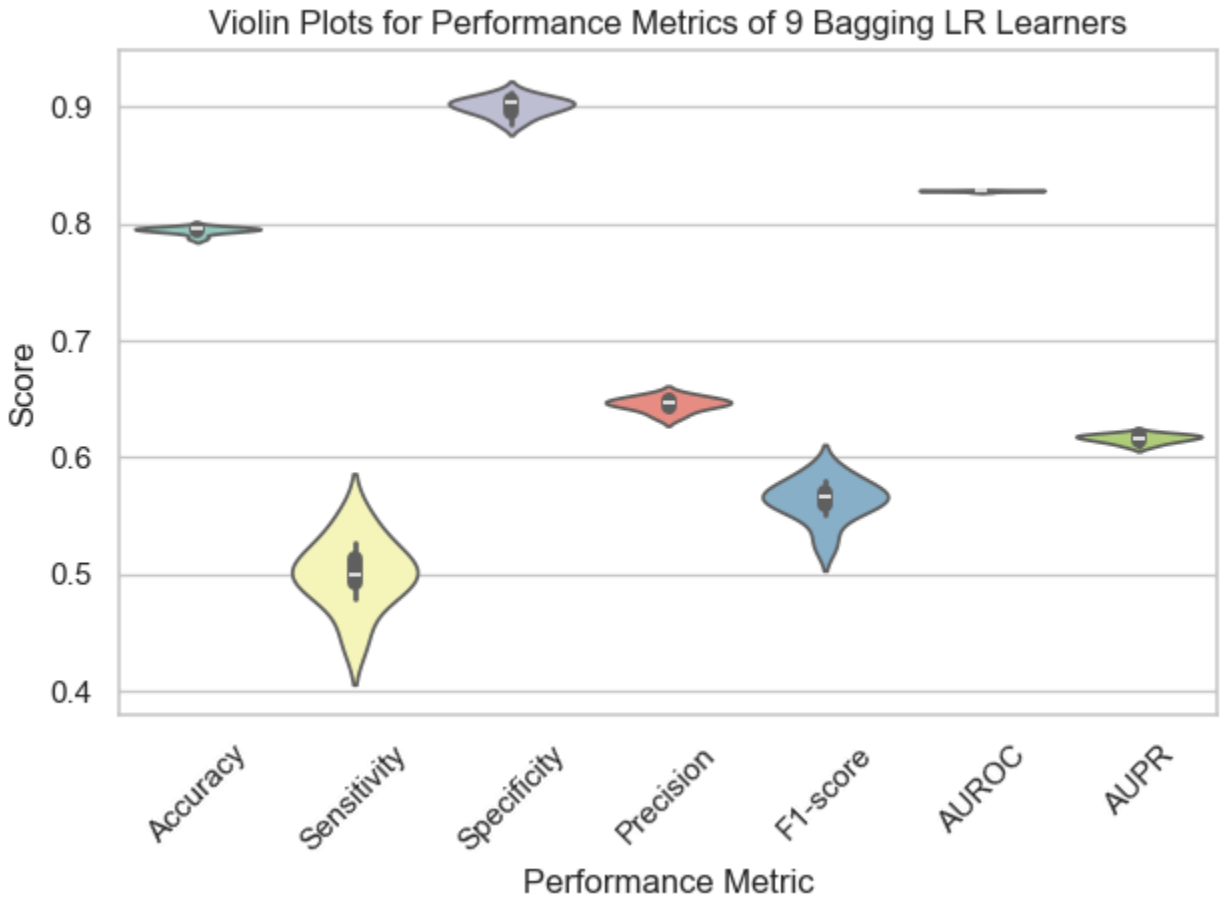
Learning Rate = 0.01 (constant), Number of Iteration = 1000(const)

Dataset1: Telco Churn Dataset

Metrics:

	Accuracy	Sensitivity	Specificity	Precision	F1-score	AUROC	AUPR
LR	0.794496 ± 0.002863	0.500594 ± 0.029526	0.900698 ± 0.007711	0.645817 ± 0.005803	0.56349 ± 0.017724	0.828284 ± 0.000543	0.616401 ± 0.003438
Voting ensemble	0.79418	0.502674	0.899517	0.643836	0.564565	0.828567	0.616611
Stacking ensemble	0.785664	0.486631	0.89372	0.623288	0.546547	0.830463	0.630463

