Student ID : 1905078

Steps to Follow:

1. **File Selection:**
   a. Choose an input option from the following:
      i. **1** for the file: "WA_Fn-UseC_-Telco-Customer-Churn.csv"
      ii. **2** for the file: "adult.data"
      iii. **3** for the file: "creditcard.csv"
2. **Target Variable:**
   a. The target variable is hard-coded into the script.
3. **Correlation-based Feature Selection (Commented Code):**
   a. The code to compute the accuracy using the top 70 features based on correlation is commented out and can be enabled as needed. Below is the relevant code block that deals with correlation and feature selection:

```
#correlation
#target_daf = pd.DataFrame(target_update, columns=['income'])
# Calculate the correlation between the features and the target (min-max scaling)
#target_series = target_daf['income']
#correlations = features_update.corrwith(target_series)
#nan_columns = correlations1[correlations1.isna()].index # Check for any NaN values in the correlations
#correlations = correlations.dropna() #dropping NaN correlations
#top_70_features = correlations.abs().sort_values(ascending=False).head(70)
#top_70_features_ind = top_70_features.index.tolist() #min-max scaling
#feature_updt = features_update[top_70_features_ind]
#features_daf1_normalized = norm.fit_transform(feature_updt)
```

4. **Feature Selection Using Mutual Information (Information Gain):** The following code block(is commented out and can be enabled as needed) calculates information gain (using mutual information) and updates the features accordingly:

```
#informtion gain {for 1st Dataset target = Churn , for 2nd Dataset target =
income and for 3rd Dataset target = Class}
target_daf = pd.DataFrame(target_update, columns=['income'])
target_series = target_daf['income']
mi_scores = mutual_info_classif(features_update, target_series, random_state=42)
mi_scores_df = pd.DataFrame(mi_scores, index=features_update.columns,
columns=['MI Score'])
mi_score_ind  = mi_scores_df.sort_values(by='MI Score',
ascending=False).index.tolist()
feature_updt = features_update[mi_score_ind]
#top_70_features = mi_scores_df.sort_values(by='MI Score',
ascending=False).head(70)
#top_70_features_ind = top_70_features.index.tolist()


#Select the top features from the original features DataFrame
#feature_updt = features_update[top_70_features_ind]
```

	Accuracy	Sensitivity	Specificity	Precision	F1-score	AUROC	AUPR
LR	0.7778566359119943	0.2815013405	0.9565637065	0.7	0.4015296367	0.6190325235	0.38725675806
Voting ensemble	0.7828246983676366	0.28150134048257375	0.9565637065637066	0.7	0.401529636671128106	0.6190325235231401	0.3872567580681068
Stacking ensemble	0.7821149751596878	0.28150134048257375	0.9565637065637066	0.7	0.401529636671128106	0.6190325235231401	0.3872567580681068

Dataset 1 : "WA_Fn-UseC_-Telco-Customer-Churn.csv (without correlation or information gain)

