Ensembling and Logistic Regression Models Training and Evaluation Report

Asif Azad — 1905004

September 20, 2024

1 Training and Testing Instructions

- Ensure that the following Python packages are installed in your environment:
  - `numpy`
  - `pandas`
  - `scikit-learn`
  - `seaborn`
  - `matplotlib`
- The notebook includes code for training and evaluating models on three datasets:
  - Telco Customer Churn
  - Adult Census Income
  - Credit Card Fraud Detection

  By default, the last three cells in the notebook are uncommented, meaning that running the notebook will execute the code for all three datasets. To run the notebook on a specific dataset, simply comment out the cells for the other datasets.
- The model configuration is defined in the `config` dictionary within each dataset block. The configuration contains the following attributes:
  - `lr`: Learning rate for the model
  - `l1_lambda`: L1 regularization factor
  - `l2_lambda`: L2 regularization factor
  - `epoch`: Number of epochs to train the model
  - `batch_size`: Batch size for mini-batch gradient descent

- `n_estimators`: Number of estimators for ensemble models
- `verbose`: Boolean flag to control verbosity of training output

You can adjust these parameters directly in the notebook for each dataset.

- Visualizations such as violin plots and metrics tables are generated automatically during the execution of the notebook, providing insights into model performance on each dataset.

2 Performance Evaluation

2.1 Telco Customer Churn Dataset

Total Features: 45
Total samples: 7010
Train samples: 4486
Validation samples: 1122
Test samples: 1402
Configuration: `lr = 0.1, l1_lambda = 0.0, l2_lambda = 0.01, epoch = 1000, batch_size = 1000000, n_estimators = 9, verbose = False`

Model	Accuracy	Sensitivity	Specificity	Precision	F1	AUROC
logistic_regressor	0.7019	0.4243	0.8474	0.5655	0.8828	0.8451
mean_ensembler	0.7047	0.4270	0.8474	0.5679	0.8816	0.8459
multiple_regressor	0.70/0.0069	0.42/0.0061	0.85/0.0085	0.56/0.0051	0.88/0.0080	0.84/0.0021
voting_ensembler	0.7011	0.4237	0.8474	0.5649	0.8831	0.8453
stacking_ensembler	0.8046	0.5776	0.5452	0.5609	0.4672	0.8419

Table 1: Performance on the Telco Customer Churn dataset.

2.2 Adult Census Income Dataset

Total Features: 54
Total samples: 29096
Train samples: 18620
Validation samples: 4656
Test samples: 5820
Configuration: `lr = 0.1, l1_lambda = 0.0, l2_lambda = 0.01, epoch = 1000, batch_size = 1000000, n_estimators = 9, verbose = False`

Model	Accuracy	Sensitivity	Specificity	Precision	F1	AUROC
logistic_regressor	0.7871	0.5337	0.8443	0.6540	0.8257	0.8895
mean_ensembler	0.7893	0.5370	0.8421	0.6558	0.8214	0.8894
multiple_regressor	0.79/0.0040	0.53/0.0062	0.84/0.0094	0.65/0.0024	0.82/0.0137	0.89/0.0011
voting_ensembler	0.7878	0.5349	0.8407	0.6538	0.8211	0.8894
stacking_ensembler	0.8361	0.6849	0.5782	0.6271	0.3868	0.8894

Table 2: Performance on the Adult Census Income dataset.

2.3 Credit Card Fraud Detection Dataset

Total Features: 30
Total samples: 20468
Train samples: 13099
Validation samples: 3275
Test samples: 4094
Configuration: `lr = 0.0001, l1_lambda = 0.0, l2_lambda = 0.002, epoch = 1000, batch_size = 1000000, n_estimators = 9, verbose = False`

Model	Accuracy	Sensitivity	Specificity	Precision	F1	AUROC
logistic_regressor	0.8752	0.1397	0.9318	0.2430	0.9883	0.9693
mean_ensembler	0.8774	0.1419	0.9318	0.2462	0.9880	0.9703
multiple_regressor	0.87/0.0072	0.14/0.0066	0.93/0.0036	0.24/0.0099	0.99/0.0011	0.97/0.0020
voting_ensembler	0.8796	0.1441	0.9318	0.2496	0.9878	0.9704
stacking_ensembler	0.9968	0.9870	0.8636	0.9212	0.0769	0.9652

Table 3: Performance on the Credit Card Fraud Detection dataset.

