Analyzing and Forecasting U.S. Immigration Trends

Group 7: Alan Lin , Trevor Petrin , Ganesh Kumar Ramar , Mingyu Chen

Repository: alanklin/AA-Capstone: Boston College Applied Analytics Capstone Project

This week we focused on the model selection process for forecasting immigration at the United States borders using US Customs and Border Patrol (USBCP) open source data. Our main focus is to understand the performance of the model before applying it to the test data. As for how the structure of our code varies versus that of the assignment, we will go into more detail in the individual sections because of our time series application versus the other groups' classification or regression tasks.

To begin, we must examine the models' performances before they are applied to the test set. As for why there is not a traditional validation set as in the classification or regression problems, this is strictly to do with the temporal nature of the time series data. If the model is validated on non sequential points, it will provide no value to the predictive power of the model.

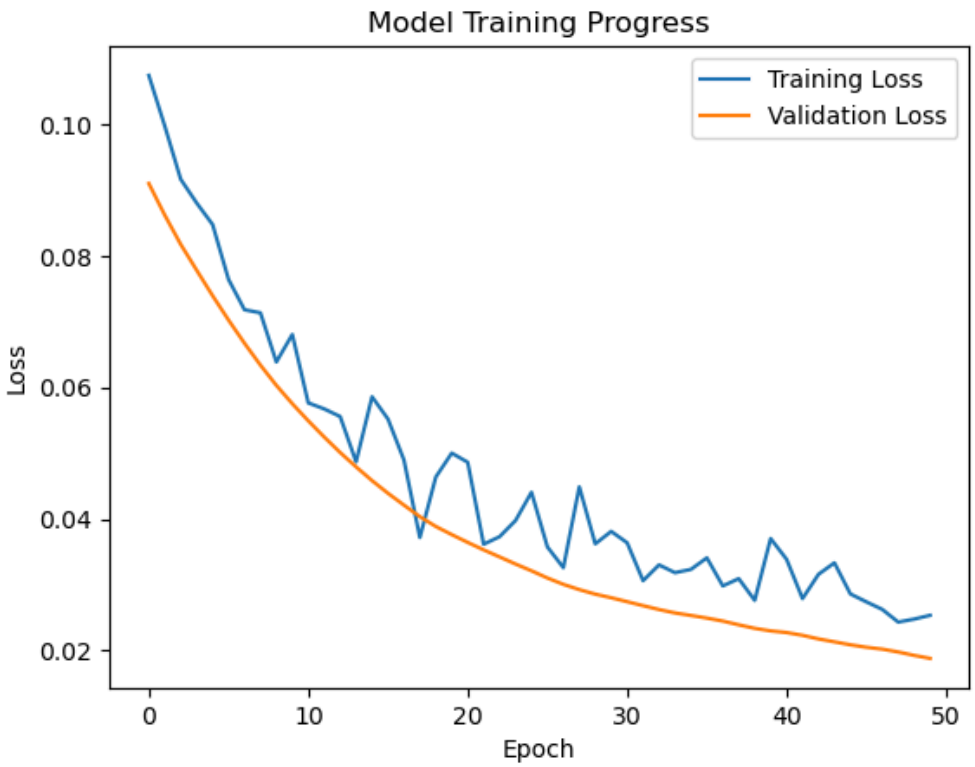
However, we do need to apply a validation set to the Machine Learning models to tune them up before their application to the test set. This was done using windowized data such that there were the same number of points predicted in the model output as there are in the test set. In other words, we used n inputs from time 0 to n-1 to predict the data at time n to n+5. This approach trains the models to predict the next several points without having spillage from the train into the test set. As for separating out a validation set, input/output combinations were pulled at random to create an 80/20 split of input-output combinations.

As for our fatal flaw with this approach, though there would not be any spillage and is somewhat close to the Hyndman textbook approach, we should have only used one step ahead forecasting for several reasons. These would include being able to have more data and for a shorter term, more individualized point forecast as is seen in the way ARIMA and ETS recursively predict their points. If we had the opportunity to go back and reimplement the validation set in this fashion, we would.

This approach will be applied to the Machine Learning approaches for their tuning process as the more simple models ARIMA and ETS would not benefit from a validation split in this instance. Mathematically, this is because ARIMA has an assumption of being stationary. The property of a time series being stationary implies that the time series has a consistent mean, variance, and autocorrelation and is sought after in the ARIMA model using the differencing parameter (d). As for ETS, the model updates through every newly observed value, eliminating the need to fine-tune parameters while the model is deployed, instead only needing to worry about the relation of the parameters either being additive, multiplicative, or in some cases outright not included.

