Summary of findings for XGBoost Analysis:

The tickers chosen for the analysis were found based on a combination of sector variation and close price variation (found through Chris’s clustering). The final tickers in this list are Microsoft (Sector: Software-Infrastructure), Home Depot (Sector: Consumer), United Health Care (Sector: Healthcare), XOM (Sector: Energy), Autodesk (Sector: Software-Application), and Waters Corporation (Sector: Healthcare). Note that the last two tickers in this list, ADSK and WAT, had a very low similarity score with the other tickers. These tickers seem to have more noise in their day-to-day movements when compared to the other tickers in the analysis.

The target variable for the regression task is log return from close to next day high. This can be described as 𝑙𝑛 (next day high / current close). The reasoning behind normalizing the next-day target but current close was to allow the models to learn meaningful information from the training data. For each ticker in the S&P 500, there is a significant upward shift in prices over the last 10 years. Normalizing each 1-day high target by the current close keeps the target range-bound to log return of day to day movements, making the target easier to scale. Since the training data accurately represents the distribution of daily close-to-high movement, we are able to compute a minmax scaler using only the training data and apply it to the remaining test data.

The test-train split for the XGBoost models was 80-10-10, representing 80% training data, 10% validation data, and 10% final holdout test data. A grid search method was used to find models that had the highest R^2 on the validation data, which gave us the best parameters to use for each ticker in the analysis. The best models for each ticker were then used to compute feature importance with TreeSHAP, a game theoretic algorithm that computes Shapley Additive Explanation values for tree ensembles. The top 50 most important features per ticker were combined to reduce the data to 50 columns, which will aid in separating noisy features for downstream neural network training.

The error metrics were calculated after inverse transforming the target back into predicted next day high. These predictions were scored on RMSE, MAE, and MAPE. \*\*\*These metrics as well as the individual validation, test R^2 are saved in a dictionary. We probably need to discuss which metrics to use to evaluate our final high or close predictions, which we’ll then use to create a table for models/metric combinations.