3 Observations

- **Effect of Initialization**: Initializing the logistic regressor with zeros rather than a Gaussian distribution significantly improved performance across all models on Dataset 3. This was particularly noticeable in terms of *accuracy* and *precision*. For instance, the logistic regressor's accuracy improved from 0.6406 (Gaussian) to 0.8752 (zeros), and precision increased from 0.1036 to 0.2429. Other models, such as the *stacking ensembler*, saw improvements as well, with the accuracy increasing from 0.9792 to 0.9968.

- **Stacking Ensembler Performance**: The stacking ensembler consistently performed better across all datasets, particularly in terms of *accuracy* and *precision*. It showed superior performance on Dataset 3, with a near-perfect sensitivity of 0.9870 after the initialization change.

4 Visualizations

Below are visualizations for each dataset generated using Seaborn:

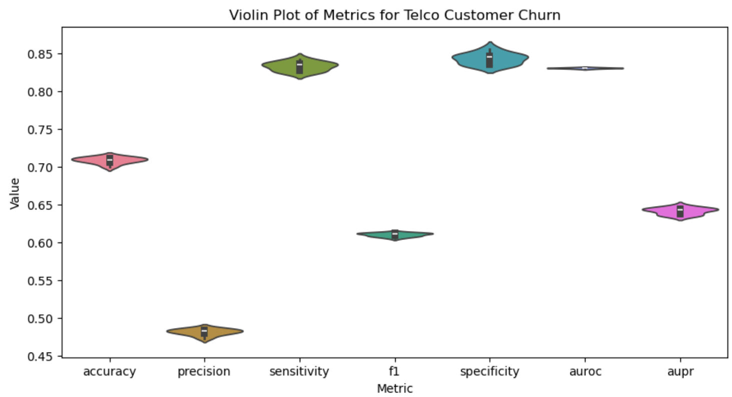

Figure 1: Telco Customer Churn - Model Performance

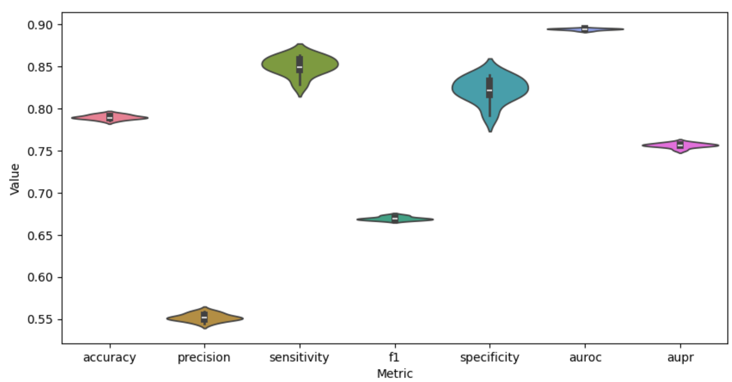

Figure 2: Adult Census Income - Model Performance

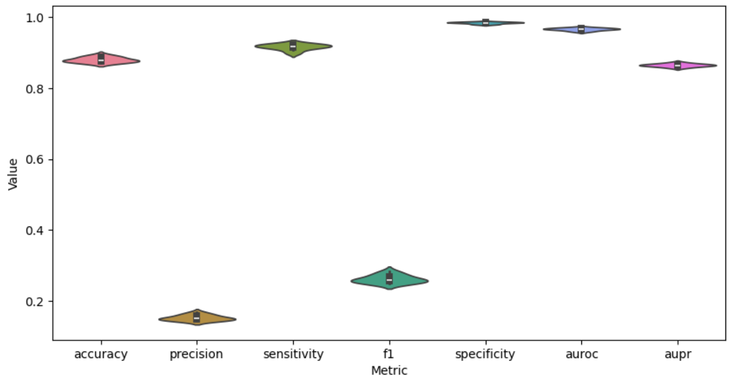

Figure 3: Credit Card Fraud Detection - Model Performance