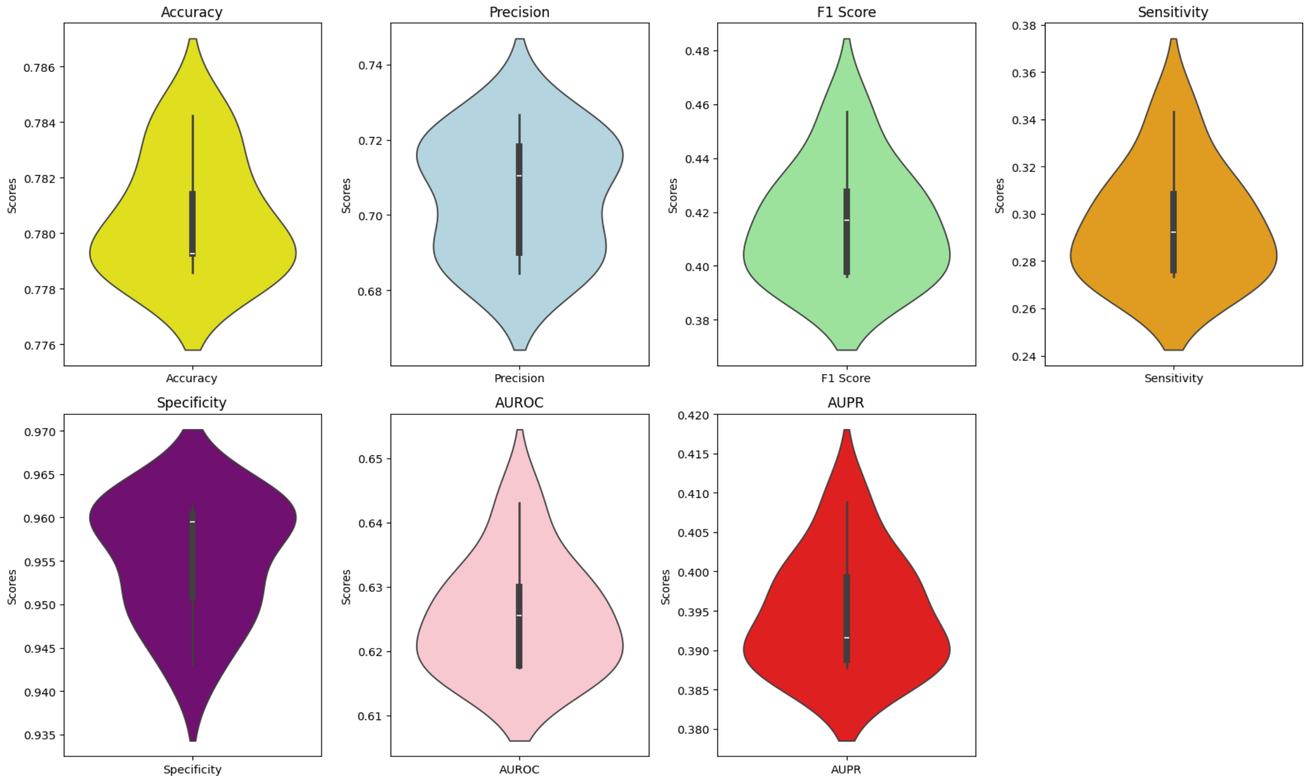

Figure 1: Violin Plot for Dataset 1

Dataset 1

Hyperparameters : learning rate = 0.1, no. of iterations = 1000, regularization = ‘l1’, lambda = 0.01

LR :

Accuracy: 0.7778566359119943
Sensitivity: 0.28150134048257375
Specificity: 0.9565637065637066
Precision: 0.7
F1 Score: 0.40152963671128106
AUROC: 0.6190325235231401
AUPR: 0.3872567580681068

Stacking Ensemble:

Accuracy: 0.7764371894960965
Sensitivity: 0.28150134048257375
Specificity: 0.9565637065637066
Precision: 0.7
F1 Score: 0.40152963671128106
AUROC: 0.6190325235231401
AUPR: 0.3872567580681068

Voting Ensemble:

```
Accuracy: 0.7785663591199432
Sensitivity: 0.28150134048257375
Specificity: 0.9565637065637066
Precision: 0.7
F1 Score: 0.40152963671128106
AUROC: 0.6190325235231401
AUPR: 0.3872567580681068
```

Hyperparameters : learning rate = 0.1, no. of iterations = 1000, regularization = ‘l2’, lambda = 0.01

LR:

```
Accuracy: 0.7778566359119943
Sensitivity: 0.28150134048257375
Specificity: 0.9565637065637066
Precision: 0.7
F1 Score: 0.40152963671128106
AUROC: 0.6190325235231401
AUPR: 0.3872567580681068
```

