Analyzing and Forecasting U.S. Immigration Trends

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

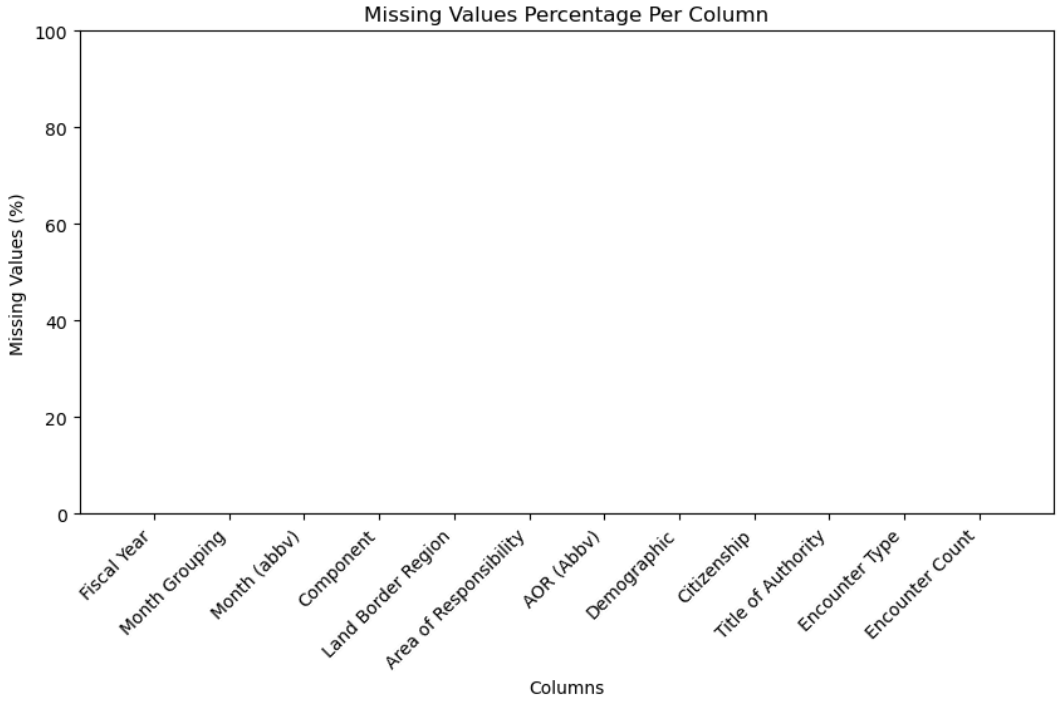
Repository: alanklin/AA-Capstone: Boston College Applied Analytics Capstone Project

For Week 4, we focused on data processing and preparation for model building. Based on our findings last week, we aren't facing common issues such as missing values, so thankfully we don't have to consider any imputation methods for this dataset. We considered combining both datasets together but it was impossible because although the Encounter Count summed up to be equivalent in both datasets, there was no primary key pairing where the data could be merged. Because the State and Sector borders do not align with some states partially covered by different borders and several states being covered by multiple regions, there is no way to determine which encounters were in which State/Sector pairing. Going forward, because the Sector dataset is more relevant to the US Customs and Border Patrol (USCBP) funding and resource allocation, we will use that rather than the State-level data.

The first pre-processing step that we've taken is to remove unnecessary columns, such as Fiscal Year, Month Grouping, Month (abbv), and AOR (abbv). After implementing a function last week to convert the Fiscal Year and Month into a corresponding Year-Date variable, we don't have to include the variables that went into this feature-engineered variable. We don't believe Month Grouping will provide any information for our forecasting, as this was just a variable that the USCBP used to conveniently analyze their own data. The abbreviated AOR (Area of Responsibility) variable can be removed because it provides no additional information as we already have the Area of Responsibility variable that is arguably easier to read and understand.

Other preprocessing steps that were taken to ready data for future modeling including parallel-time-series LSTM included some feature creation using the proportions of variables observed at each aggregated time step. Such features were incorporated because of their value to USCBP or because of their significance seen from the EDA phase. The first variable created was the number of encounters whose country of citizenship was from Mexico because it is the largest country of origin. Forecasting the number of individuals from here could provide a general trend to whether the migration is increasing or decreasing. This was done via a similar process of the

total aggregation, just instead of aggregating the complete data, only summing the data whose country of origin was Mexico.



