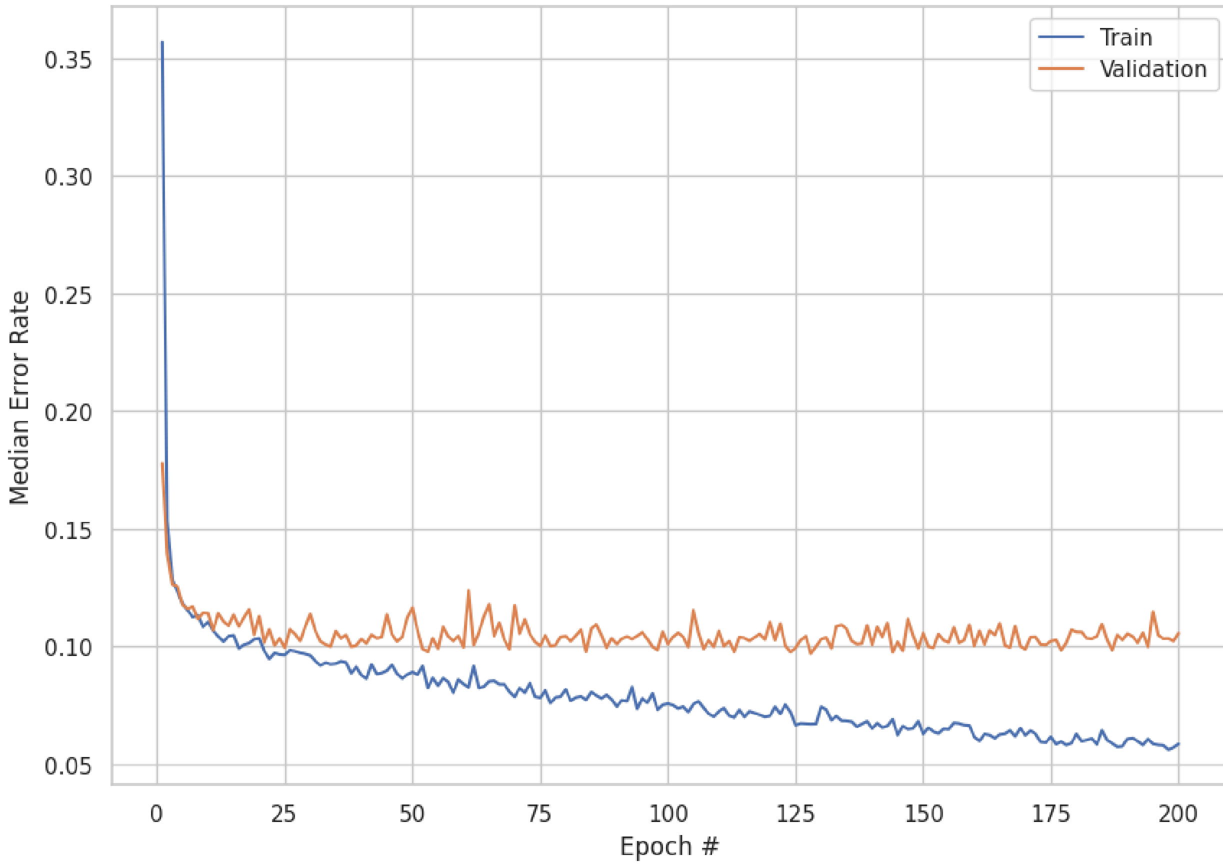
Arpita Ram Samant, Roy Zhao, Rachel Seong, Salman Shafi, John Saleeb