In our defense of performing no true validation set, none of the literature we could find aside from the Hyndman gave an example where a validation set was recommended, and at that even Hyndman in all other examples modeled straight on the test data.

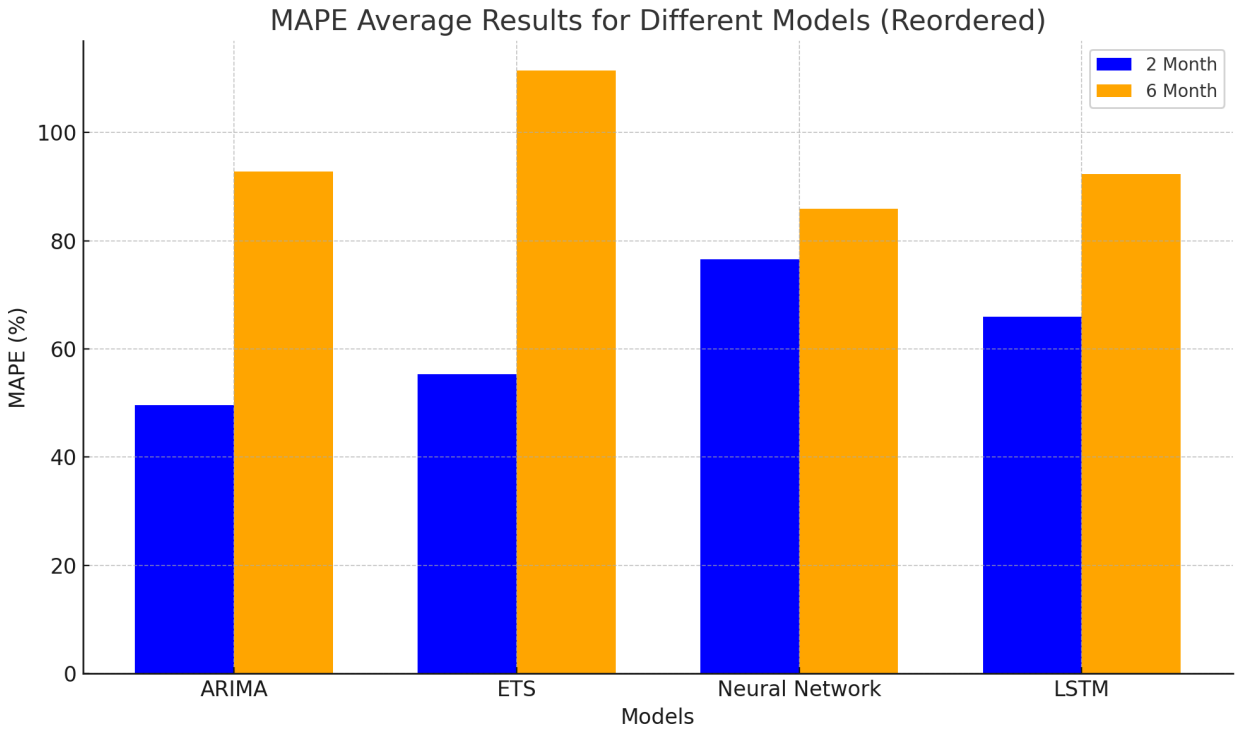
For displaying the validation data, the reality is we have 41 sectors and at least 20 individual models per sector, so it would be unrealistic to graph all of the data here, so for the model validation process, we shall use the validation loss of the Big Bend Sector to illustrate our process.



