STTAI-Assignment6-Reproducibility-HPO

[GitHub Link]

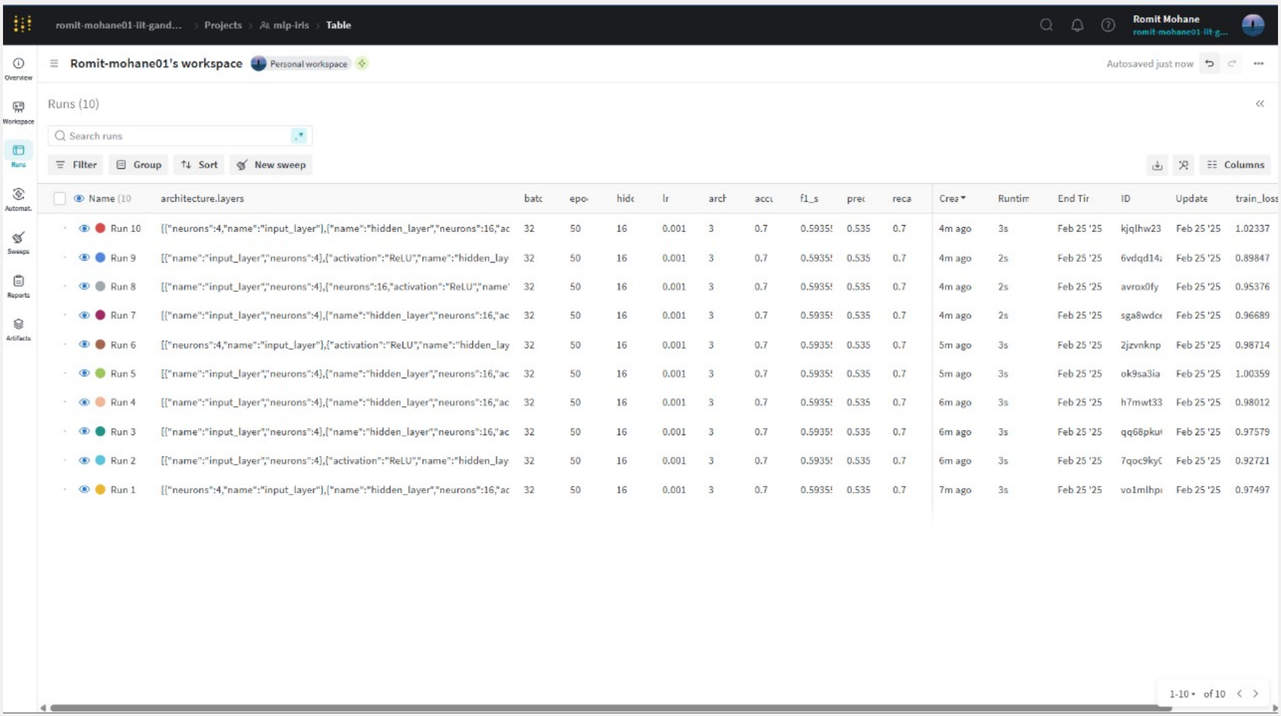
Team 25
Members:

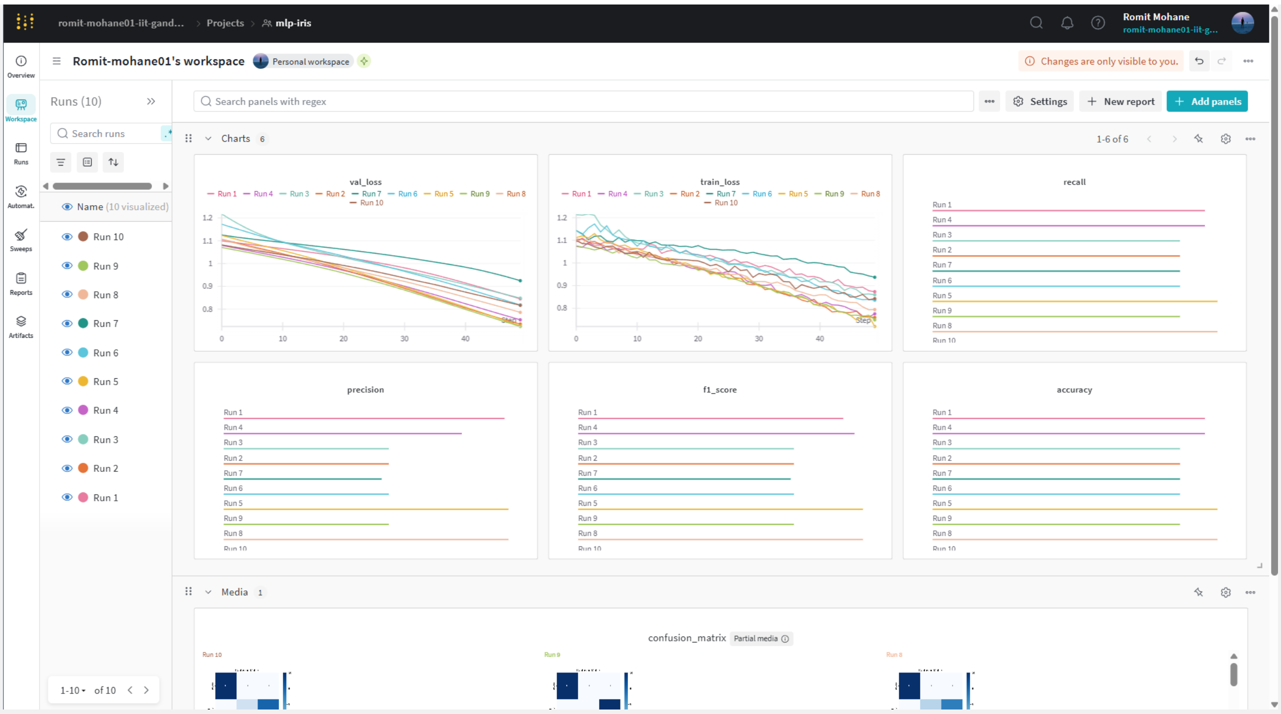
Name	Roll Number
Romit Mohane	23110279
Rudra Pratap Singh	23110281

Introduction

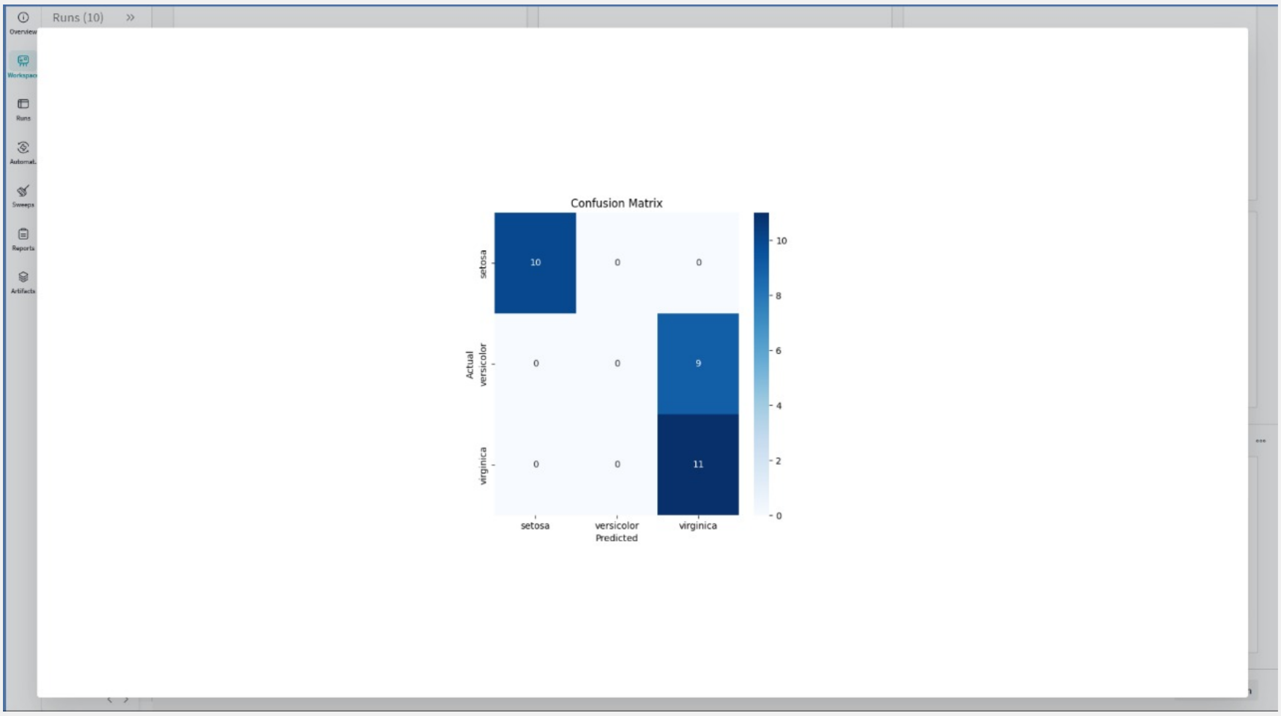
The goal of this assignment is to learn about experiment tracking, version control, and reproducibility in machine learning workflows. You will set up experiment tracking using Weights and Biases.