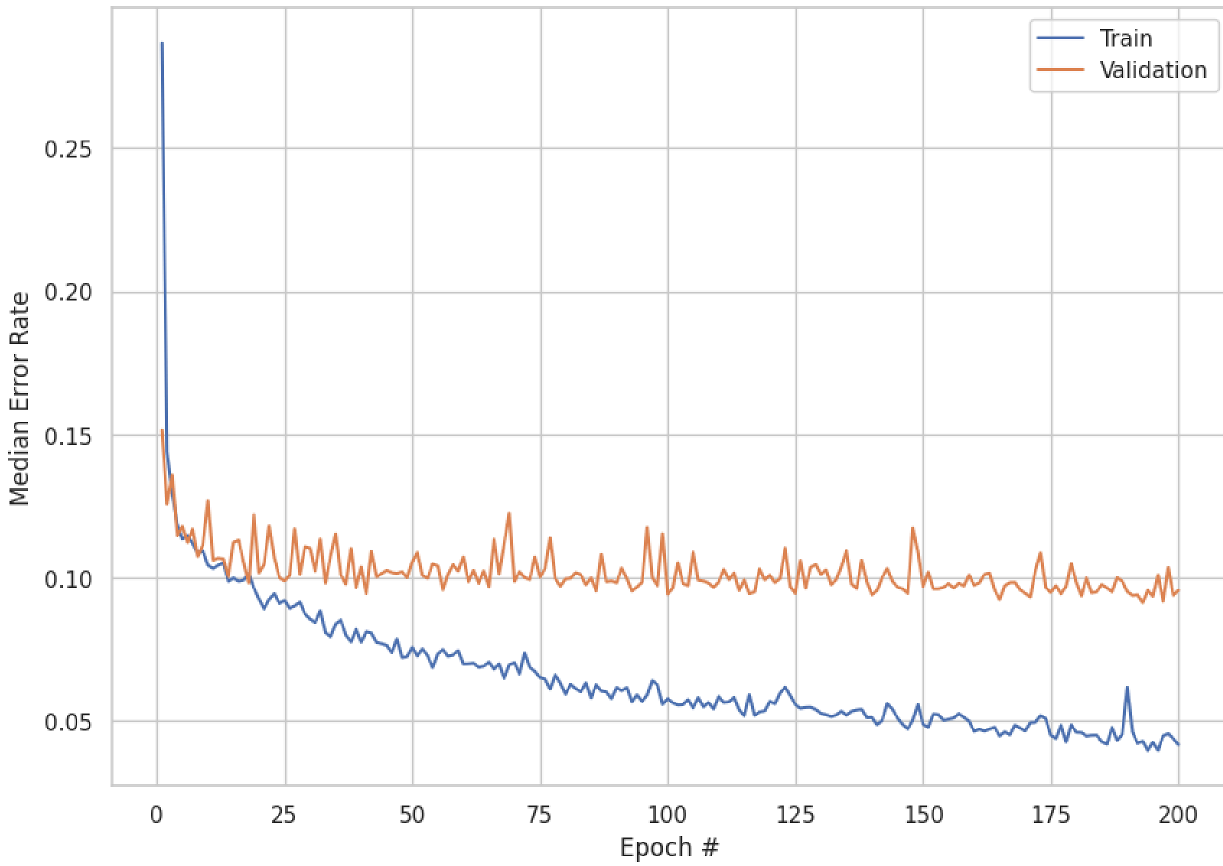
Section C Team 36 Report, Case 1 Part 2

1. A plot of the training errors and validation errors over epochs for a base multilayer perceptron model with 2 hidden layers of sizes 256 and 128.