Despite the total number of missing data in the original dataset being 0 (above), when the data is aggregated, this can cause some missing values by omission. For example, say Big Bend Sector had no encounters in March 2020, this would need to be filled using an NA value, which was accounted for in the code.

Next, the number of unaccompanied minors was aggregated, stemming from the reason that the demographic is a hot-button issue for the media and human rights groups alike. Having a forecast of the number of children coming into the country unaccompanied will give deeper insight into how to have the proper facilities and funding such that a humanitarian catastrophe does not emerge with them. Additionally, in the court of public opinion, the way children are handled may also negatively impact public perception of other border enforcement agencies such as ICE, who are already facing public backlash. The last thing USCBP needs to do is make their lives harder as well.

For structuring the data in a way such that it can be input into a model, we needed to aggregate the data by month and border region to be able to add value to the data without going too granular by over-aggregating. It was a relatively straightforward process, just using the python groupby() function to achieve the output pictured down below.

Area of Responsibility	Year-Date	Encounter Count
Atlanta Field Office	2019-10-01	1022
Atlanta Field Office	2019-11-01	762
Atlanta Field Office	2019-12-01	564
Atlanta Field Office	2020-01-01	527
Atlanta Field Office	2020-02-01	521

We can add on those feature engineered variables as described above - Encounters from Mexico and Encounter Count of Unaccompanied/Single Minors. Targeting high value points of interest may provide more insight as we move forward in the model building process with parallel LSTMs.

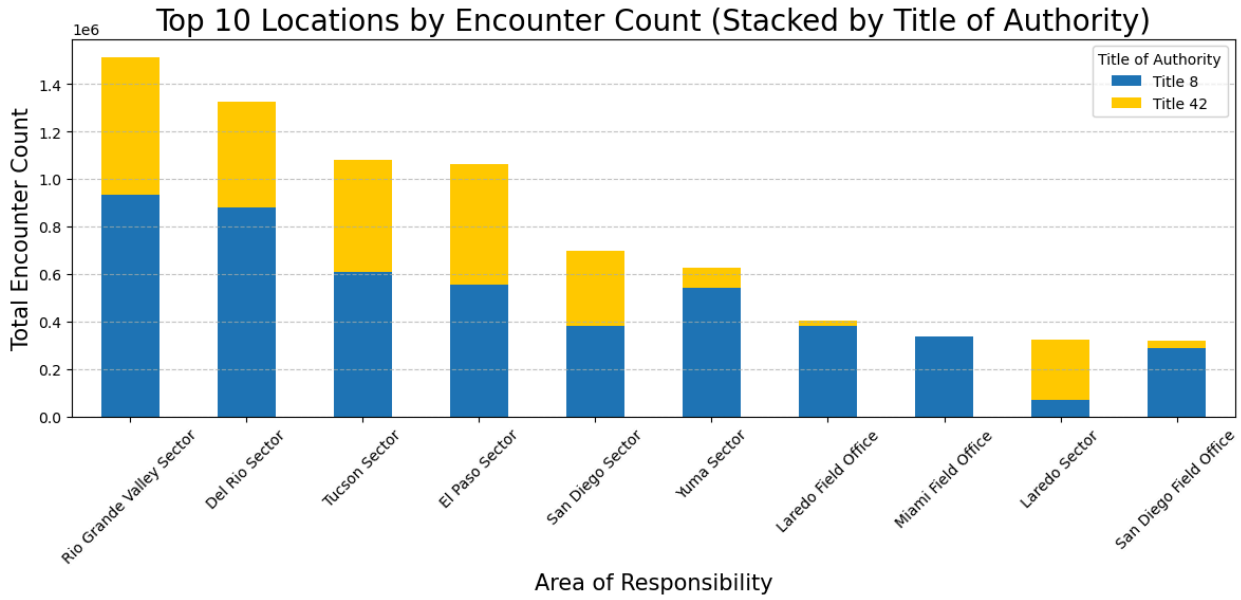
	Area of Responsibility	Year-Date	Encounter Count	Encounter Count Mexico	Encounter Count UM
0	Atlanta Field Office	2019-10-01	1022	25.0	0.0
1	Atlanta Field Office	2019-11-01	762	24.0	0.0
2	Atlanta Field Office	2019-12-01	564	22.0	0.0
3	Atlanta Field Office	2020-01-01	527	19.0	3.0
4	Atlanta Field Office	2020-02-01	521	43.0	0.0