Stacking Ensemble:

```
Accuracy: 0.7778566359119943
Sensitivity: 0.28150134048257375
Specificity: 0.9565637065637066
Precision: 0.7
F1 Score: 0.40152963671128106
AUROC: 0.6190325235231401
AUPR: 0.3872567580681068
```

Voting Ensemble:

```
Accuracy: 0.7792760823278921
Sensitivity: 0.28150134048257375
Specificity: 0.9565637065637066
Precision: 0.7
F1 Score: 0.40152963671128106
AUROC: 0.6190325235231401
AUPR: 0.3872567580681068
```

	Accuracy	Sensitivity	Specificity	Precision	F1-score	AUROC	AUPR
LR	0.7898944347647843	0.2038999576091564	0.9767504731008381	0.7366003062787136	0.3193891102257636	0.5903252153549973	0.34266996708450437
Voting ensemble	0.7897919442451573	0.2038999576091564	0.9767504731008381	0.7366003062787136	0.3193891102257636	0.5903252153549973	0.34266996708450437
Stacking ensemble	0.7891770011273957	0.2038999576091564	0.9767504731008381	0.7366003062787136	0.3193891102257636	0.5903252153549973	0.34266996708450437

Dataset 2 : "adult.data" (without correlation or information gain)

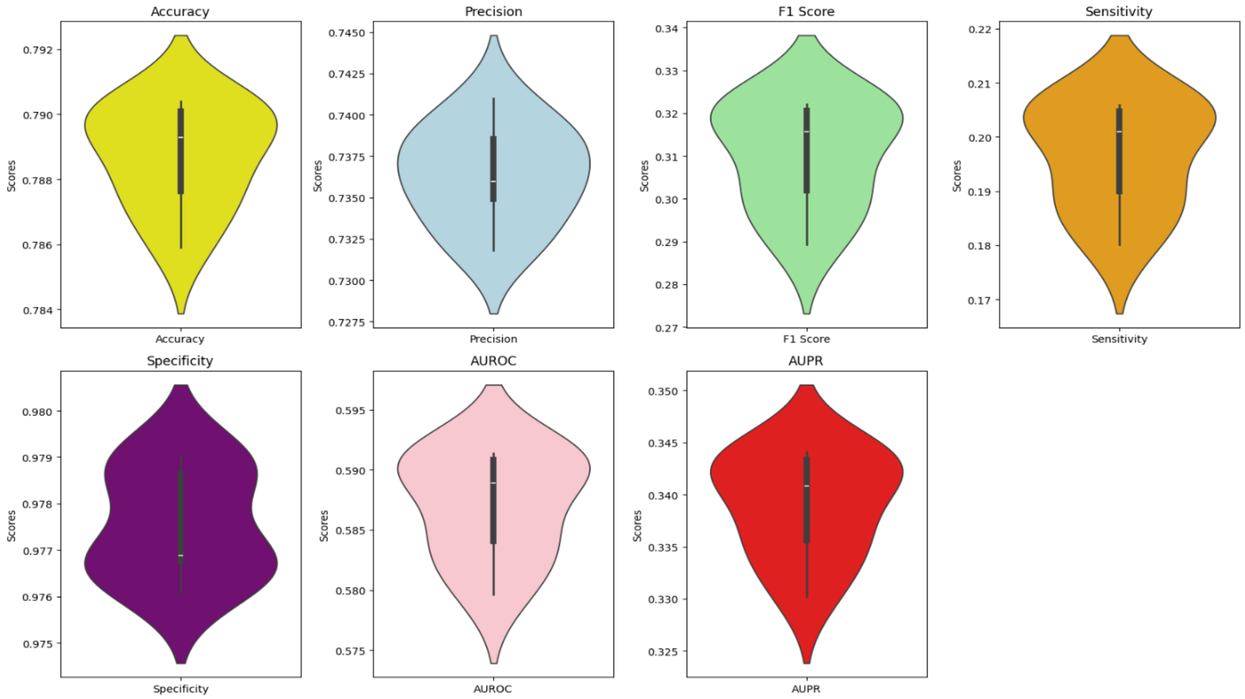

Figure 2: Violin Plot for Dataset 2

Dataset 2

Hyperparameters : learning rate = 0.1, no. of iterations = 1000, regularization = ‘l1’, lambda = 0.01

