Develop Modeling Report 1 Version 2

The model approach for this analysis is to fit a simple OLS model. Starting with an OLS model helps in variable selection as well as laying out the obstacles for more complex models. Thus, these models begin with rudimentary linear models using a selected subset of hyperparameters for the number of variables, overfitting with all of the variables, reverse variable selection, and then ridge regression. There is also some post-processing occurring, as linear models will produce an array of fitted values that are not always in the form of monetary policy actions necessary. As such, all fitted values are converted or "rounded" to its nearest monetary policy action-accepted value.

1. Simple OLS model

For this model, all of the variables in the training set were first evaluated for their p-value. This is a simplistic and straightforward test that allows for ranking of specific variables' relevance to prediction of the output variable. It does suffer, however, from the assumption of normality in the sample set, and does not consider interaction effects. Then, the top 15 variables in terms of p-value were selected for use in a simple OLS model. This hyperparameter was chosen somewhat haphazardly: doing a visual test, including the top 10 variables included a mix of macroeconomic variables, sentiment score, and also a vectorized word. This achieved at least one metric from the range of variables intended to be included in the model during variable creation.

The model is not as complex as other models to come, and is not intended to be. This will serve as a baseline model to which further models will be compared. As such, the creation of the model is simple as well, using the "Difference" variable as the dependent variable and including the resulting 15 independent variables as predictors.

Metric	Training Set	Validation Set
MSE	0.0301047	0.0885417
RMSE	0.173507	0.29756
R^2	0.252001	-0.36227
Adjusted R^2	-0.480415	-3.47603
Accuracy	0.675393	0.291667

Regarding accuracy, this model performs as well as one would think. A training accuracy of 0.675 is a good place to start, but the validation accuracy is much worse. As the larger purpose of this analysis is to be able to predict monetary policy action more often and more accurately than the market, the model will have to be much better. The adjusted R^2 being larger than 1 also raises eyebrows, but is due to the fact that the validation set is so small relative to the number of factors.

2. All variables subset

Next, the full training set was used, leveraging the swathe of vectorized words that were available as well as the macroeconomic variables that investors are likely to be monitoring as well. During variable augmentation, the hyperparameter of 100 variables was chosen as a balance between variable selection and simplicity. This resulted in 26 macroeconomic variables or augmented variables, while the rest were vectorized words. While this is a clumsy way to fit a model, the adjusted R^2 should penalize for unnecessary variables. This should provide a balance in model selection after the first OLS model.

The only complexity this model adds is dimensionality. There are a few problems with fitting too many variables. First is overfitting, where the variables start fitting to the random element rather than the underlying trend. The second is multicollinearity, where some of the predictor variables are endogenous to each other. This is certainly the case, as the LEI, CEI, and LAG indicators cover many of the other macroeconomic variables, and there are some rolling averages already included. The last is model complexity, though this is not as much a problem as 74% of the variables are just vectorized words. That said, actual utilization of a model this complex would prove difficult.

This model performs much better than the last one, with a training accuracy of 0.927 and a validation accuracy of 0.5. This is surprising, given the vastly greater number of variables. Most of the variables were not statistically significant, though, which does suggest overfitting, as well as the fact that the adjusted R^2 is higher than 1 and the validation accuracy is so different from the training accuracy. Further, a model that has few statistically significant variables and performs relatively well suggests the tradeoff between accuracy and model simplicity.

Metric	Training Set	Validation Set
MSE	0.00458115	0.03125
RMSE	0.0676842	0.176777
R^2	0.886174	0.519199
Adjusted R^2	0.772348	1.15359
Accuracy	0.926702	0.5

3. Reverse Variable Selection

Starting from a model that includes all of the variables, it make sense to perform reverse variable selection to cut down the number of variables and increase ease of use. The benefit of these models is that it iterates the significance of each variable until no further variables are eligible for removal. It results in a smaller set of variables that hopefully all have statistical significance—in this case, the p-value.

It also benefits from retaining model simplicity in that it is still an OLS model, just with better-chosen variables than hand-picking them. To that end, replicability becomes a strength of these models relative to using every single variable and overfitting.

The model performs admirably compared to the previous model in that it has a training accuracy of 0.864 and a validation accuracy of 0.625. This balance compared to the previous model does suggest there is some balancing of bias occurring, as the model is stronger in generalized terms, but weaker in the training set. To that end, it contains 7 macroeconomic variables, 4 Minutes-derives variables (Net Sentiment Score, Positive Frequency, Negative Frequency), and 16 vectorized words. Further, it includes macroeconomic variables that sync with textbook theories of monetary policy action in that it includes the “Level”, “Average Hourly Earnings”, and “LEI” variables.

The benefit of reverse variable selection is that you can also filter by the hyperparameter AICc. This is a metric that calculates goodness of fit, but also penalizes models with many parameters relative to the sample size. Because the sample size is relatively small for a dataset, the AICc is useful as it provides a more accurate measure of model quality.

Metric	Training Set	Validation Set
MSE	0.00948953	0.03125
RMSE	0.0974142	0.176777
R^2	0.764218	0.519199
Adjusted R^2	0.723465	3.21169
Accuracy	0.863874	0.625

This model, by far, produces the best training and validation accuracy. It also does this with only 27 features, vastly reducing the dimensionality relative to the second model, which used 191 features. Interestingly, the RMSE is higher than the previous model but has a higher accuracy, suggesting that even including the rounding, the fitted values were more incorrect in degree than the previous model. Of the three models, this is still the best model.