However, this is not the best format for creating individual tensors by Area-of-Responsibility, thus the pivot() function was used to create a matrix such that the rows were the Area-of-Responsibility, the columns were the Year-Date, and the Encounter count was the values. This approach was taken for the standard time series portion for the initial model building, and will need to be expanded to include a 1x3 matrix for the parallel time series forecasting approach. Expanding the dataset to do this can be done in future weeks, with additional models being built to handle the different-shaped tensors.

Year-Date	2019-10-01	2019-11-01	2019-12-01	2020-01-01	2020-02-01	2020-03-01	2020-04-01	2020-05-01	2020-06-01	2020-07-01	...	2023-03-01	2023-04-01	2023-05-01
Area of Responsibility														
Atlanta Field Office	1022.0	762.0	564.0	527.0	521.0	847.0	1037.0	2141.0	1532.0	1428.0	...	1102.0	1206.0	1260.0
Baltimore Field Office	443.0	550.0	658.0	698.0	660.0	367.0	85.0	177.0	150.0	116.0	...	1599.0	1614.0	1764.0
Big Bend Sector	653.0	526.0	543.0	604.0	562.0	675.0	507.0	628.0	660.0	746.0	...	1200.0	1181.0	1421.0
Blaine Sector	32.0	24.0	36.0	25.0	27.0	16.0	6.0	13.0	5.0	15.0	...	110.0	174.0	115.0
Boston Field Office	1395.0	1001.0	1558.0	950.0	1350.0	611.0	292.0	54.0	279.0	208.0	...	4173.0	3115.0	3553.0

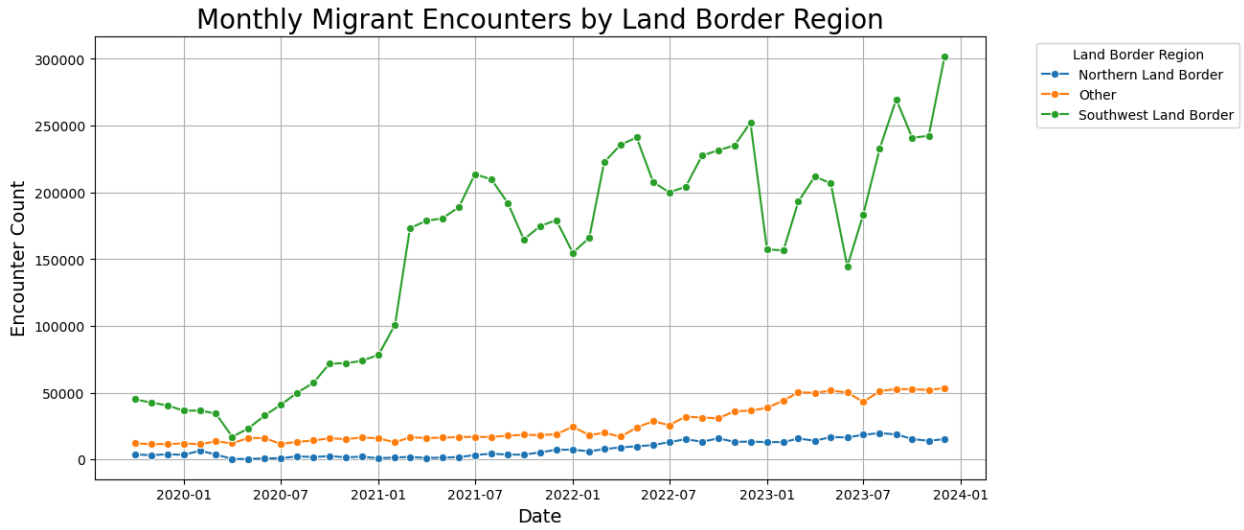
5 rows × 51 columns

As for the input, it will now be a 1x51 tensor used to forecast a 1x12 tensor corresponding to the calendar 2024 year. A pipeline was used to replicate this process on the test data, giving the 1x12 test set. For the corresponding model building process, there will need to be a list of models created, one corresponding for every row of the data frame such that performance is maximized across the sectors. It would not make sense to train/validate a singular model across multiple sectors because of the uncommon factors surrounding them.



