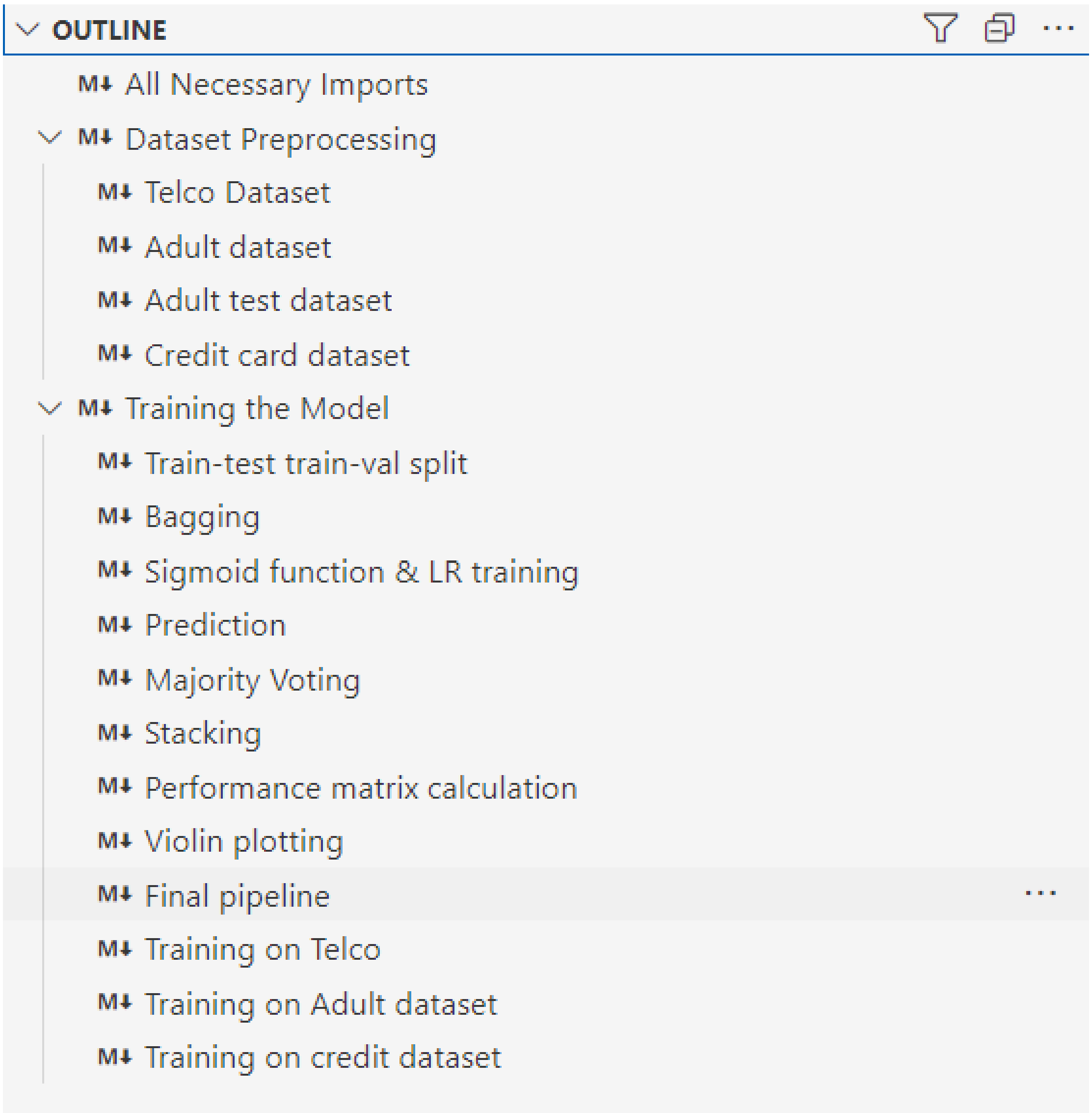
My Code Outline



- For all three datasets, their corresponding pre-processing steps will be displayed under the relevant markdown sections. We will execute these steps as needed.
- All datasets are stored in the same folder as the .ipynb file for consistency.
- First,we need to execute the necessary import statements for all required libraries.
- Then, we should run the functions listed under the Dataset Preprocessing markdown. Since each dataset has been analysed separately, the preprocessing will function independently for each.
- No part of the code should be commented out. The entire code will be executed.
- All cells can be run consecutively, and the results will be displayed separately for each of the three datasets in a structured manner.