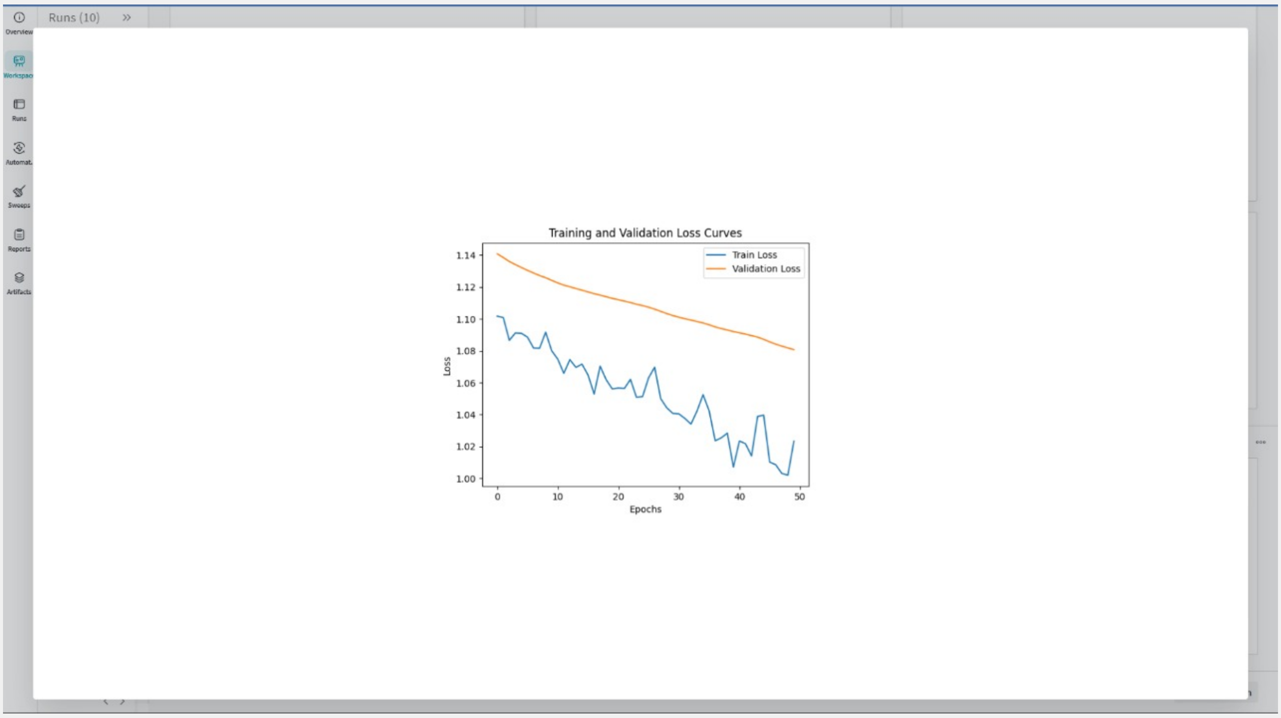
Section 1: MLP Model Implementation & Experiment Tracking