For example, there will be different push and pull factors for individuals coming to the Southern border in sectors such as Big Bend will have different push factors such as economic uncertainty versus those coming from the Northern border. Even along the Southern border where there are a plethora of different regions, there may still be differences in the proportions of migrants from

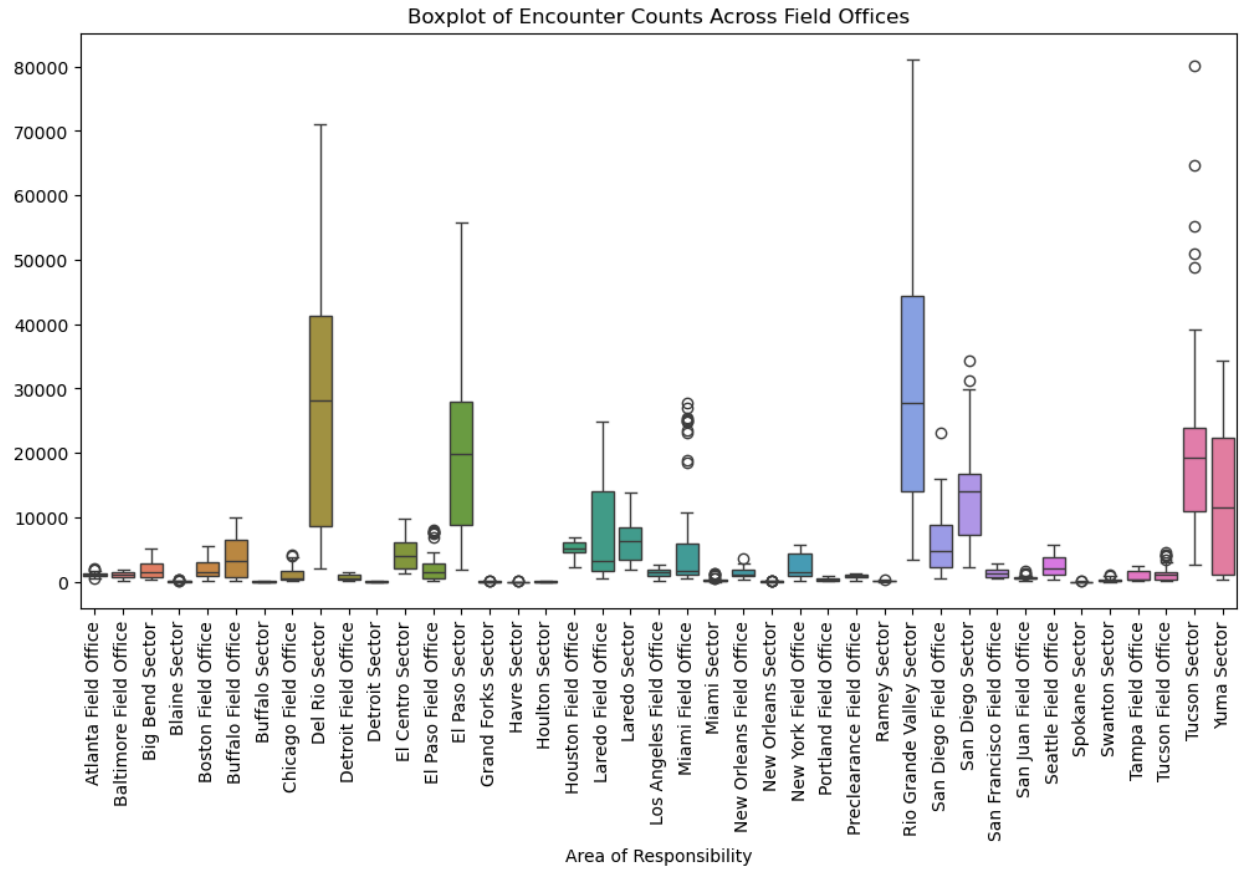
different countries, leading to differences in the patterns there. For example, there was a disproportionate number of Title 42 expulsions used on the Southern border, it would not be reasonable to extend that to the Northern border. The same goes for trying to predict Field Offices versus Border Sectors as can be seen in the figure above.