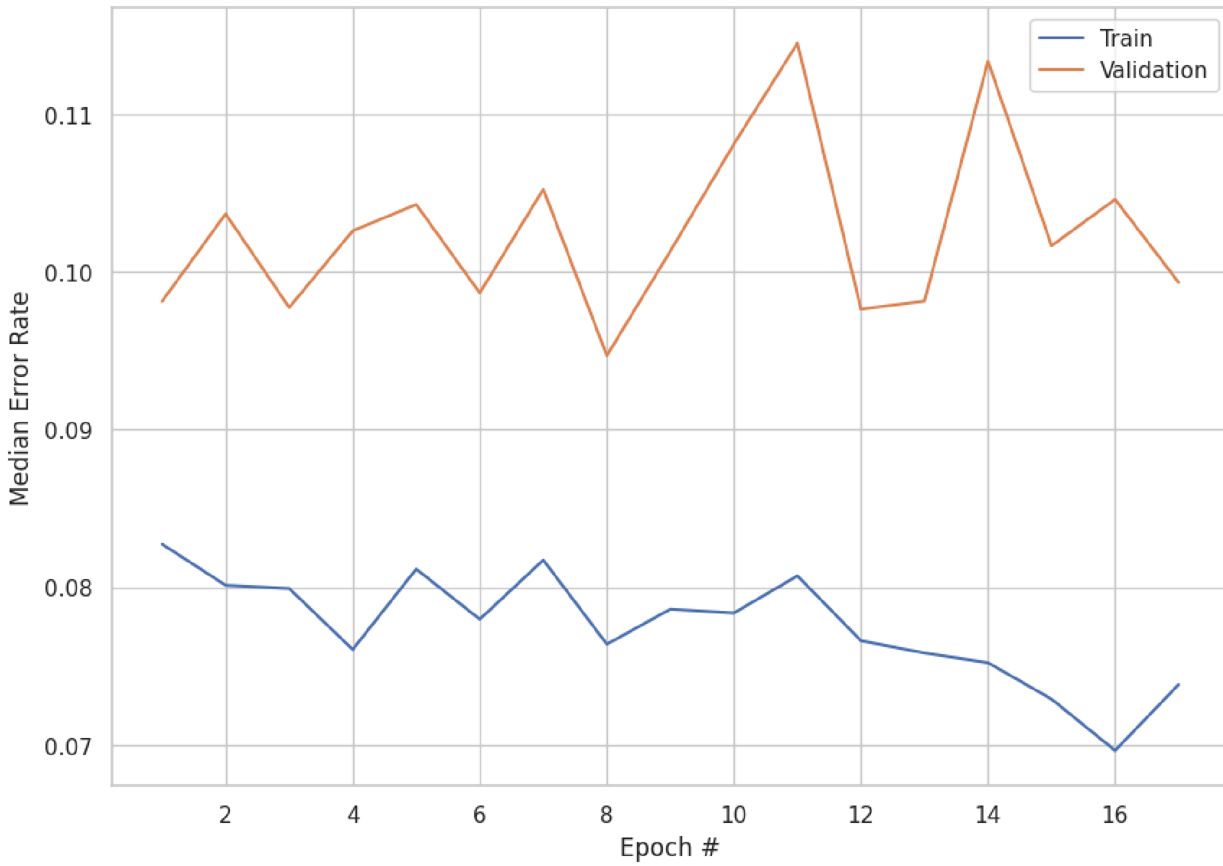
The base MLP model with two hidden layers finished with a training MER of about 0.06 and a validation MER around 0.10. Both errors drop quickly in the first few epochs, showing the model learns fast and converges well. After about 50 epochs, the curves level off, which means the model has basically hit its peak performance. There's a small but steady gap between training and validation errors, suggesting some mild overfitting, but overall the validation performance stays stable and the model generalizes reasonably well to new data.

2. A plot of the training errors and validation errors over epochs for a multilayer perceptron model with 4 hidden layers of sizes 512, 256, 128, 64.