> **In file** `Section1.ipynb`
>
> *Screenshots*
> Model architecture, Hyperparams, logged metrics
>
> 
>
>
> Final Evalutation results



Confusion matrix visualisation



> Training and Validation loss curves
>
> 
>

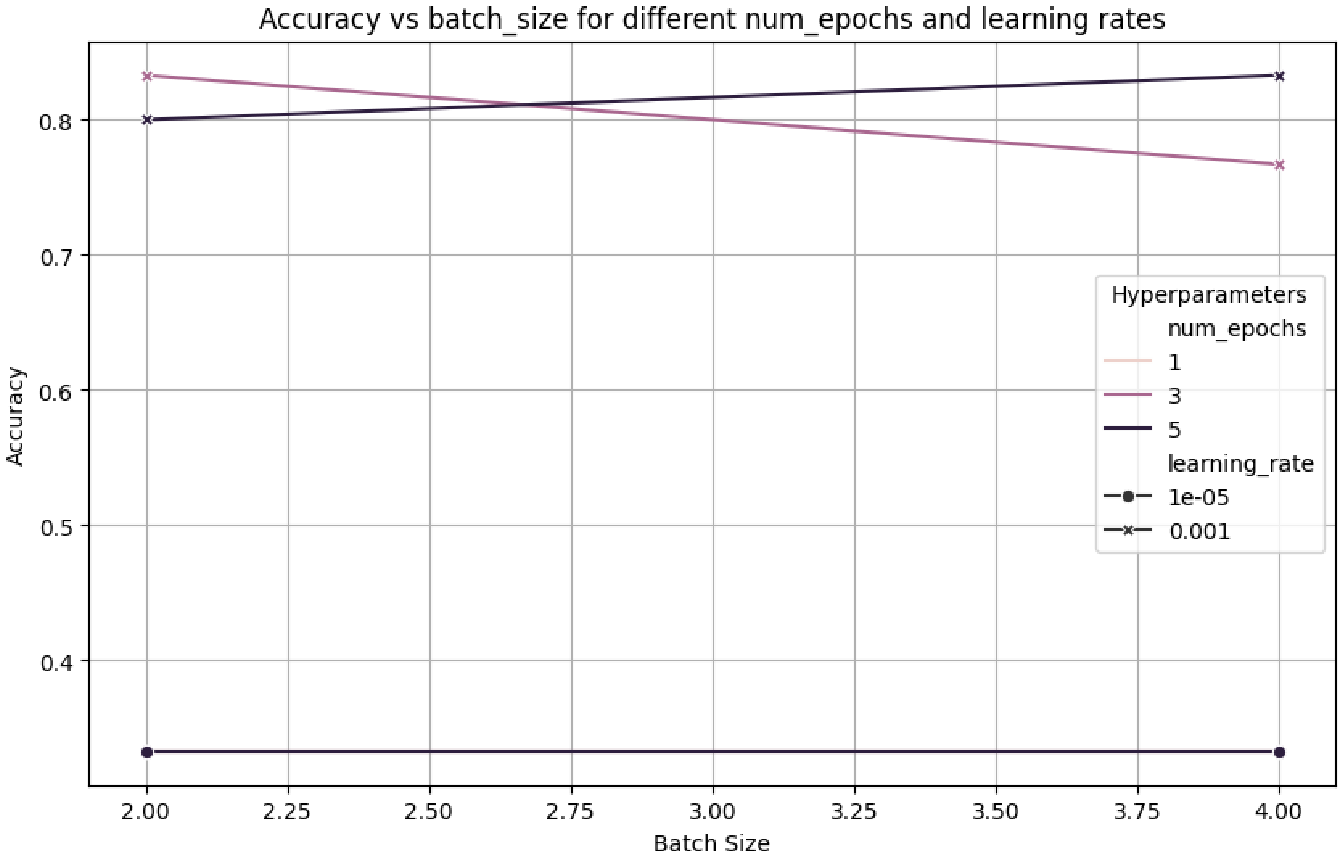
Section 2: Hyperparam optimization

> **In file `Section2.ipynb`**

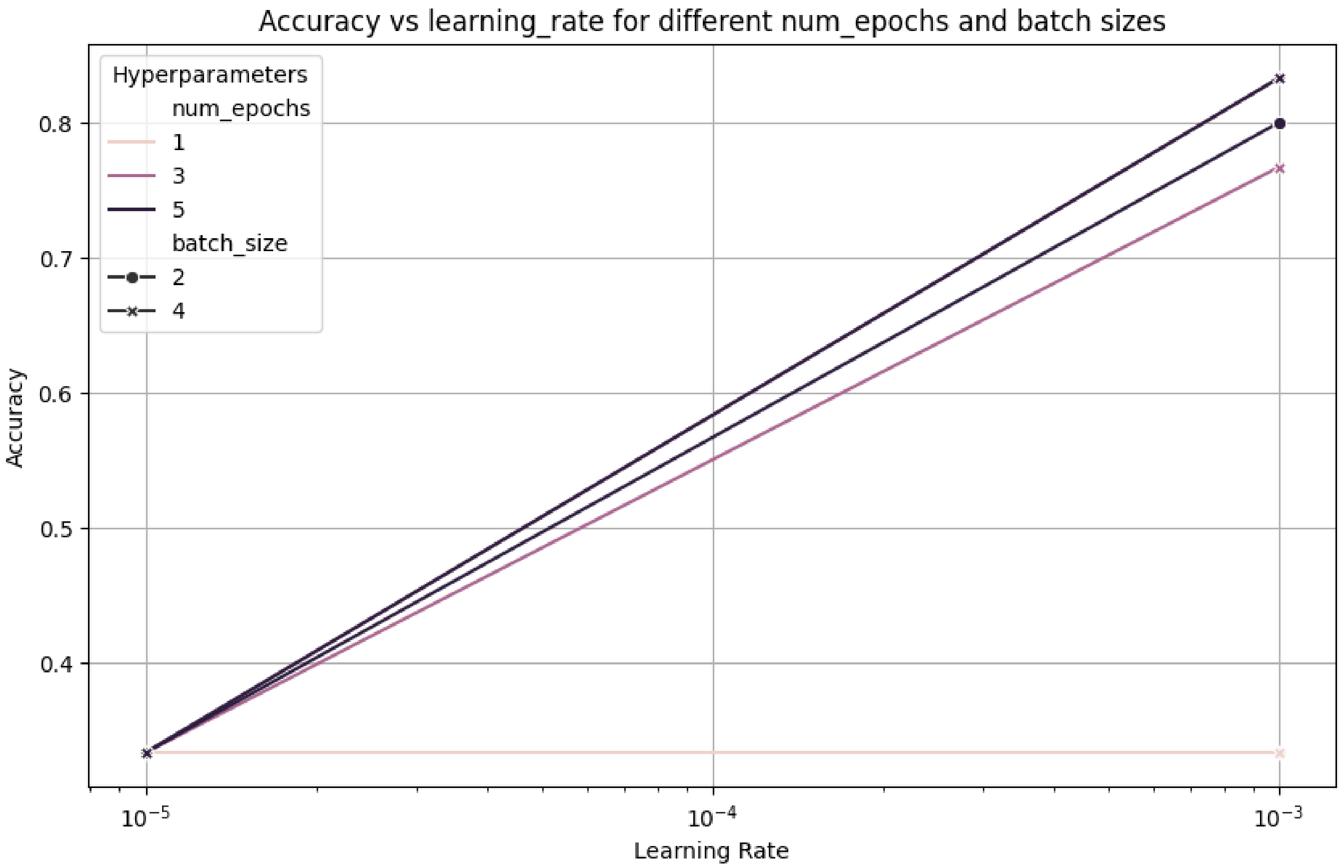
Task 1:

> Relation between the different HyperParam and their impact on the performance of the model.
>
> | Hyperparam | Relation | Description |
> | --- | --- | --- |
> | epochs | direct | as the model gets more epochs to see the data, it learns more > patterns from it. |
> | learning rate | direct | as learning rate increases, the optimizer takes bigger > gradient steps and approaches the minima faster |
> | batch size | inverse | larger batch size causes less updates per epoch, degrading > performance |