The bottom line for the separate model building is that for this data it is not acceptable to create a model at the generalized level, to best predict the trends this must be specific here. The second problem coming from using a generalized model is the scaling of the data points. It would not be practical to have so many values so close to 0 when the Northern border would be scaled with the same range as the Southern border, which would result in the model having training issues. As can be seen in the graph above, the sheer magnitude of the Southern border dwarfs that of the northern border and other entry points.

To determine the type of scaler we will be using, we found it helpful to visualize the distribution of the encounter counts across each AOR. Initially, we considered StandardScaler but after looking at the distribution shapes, it might not be a safe assumption that Encounter Count is a Normally distributed target variable. We also cannot consider large encounter counts to be outliers as this is valuable information that should be in the model - if a Sector/AOR has faced high volume before, then it remains a possibility in the future. Predicting these spikes occurring may be one of the most important end goals of our model. We decided to preserve the original distribution by using a MinMaxScaler for each individual AOR. It wouldn't be right to use a

single MinMaxScaler to fit on the entire dataset, as the scale of encounter counts at each AOR is clearly unique.



As for the implementation of the code for how this will be handled, there will be a Dataframe containing the name of the Area of Responsibility, followed by two columns, one with the min max scaler to be used on the input data, and the other containing the model for which the sector data was used to train. The second column may need to either be expanded to include an array of models or add separate columns for the different models that the sector was used to train. As for the positioning of the scaler object, they were created in the week 4 notebook, however, a new global variables file, _Vars.py will be used to carry the scalers between notebooks. The alternative to this approach would be either saving the scalers as a different data type or redeclaring them in every notebook, neither option would be convenient. Granted, it is not necessarily good practice either to have global variables, but at this point, the data is finalized and the manner in which it is scaled will not be changed.

As an aside, we also considered aggregating Encounter Count while preserving vital information such as Demographic, Citizenship, Title of Authority, etc. To better understand our goal, an example is provided below.

Example Input

Year-Date	Area of Responsibility	Demographic	Citizenship	Title of Authority	Encounter Type	Component	Land Border Region	Encounter Count
2019-10-01	Boston Field Office	FMUA	BRAZIL	Title 8	Inadmissibles	Office of Field Operations	Northern Land Border	2
2019-10-01	Boston Field Office	FMUA	BRAZIL	Title 8	Inadmissibles	Office of Field Operations	Northern Land Border	3
2019-10-01	Boston Field Office	Single Adults	CANADA	Title 8	Inadmissibles	Office of Field Operations	Northern Land Border	1031

Output (After Aggregation)

Year-Date	Area of Responsibility	Demographic	Citizenship	Title of Authority	Encounter Type	Component	Land Border Region	Encounter Count
2019-10-01	Boston Field Office	FMUA	BRAZIL	Title 8	Inadmissibles	Office of Field Operations	Northern Land Border	5
2019-10-01	Boston Field Office	Single Adults	CANADA	Title 8	Inadmissibles	Office of Field Operations	Northern Land Border	1031

In the example, there are two rows for the Boston Field Office where the value for each column matches up. Instead of having two separate entries that hold incredibly similar information - barring the Encounter Count - we can aggregate the two to capture the total encounter count in each month for each unique set of variables. I tested the aggregation code on a separate dataset and confirmed that it works as intended before applying it to our Sector dataset. It turned out that each row in the dataset was already unique, as the number of rows after aggregation did not change, implying that this aggregation technique was already performed by the USCBP.

This was extended to the entire dataset, with each month and sector having their total encounters aggregated to provide a more granular explanation of the overall encounter trends.