Violin plot for 9 Bagging LR learners:



Dataset2: Adult Census Dataset

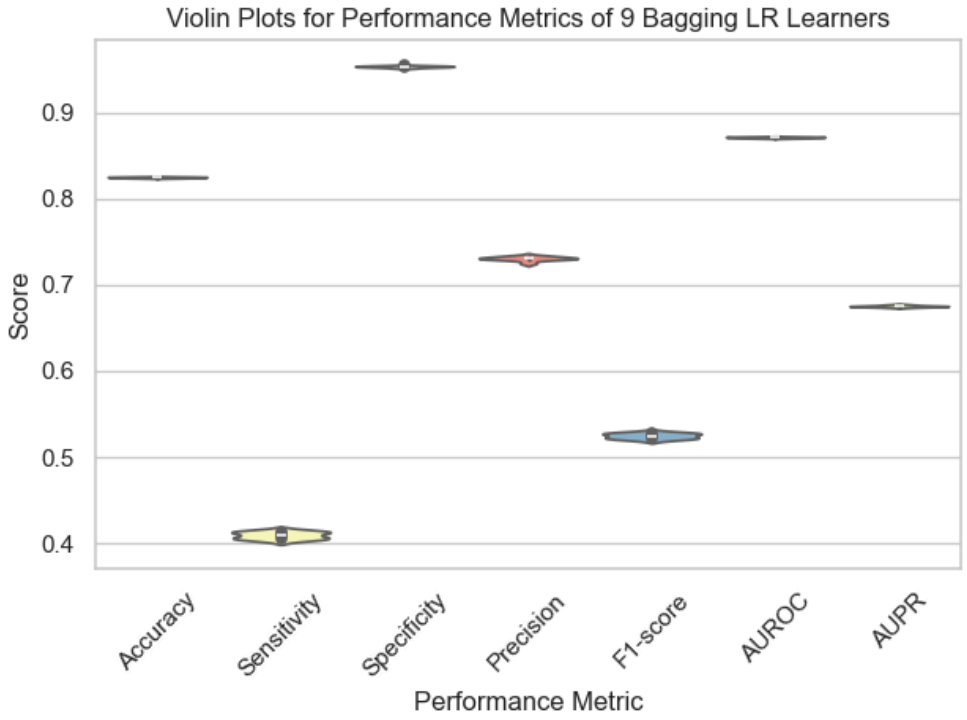
Metrics:

	Accuracy	Sensitivity	Specificity	Precision	F1-score	AUROC	AUPR
LR	0.824854 ± 0.000521	0.409668 ± 0.004028	0.953279 ± 0.000911	0.730637 ± 0.00242	0.524964 ± 0.003018	0.871252 ± 0.000576	0.675452 ± 0.000921
Voting ensemble	0.82482	0.409755	0.953207	0.730358	0.524979	0.871296	0.67561
Stacking ensemble	0.836835	0.519562	0.934974	0.711937	0.600724	0.888824	0.715227

Performance Improvement using Information gain top 20 feature selection:

	Accuracy	Sensitivity	Specificity	Precision	F1-score	AUROC	AUPR
LR	0.825313 ± 0.000465	0.427114 ± 0.00418	0.948483 ± 0.000833	0.719466 ± 0.001624	0.536004 ± 0.003018	0.869899 ± 0.000707	0.678059 ± 0.001196
Voting ensemble	0.825251	0.427491	0.948286	0.71886	0.536147	0.869957	0.67819
Stacking ensemble	0.83967	0.528951	0.935781	0.71813	0.609192	0.889849	0.719766