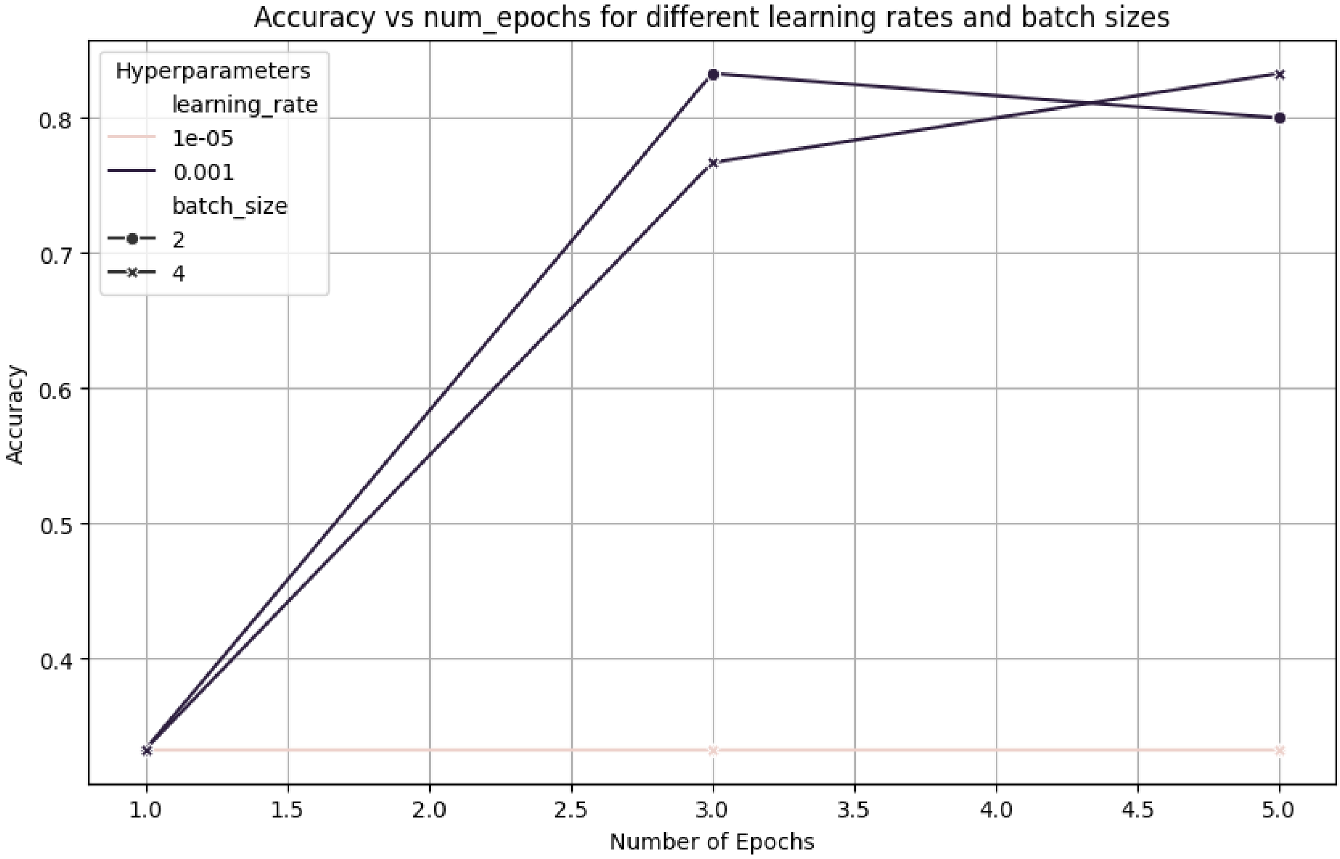
Accuracy vs batch size



Accuracy vs lr



Accuracy vs epochs



Task 2:

Tables with configurations and column with Accuracy for different search strategies

Random

	Model Name	Batch Size	Num Epochs	Learning Rate	Train Accuracy	Validation Accuracy
0	NeuralNetTorch/4f3e5_00000	2	1	0.000010	0.390476	0.400000
1	NeuralNetTorch/4f3e5_00001	4	5	0.000268	0.704762	0.666667
2	NeuralNetTorch/4f3e5_00002	3	3	0.000888	0.809524	0.666667
3	NeuralNetTorch/4f3e5_00003	4	6	0.000722	0.828571	0.700000
4	NeuralNetTorch/4f3e5_00004	4	5	0.000546	0.800000	0.733333

