Week 2 Summary Report

Vidit Kumar

May 2025

1 Exploratory Data Analysis

To plot and observe the data, I have performed the following transformations:

1. **Dropped classes:** Columns such as `PatientId` and `AppointmentID` have been dropped.
2. **Label Encoding:** Features like `AppointmentDay`, `ScheduledDay`, `Gender`, and `No-show` have been label encoded.
3. **Scaling:** After experimenting with `StandardScalar`, `RobustScalar`, and `MinMaxScalar`. The scores for models trained using Robust and MinMax Scalars were pretty comparable. So, I used `RobustScalar` for my final model.

and the data has been split using train test split with 80 percent being in the training set and 20 percent being in the test set which was then made into mini-batches of using `DataLoader` provided by Pytorch

2 Model 1

This model was made using `NumPy` it uses stochastic gradient descent optimized using Adam. The weights and biases have been initialized randomly without using any initialization techniques. Hyper-parameters used for this model are given below:

- **Learning Rate:** 1e-5
- **Beta 1:**0.9
- **Beta 2:**0.999
- **Epsilon:**1e-8
- **Epochs:**20

Layer No.	**Dimensions (Input $\rightarrow$ Output)**	**Activation** height1
$11 \rightarrow 64$	ReLU	
2	$64 \rightarrow 16$	ReLU
3	$16 \rightarrow 2$	Softmax

Table 1: Architecture of `my ffnn`

3 Model 2

This model is written using `Pytorch`, the weights and biases have been initialized using `He initialization`. This model also uses gradient descent optimized using Adam with weight decay, it also uses mini-batches to process data with a weight bias for the model.

- **Learning Rate:** 1e-3
- **Weight Decay:**0.9
- **Alpha:**1.5
- **Gamma:**2
- **Epsilon:**1e-8
- **Epochs:**20

Layer No.	**Dimensions (Input $\rightarrow$ Output)**	**Activation**	**Dropout**
1	$11 \rightarrow 32$	ReLU	Dropout(p=0.5)
2	$32 \rightarrow 8$	ReLU	Dropout(p=0.3)
3	$8 \rightarrow 2$	Softmax	Dropout(p=0)

Table 2: Architecture of `ffnn`

4 Conclusion

I also experimented with deeper neural networks but this architecture outperformed them. `Model 1` was not very experimented with. But in `model 2` I tried experimenting with Custom Loss and Focal loss, loss matrix was found to be:

$$\begin{bmatrix} 0 & 2.5 \\ 10 & 0 \end{bmatrix}$$

However, they were not very helpful either. I also experimented with different activations. And I am unable to replicate the results for `Model 1`.

5 Evaluation Metrices

5.1 Convergence Times

Model 2 converged faster because it used mini-batch gradient descent along with the highly optimized back-end provided by Pytorch. Both them were trained on CPU locally.

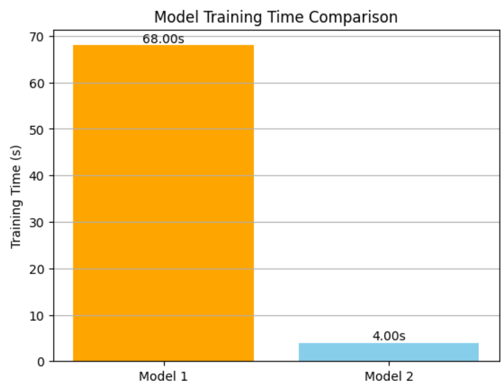

Figure 1: A plot for convergence times of models

5.2 F1 Scores

I used the weighted F1 score with sample weights as a tensor of fractions of the respective class, as it accounts for class imbalance, according to this specific type of imbalance.

$$\text{Weighted\_F1} = \frac{\sum_{i=1}^{C} \text{Support}_i \cdot \text{F1}_i}{\sum_{i=1}^{C} \text{Support}_i} \quad \text{where} \quad \text{F1}_i = 2 \cdot \frac{\text{Precision}_i \cdot \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i}$$

5.3 PR AUC

This integral given below is used to calculate `PR AUC`

$$\text{PR AUC} = \int_0^1 P(r)\,dr$$

This approximation is calculated instead.

$$\text{PR AUC} = \sum_{i=1}^{n-1} (r_{i+1} - r_i) \cdot P_{i+1}$$

weighted PR AUC was calculated with sample weights as a tensor of fractions of the respective class.

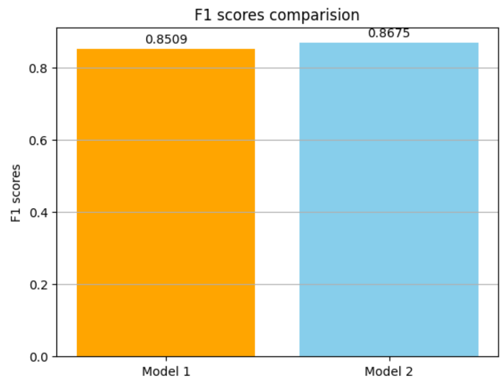

Figure 2: A plot for F1 scores of models

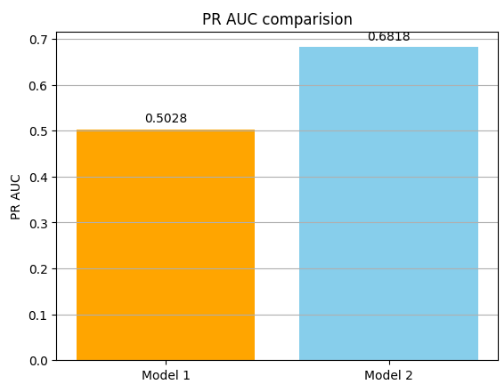

Figure 3: A plot for PR AUC of models

5.4 Confusion Matrix

It provides a tabular summary of the model's prediction outcomes compared to the actual labels.

6 Conclusions

Model 2 outperformed Model 1 because of highly optimized techniques used by `Pytorch` even with similar structure. The evaluation metrics have been manipulated according to the imbalance in the dataset. For some reason images are not rendering properly.

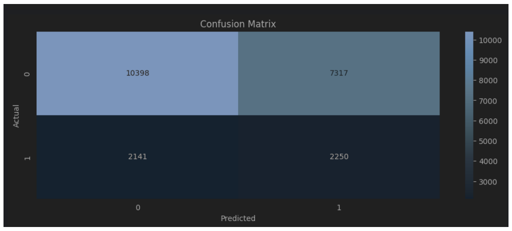

Figure 4: Confusion matrix for model 1

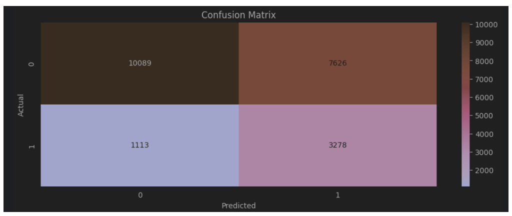

Figure 5: Confusion matrix for model 2