Comparative Analysis on Telco Dataset

Learning rate = 0.6
Beta = 0.1

Sigmoid function & LR training

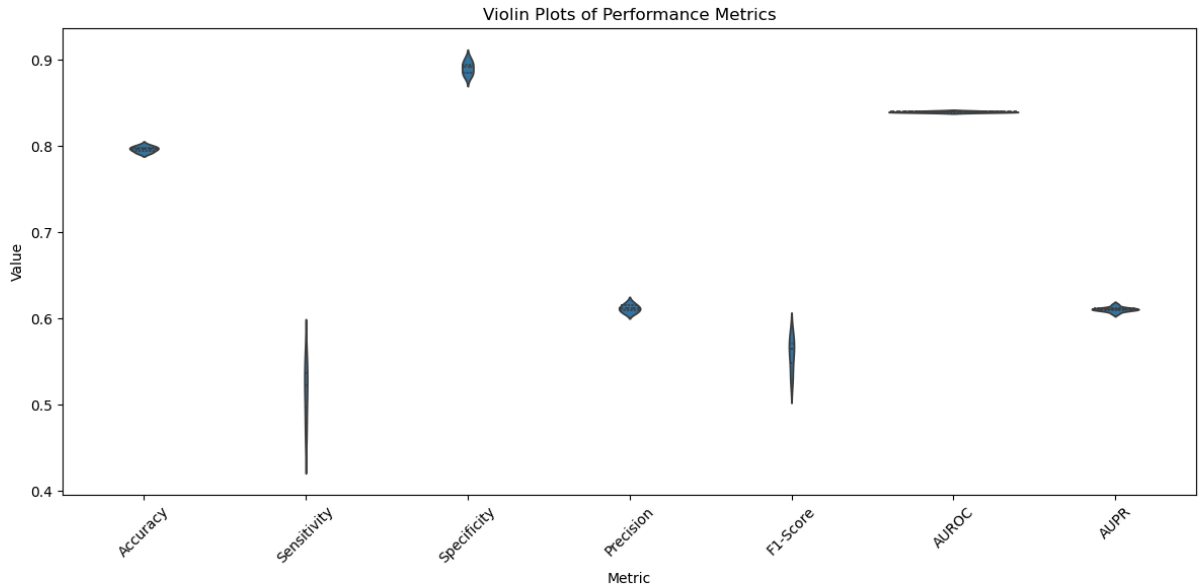
```
def sigmoid_function(z):
    return 1 / (1 + np.exp(-z))    # match probability to [0, 1]

def train_logistic_regression(X, y, regularization='l1', learning_rate=0.6, n_iters=50):
    beta = 0.1
```

Table:

	Accuracy	Sensitivity	Specificity	Precision	F1-Score	AUROC	AUPR
LR*	0.7971 ± 0.0015	0.5221 ± 0.0215	0.8890 ± 0.0070	0.6115 ± 0.0062	0.5629 ± 0.0106	0.8395 ± 0.0007	0.6091 ± 0.0023
Voting ensemble	0.7986	0.5227	0.8908	0.6154	0.5653	0.8398	0.6094
Stacking ensemble	0.7979	0.5170	0.8917	0.6149	0.5617	0.7255	0.4641

Plot for 9 base learners:



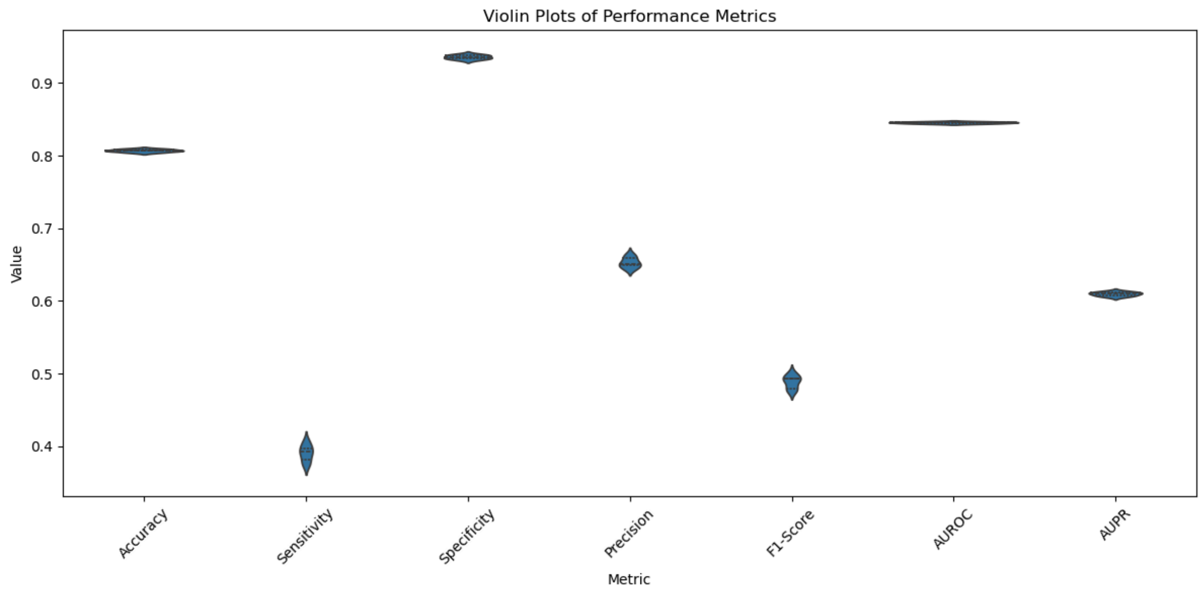
Comparative Analysis on Adult Dataset

Learning rate = 0.6
Beta = 0.1

Table:

	Accuracy	Sensitivity	Specificity	Precision	F1-Score	AUROC	AUPR
LR*	0.8091 ± 0.0018	0.4020 ± 0.0151	0.9350 ± 0.0028	0.6569 ± 0.0041	0.4986 ± 0.0119	0.8462 ± 0.0012	0.6121 ± 0.0025
Voting ensemble	0.8100	0.4056	0.9352	0.6593	0.5023	0.8464	0.6123
Stacking ensemble	0.8093	0.3950	0.9375	0.6616	0.4946	0.6774	0.4159

Plot for 9 base learners:



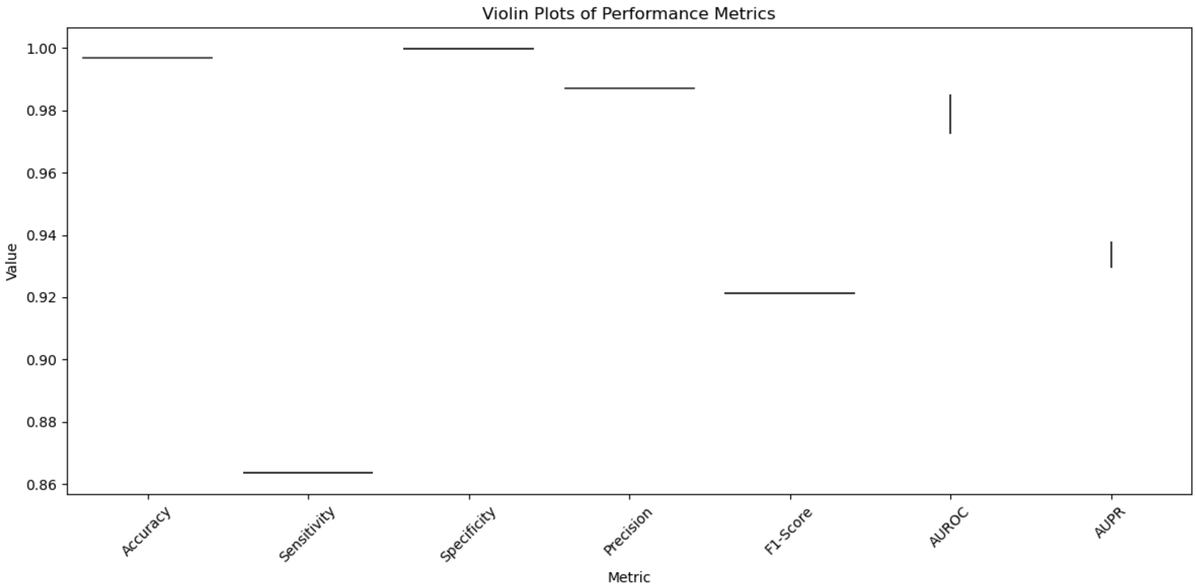
Comparative Analysis on Credit card Dataset

Learning rate = 0.6
Beta = 0.1

Table:

	Accuracy	Sensitivity	Specificity	Precision	F1-Score	AUROC	AUPR
LR*	0.9968 ± 0.0000	0.8636 ± 0.0000	0.9998 ± 0.0000	0.9870 ± 0.0000	0.9212 ± 0.0000	0.9772 ± 0.0022	0.9327 ± 0.0008
Voting ensemble	0.9968	0.8636	0.9998	0.9870	0.9212	0.9776	0.9326
Stacking ensemble	0.9785	0.0000	1.0000	0.0000	0.0000	0.9317	0.8554

Plot for 9 base learners:



Observations

- The model shows improved performance with an increased learning rate.
- Due to the class imbalance in the Credit Card Fraud Detection dataset, the performance metrics are predominantly influenced by the negative class.
- Increasing the number of epochs tends to improve the overall accuracy of the model.