Grid

	Model Name	Batch Size	Num Epochs	Learning Rate	Train Accuracy	Validation Accuracy
0	NeuralNetTorch/34c05_00000	2	1	0.00100	0.761905	0.666667
1	NeuralNetTorch/34c05_00001	4	1	0.00001	0.342857	0.333333
2	NeuralNetTorch/34c05_00002	2	1	0.00001	0.533333	0.600000
3	NeuralNetTorch/34c05_00003	2	1	0.00100	0.752381	0.766667
4	NeuralNetTorch/34c05_00004	4	3	0.00001	0.390476	0.433333
5	NeuralNetTorch/34c05_00005	2	1	0.00100	0.333333	0.333333
6	NeuralNetTorch/34c05_00006	2	5	0.00100	0.666667	0.766667
7	NeuralNetTorch/34c05_00007	4	3	0.00001	0.523810	0.566667
8	NeuralNetTorch/34c05_00008	4	3	0.00001	0.552381	0.600000
9	NeuralNetTorch/34c05_00009	2	5	0.00001	0.333333	0.333333
10	NeuralNetTorch/34c05_00010	2	3	0.00100	0.733333	0.766667
11	NeuralNetTorch/34c05_00011	4	5	0.00100	0.876190	0.800000

Bayesian

	Model Name	Batch Size	Num Epochs	Learning Rate	Train Accuracy	Validation Accuracy
0	NeuralNetTorch/8f5501b0	2	1	0.000010	0.390476	0.400000
1	NeuralNetTorch/5f4d1eb5	3	4	0.000049	0.400000	0.500000
2	NeuralNetTorch/875f4c18	3	7	0.000998	0.819048	0.833333
3	NeuralNetTorch/ad330e8b	2	4	0.000706	0.600000	0.533333
4	NeuralNetTorch/99c07629	4	4	0.000854	0.695238	0.766667
5	NeuralNetTorch/78b2ea69	2	8	0.000646	0.904762	0.966667
6	NeuralNetTorch/5dfac753	3	5	0.000331	0.666667	0.666667
7	NeuralNetTorch/a9d13025	3	6	0.000230	0.333333	0.333333
8	NeuralNetTorch/95db37e8	3	7	0.000340	0.685714	0.733333
9	NeuralNetTorch/01b2ef91	3	5	0.000496	0.742857	0.766667
10	NeuralNetTorch/481e02af	3	3	0.000931	0.819048	0.766667
11	NeuralNetTorch/52560615	4	9	0.000602	0.742857	0.766667