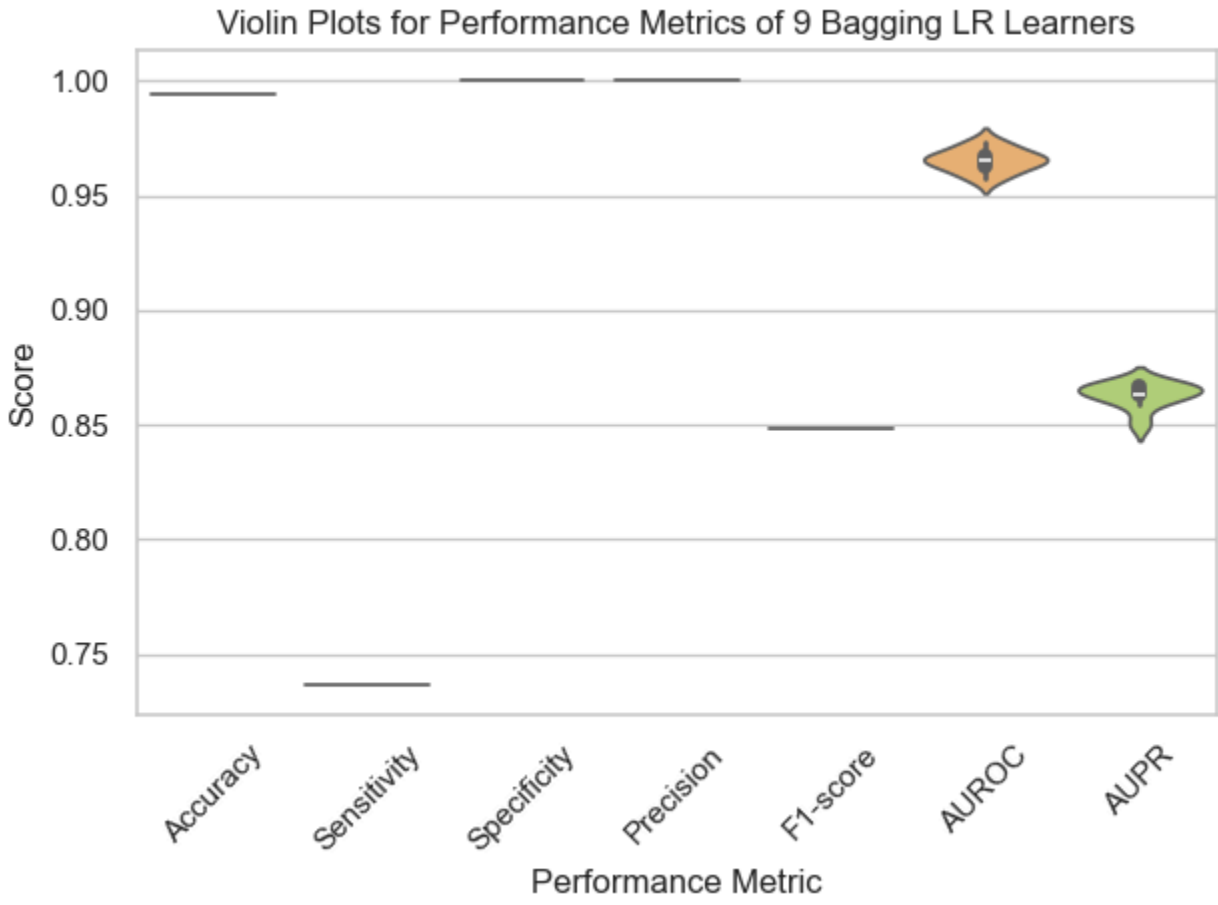
Violin Plot 9 Bagging LR learners:



Dataset 3: Credit Card Fraud Dataset
(results showing for 20k negative samples)
Metrics:

	Accuracy	Sensitivity	Specificity	Precision	F1-score	AUROC	AUPR
LR	0.993894 ± 0.0	0.736842 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	0.848485 ± 0.0	0.965595 ± 0.004848	0.863146 ± 0.005573
Voting ensemble	0.993894	0.736842	1.0	1.0	0.848485	0.967729	0.865146
Stacking ensemble	0.993894	0.747368	0.99975	0.986111	0.850299	0.967266	0.869068

Violin Plot 9 Bagging LR learners:



Observation:

1. Performance gets slightly improved when Information gain is used in place of correlation function to determine top 20 features (Dataset 2)
2. L1 regularization improves the performance on Dataset1 slightly.
3. Data is cleaned (duplication and NA removal) and then encoded with one hot encoding and scaled with standard scalar to get better performance