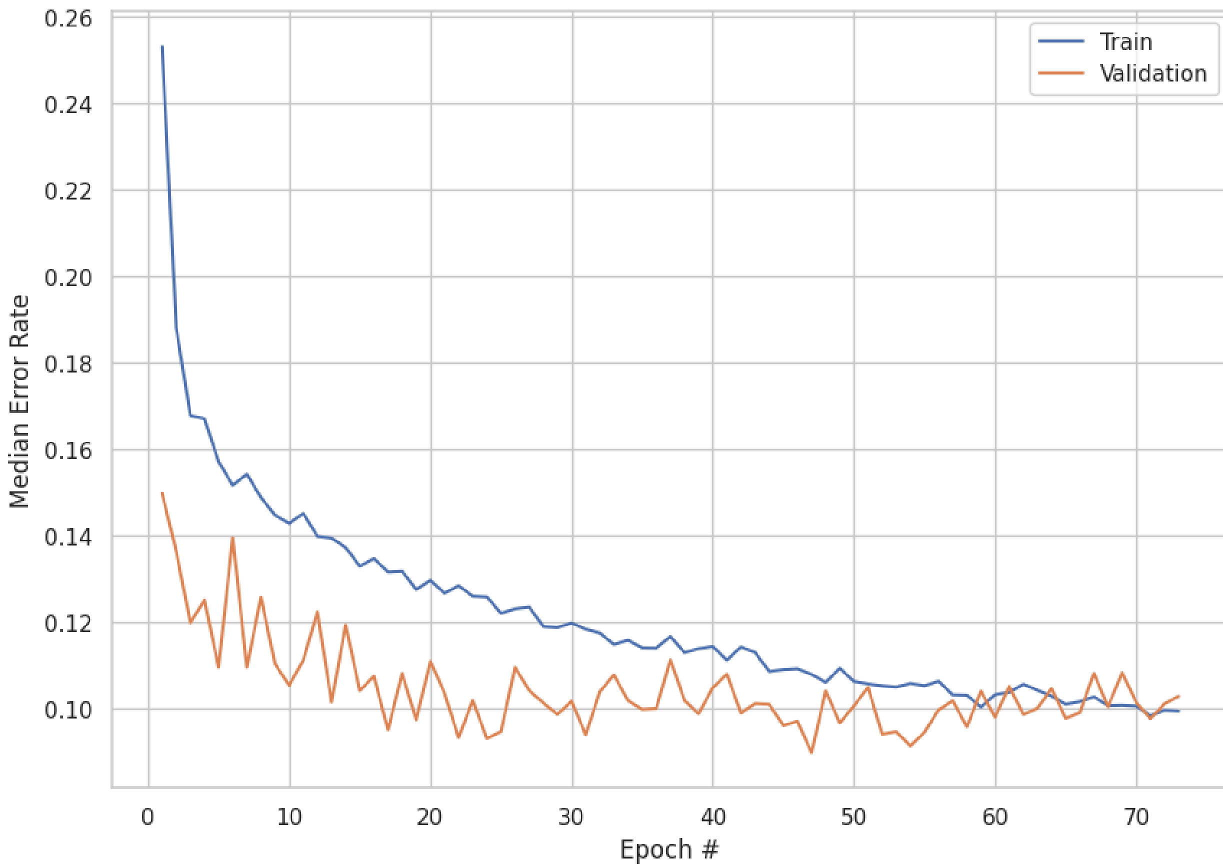
: The four-layer MLP model ended up with a training Median Error Rate(MER) of 0.06 and a validation MER of 0.10, so it's predicting house prices within roughly ±10% of their actual values. Looking at the graph, both errors drop quickly in the first few epochs and then flatten out around 50–75 epochs. This tells us the model converges efficiently and generalizes well to new data without overfitting.

3. A plot of the training errors and validation errors over epochs for a multilayer perceptron model with 4 hidden layers of sizes 512, 256, 128, 64 and norm regularization (Lasso + Early Stopping).



: The MLP with norm regularization and early stopping finished with a training MER around 0.07 and a validation MER that settled near 0.10. The training error drops steadily as the model learns, while the validation error stays relatively flat with a bit of noise. This pattern shows the model is learning well and generalizes reasonably, though there's still a small gap between training and validation errors. Overall, it looks like the norm regularization and early stopping are helping keep overfitting under control and maintaining stable validation performance.

4. A plot of the training errors and validation errors over epochs for a multilayer perceptron model with 4 hidden layers of sizes 512, 256, 128, 64 and norm regularization and dropout layers.



: The four-layer MLP with dropout and norm regularization converged smoothly, with the training MER dropping to about 0.10 and the validation MER holding steady around 0.09 with minimal fluctuations. The small gap between them and the stable validation curve suggest the model is generalizing well on this data split. That said, it'd be worth testing on a separate held-out set and running it a few more times to confirm these results hold up consistently.