LR :

```
Accuracy: 0.7897919442451573
Sensitivity: 0.20347604917337855
Specificity: 0.9767504731008381
Precision: 0.7361963190184049
F1 Score: 0.31883095317170373
AUROC: 0.5901132611371083
AUPR: 0.34237800478885916
```

Stacking Ensemble:

```
Accuracy: 0.7891770011273957
Sensitivity: 0.20347604917337855
Specificity: 0.9767504731008381
Precision: 0.7361963190184049
F1 Score: 0.31883095317170373
AUROC: 0.5901132611371083
AUPR: 0.34237800478885916
```

Voting Ensemble :

```
Accuracy: 0.788767039048888
Sensitivity: 0.20347604917337855
Specificity: 0.9767504731008381
Precision: 0.7361963190184049
F1 Score: 0.31883095317170373
AUROC: 0.5901132611371083
AUPR: 0.34237800478885916
```

Hyperparameters : learning rate = 0.1, no. of iterations = 1000, regularization = ‘l2’, lambda = 0.01

LR :

```
Accuracy: 0.7898944347647843
Sensitivity: 0.2038999576091564
Specificity: 0.9767504731008381
Precision: 0.7366003062787136
F1 Score: 0.3193891102257636
AUROC: 0.5903252153549973
AUPR: 0.34266996708450437
```

Stacking Ensemble:

```
Accuracy: 0.7896894537255303
Sensitivity: 0.2038999576091564
Specificity: 0.9767504731008381
Precision: 0.7366003062787136
F1 Score: 0.3193891102257636
AUROC: 0.5903252153549973
AUPR: 0.34266996708450437
```

Voting Ensemble:

```
Accuracy: 0.7896894537255303
Sensitivity: 0.2038999576091564
Specificity: 0.9767504731008381
Precision: 0.7366003062787136
F1 Score: 0.3193891102257636
AUROC: 0.5903252153549973
AUPR: 0.34266996708450437
```

	Accuracy	Sensitivity	Specificity	Precision	F1-score	AUROC	AUPR
LR	0.998413985126705	0.0	1.0	0.0	0.0	0.5	0.0015860148732950341
Voting ensemble	0.998413985126705	0.0	1.0	0.0	0.0	0.5	0.0015860148732950341
Stacking ensemble	0.998413985126705	0.0	1.0	0.0	0.0	0.5	0.0015860148732950341

Dataset 3 : “creditcard.csv” (without correlation or information gain)