Hyperband

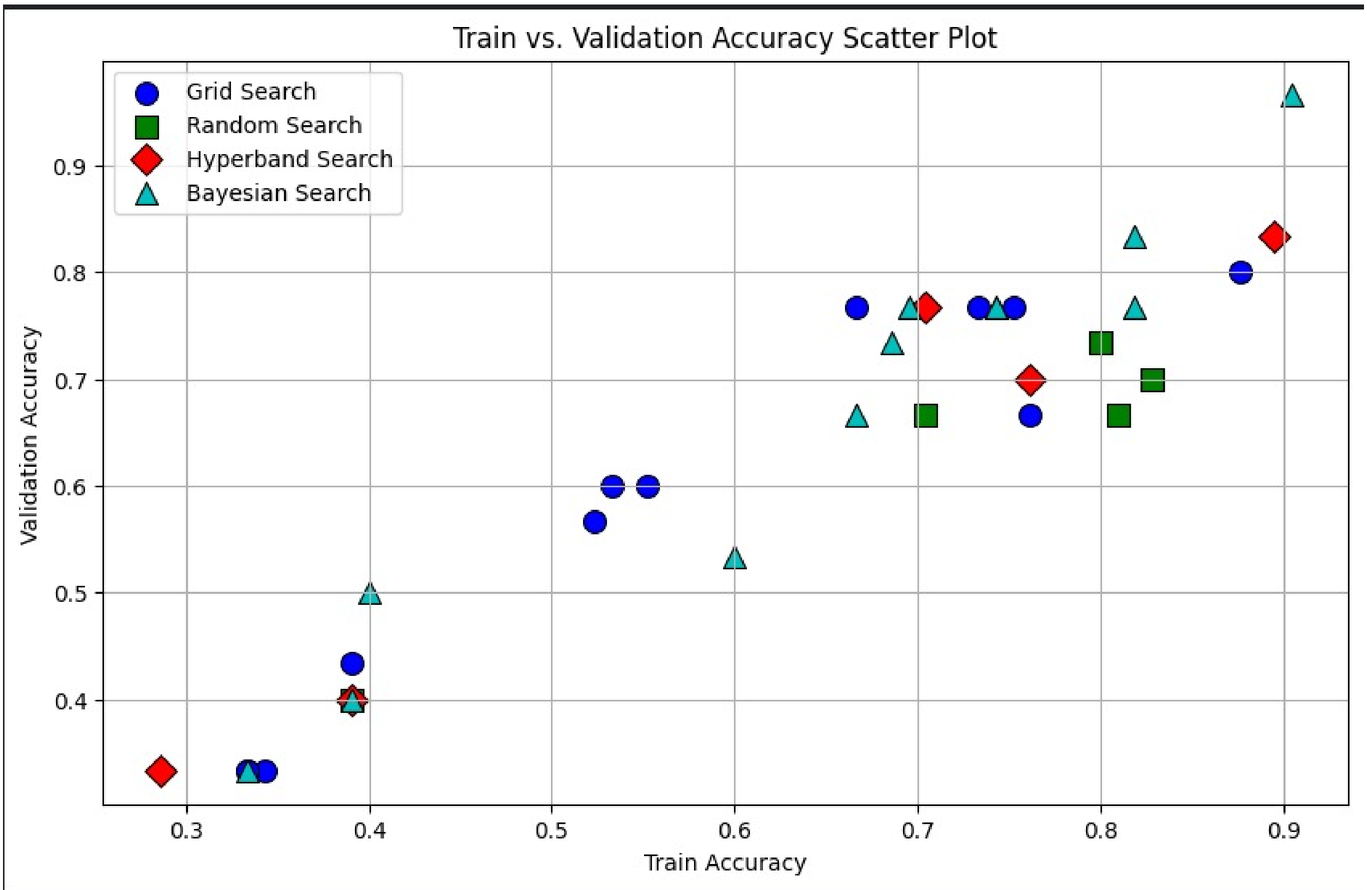
	Model Name	Batch Size	Num Epochs	Learning Rate	Train Accuracy	Validation Accuracy
0	NeuralNetTorch/6e128_00000	2	1	0.000010	0.390476	0.400000
1	NeuralNetTorch/6e128_00001	2	7	0.000743	0.895238	0.833333
2	NeuralNetTorch/6e128_00002	3	9	0.000905	0.704762	0.766667
3	NeuralNetTorch/6e128_00003	4	7	0.000339	0.761905	0.700000
4	NeuralNetTorch/6e128_00004	2	1	0.000667	0.285714	0.333333

Best accuracies and F1 scores for each strat

Search Method	Accuracy	F1 Score
Grid Search	0.7333	0.6755

Search Method	Accuracy	F1 Score
Random Search	0.7333	0.7070
Hyperband Search	0.8333	0.8295
Bayesian Search	0.9667	0.9666

Plot the scatter plot for training vs validation loss.



Manual tuning vs. Automated search

- Manual tuning is better when you have deep domain knowledge and a small number of hyperparameters, as it allows for intuitive, experience-driven adjustments.
- Automatic tuning (e.g., grid search, random search, Bayesian optimization) is better for larger, complex spaces, as it systematically explores combinations, saves time, and reduces human bias, making it more efficient and scalable for optimizing performance. With respect to the different Automatic search strategies:
  - Grid search is better for small hyperparameter spaces because it exhaustively searches all combinations, ensuring the best set is found.
  - Random search is better for larger spaces because it samples randomly and often finds good hyperparameters faster with fewer iterations, as it doesn't waste time on poor combinations.
  - For very large spaces, Bayesian optimization is superior as it uses past evaluations to focus on promising regions, balancing exploration and exploitation efficiently.