5. A table listing all the model hyperparameters that you have tried with the corresponding validation errors that you found.

*other models were used(learning rate/epoch modification etc) but were not documented.

Model	**Hyper-Parameters**	**Validation Errors**
K-Fold CV	k=5, 200 epochs, patience = 10	MER = 0.1018 ± 0.0031
XGB Regression	n=200, learning_rate=0.05, depth=6, subsample/colsample = 0.8	RMSE = 115,688.426

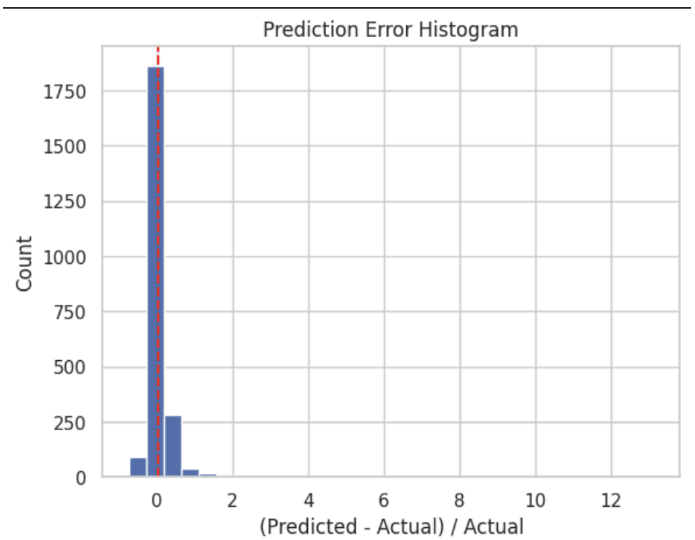
CatBoost Regression	n=1000, learning_rate=0.05, depth=6, leaf_weight = 3	RMSE = 112,467.844
AdaBoost Regression	n=200, learning_rate=0.5, depth = 4	RMSE = 224,422.200
K-Fold XGB	k=5, n=200, learning_rate=0.05, depth = 6, subsample/colsample = 0.8	(Average) RMSE = 118,877.567

: Although fine-tuning through alterations for hyper-parameters helps, what we found most helpful was to ensemble the models altogether. When our two best models (a weighted average of the regression models, XGB regression) showed best promising, we tried different variations of these two final models, which reduced the learning rate and output the best predictions.
learning reduces me

6. Your profit analysis of the iBuyer business model based on the predicted price on the valid data and answers to the four questions therein.

Question 1: what is the bias of the prediction errors? Include the histogram of prediction errors and the bias in your report.

- The mean signed error (bias) is 0.0444, indicating that the model on average overestimates house prices by about 4.4%. Not a bad performance! The histogram shows that most prediction errors are close to zero, proving the model performs reasonably well overall.



Question 2: Consider the hypothetical scenario where the offers are all accepted regardless of their values, what is the average percentage profit? Do you see a big difference compared to the profit margin α? Include your answers in the report.

- The mean profit when all offers are accepted is 0.1330 (13.3%), which is slightly higher than the target profit margin of 12%. This suggests that, on average, the model's predicted prices are slightly lower than the actual market prices.

Question 3: Based on the sale price in the valid data and the acceptance rule, what is the mean percentage profit among all accepted offers? Do you see a big difference compared to the targeted profit margin α? Include your answers in your report.

- The acceptance rate is 47.6%, and the mean profit among accepted offers is -0.0349 (−3.49%). This indicates that, under the acceptance rule, nearly half of the sellers would accept the iBuyer's offers, but the company would incur a small average loss on those transactions.

Question 4: What is the bias of the prediction errors when restricting to those properties whose owners accepted the offer? Based on the histogram and bias, can you explain your answers to Question 3?

- The mean bias among accepted offers is 0.2162 (21.62%), indicating that the model overestimates property prices by as much as 22% for those accepted quotes. Combined with the right-skewed histogram, we may conclude that iBuyer tend to offer higher prices than market value, overvaluing house prices above actual sale price, which explains the negative average profit observed in Q3.