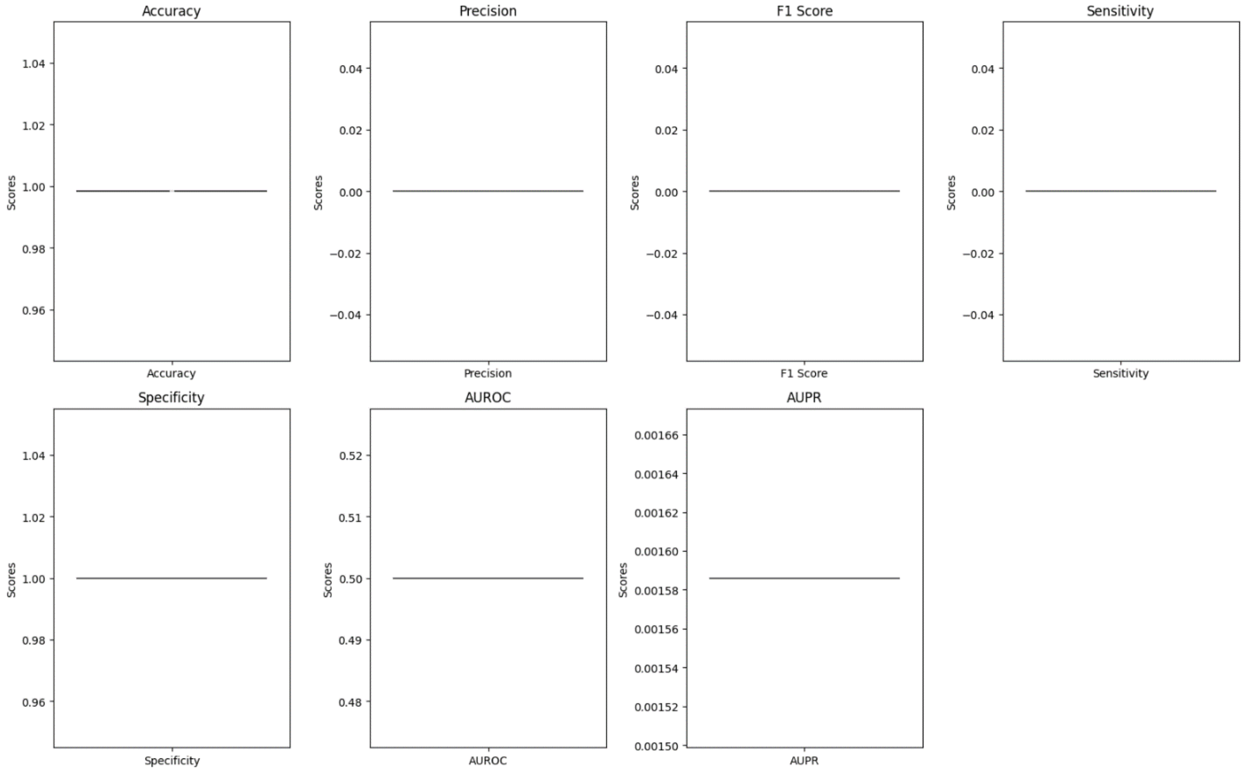

Figure 3 : Violin Plot for Dataset 3

Dataset 3

Hyperparameters : learning rate = 0.1, no. of iterations = 1000, regularization = ‘l1’, lambda = 0.01

LR :

```
Accuracy: 0.998413985126705
Sensitivity: 0.0
Specificity: 1.0
Precision: 0.0
F1 Score: 0.0
AUROC: 0.5
AUPR: 0.0015860148732950341
```

Stacking Ensemble:

```
Accuracy: 0.998413985126705
Sensitivity: 0.0
Specificity: 1.0
Precision: 0.0
F1 Score: 0.0
AUROC: 0.5
AUPR: 0.0015860148732950341
```

Voting Ensemble :

```
Accuracy: 0.998413985126705
Sensitivity: 0.0
Specificity: 1.0
Precision: 0.0
F1 Score: 0.0
AUROC: 0.5
AUPR: 0.0015860148732950341
```

Hyperparameters : learning rate = 0.1, no. of iterations = 1000, regularization = ‘l2’, lambda = 0.01

LR :

```
Accuracy: 0.998413985126705
Sensitivity: 0.0
Specificity: 1.0
Precision: 0.0
F1 Score: 0.0
AUROC: 0.5
AUPR: 0.0015860148732950341
```

Stacking Ensemble:

```
Accuracy: 0.998413985126705
Sensitivity: 0.0
Specificity: 1.0
Precision: 0.0
F1 Score: 0.0
```

```
AUROC: 0.5
AUPR: 0.0015860148732950341
```

Voting Ensemble:

```
Accuracy: 0.998413985126705
Sensitivity: 0.0
Specificity: 1.0
Precision: 0.0
F1 Score: 0.0
AUROC: 0.5
AUPR: 0.0015860148732950341
```