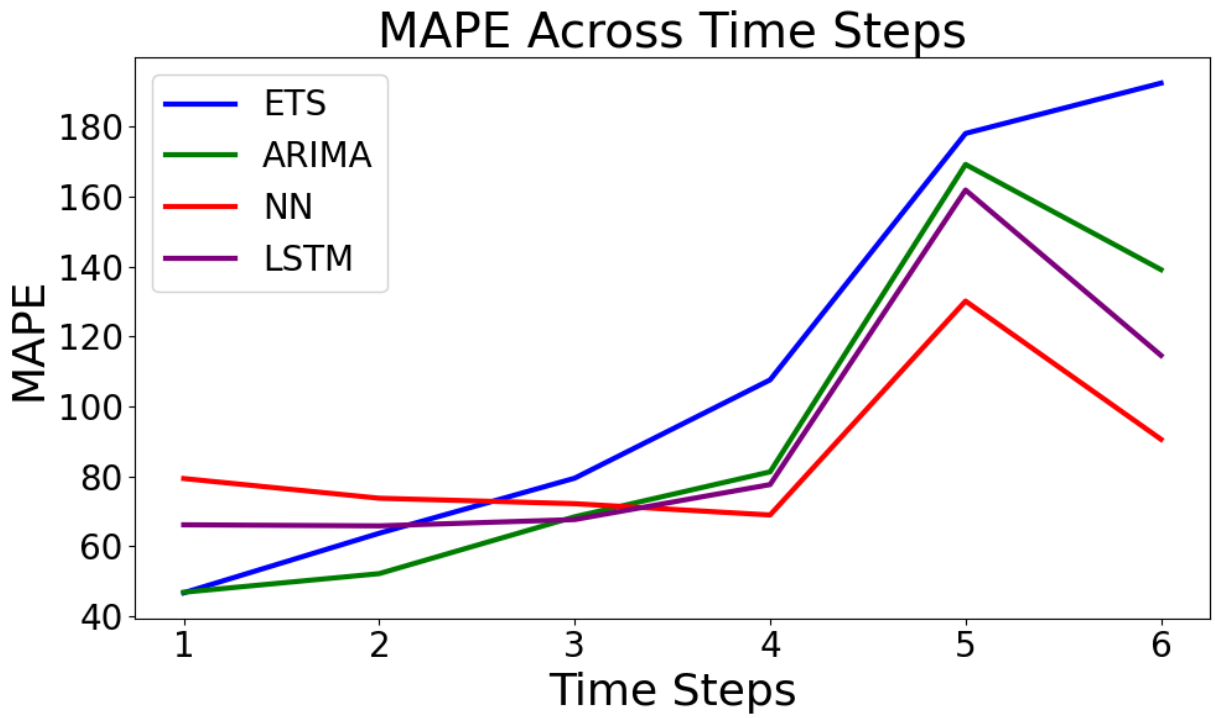
As can be seen from the model progress graph, our validation in tuning went about as smoothly as it could have gone for the Neural Network hyperparameter tuning, taking a smooth descent with the validation loss, though the training loss was slightly choppy. The winning model for every Machine Learning model was different for each sector, this illustrates our overall general process for selection.

As we have covered extensively in the past two weeks, ARIMA recorded the lowest average MAPE (across all sectors) at just under 50% when forecasting two months ahead. ARIMA, NN, and LSTM were comparable in the long-term, but we recognize that even two months of foresight may be incredibly impactful to USCBP already. For the final winning model, we decided that ARIMA was our best bet considering its exceptional short-term performance as well as explainability.

For the bias-variance tradeoff of the models, it is extremely straightforward here that the more complex models were overfitting on the training data whereas the more simple models were generalizing better using the more recent points more effectively.



We wanted to separate out the performance of the models at both the 2 and 6 month intervals to emphasize the models' overall better performance short-term before dipping in the long run. In this case, the simpler models performed better in part because of implementation, but mostly because it was able to pick up on the trends in the data better in the short term. The ARIMA and ETS well outperformed the Neural Network and LSTM in terms of Mean Absolute Percentage Error (MAPE) through the first two months before losing their ground around month 5. As was mentioned in the document last week, this was because of the re-election of President Trump whose hardline immigration policies and rhetoric deter immigrants from seeking refuge in the United States. In the long run, though, the Neural Network and LSTM were on par with or slightly outperformed the ARIMA model because they both were better able to account for the drop across all models, and their peaks in month 5 were much lower than the more basic methods across all models.



For the test dataset metrics, we used MAPE because of the variability in scale across all sectors. Some sectors on the Southern Border were encountering between 50-80,000 per month during the peak months of immigration whereas some sectors along the Northern Border were encountering less than 500 in those same months. The scale of the error clearly matters here rendering another metric such as RMSE or MAE effectively useless in this situation. As for debating whether the model is 'good enough', given how naive our models are in terms of input data, we found the ARIMA's performance of a MAPE below 50% in the first 2 months of predictions across all models as a solid performance.

MAPE Average Results			
	Train	2 Month	6 Month
ARIMA	**53.7%**	**49.6%**	**92.8%**

The ARIMA overall generalized very well to the test data within a short time frame of predictions before beginning to fall off around month 5 of the test set. These results suggest that the model is not overfitted on the train data and that the lack of performance in later months could not have been accounted for in the model.

Overall, we are extremely satisfied with the results of our model and look forward to future discussion on risk, ethics, bias, and packaging it all up.