GDP Growth Rate Regression

Introduction

The goal of this project was to evaluate the ability to predict the GDP growth rate based on transportation metrics from the U.S. Department of Transportation (DOT) Bureau of Transportation Statistics (BTS). This project is a follow-on to a project to create a classifier that uses this same data to simply predict if the GDP is increasing or decreasing.

The source code for this project is accessible at:

https://github.com/schkotty/Data-Science/blob/master/GDP-Predictor/GDP-Predictor-NeuralNetRegression.ipynb

Freight transportation is impacted by many factors that impact GDP such as:

- Consumption of durable and nondurable goods
- Business capital investment such as spending on plants and equipment
- Government expenditures
- Exports
- Imports

Gross Domestic Product (GDP) is the monetary value of all goods and services produced by a country's borders in a specific time period. It functions as a broad measure of the country's overall economic health.

When GDP declines for two or more consecutive quarters, the economy is in a recession. If the GDP growth rate becomes too high, then the Federal Reserve may attempt to slow the economy by raising interest rates.

GDP for the United States is reported by the U.S. Department of Commerce (DOC) Bureau of Economic Analysis (BEA).

Business Opportunity

With GDP being a measure of economic health, it can be used by companies to identify high-level economic trends that may impact companies in different ways. Some areas where this could be helpful include:

- Forecasting sales activity
- Forecasting purchases
- Forecasting hiring plans
- Evaluating consumer and/or company confidence in the economy

As mentioned at https://www.bea.gov/resources/learning-center/what-to-know-gdp, the U.S. Bureau of Economic Analysis estimates the U.S. GDP for each year and each quarter. The BEA estimates each GDP value three times, using additional source data and with improved accuracy. The schedule for releasing the estimates is based on the schedule below.

- 1 month after quarter end – the advanced estimate
- 2 months after quarter end – the second estimate
- 3 months after quarter end – the third estimate

This represents an opportunity for models to be used to evaluate GDP growth rates sooner than the releases from the BEA. For companies that utilize GDP growth rates as a factor to make decisions as described above, having access to information earlier could be a significant benefit in making data-based decisions faster than they would otherwise. This could in turn result in capitalizing on revenue opportunities faster or taking action to reduce expenses sooner to reduce losses.

Analytical Framework

Analytical Question

The goal of this project is to use supervised machine learning to build a regression model that predicts the GDP growth rate based on transportation freight metrics. As mentioned above, transportation freight is impacted by many activities that affect GDP. However, this small set of features may not support high levels of accuracy from the resulting model. I did this to see what level of performance is possible using this simple approach.

I also see this project as providing insights into the use of transportation freight data as a predictor for financial models in general. If this small set of features can be used to model GDP growth, then a company's transportation freight data may also be able to be used to predict financial and other results for a company.

Data Acquisition

Freight Data

Transportation data published by the U.S. Bureau of Transportation Statistics was used. The BTS data includes a "Transportation Services Index" (TSI). Per the BTS, regarding the TSI:

> The TSI is a monthly measure of the volume of services performed by the for-hire transportation sector. The index covers the activities of for-hire freight carriers, for-hire passenger carriers, and a combination of the two.

> The TSI is still under development and is therefore experimental. It is being examined for refinements in data sources, methodologies, and interpretations.

The BTS has found that the TSI tends to move before an economic change occurs, as described at:

https://datahub.transportation.gov/stories/s/TET-indicator-1/9czv-tjte#demand-for-for-hire-transportation-services

The data from the BTS includes separate monthly freight and passenger transportation values, as well as a combined total value. In this project only the freight value of the TSI index was used.

The BTS releases freight data approximately seven to eight weeks after a month ends. The TSI is released approximately five to six weeks after a month ends.

As of May 23, 2020, the BTS has not released air freight data for March 2020 yet. At this time this prevents 2020 Q1 from being included in the analysis/model.

The data used is described below, with data element names as provided by the BTS. The BTS provides both seasonally adjusted and unadjusted for all these values other than the TSI. Seasonally adjusted values were used. The BTS provides more details on the data at:

https://www.bts.dot.gov/learn-about-bts-and-our-work/statistical-methods-and-policies/technical-note-tsi-documentation

Data Element	Description
OBS_DATE	Month date value for one row of data
RAIL_FRT_CARLOADS_D11	Count of rail freight carloads transported for the month
RAIL_FRT_INTERMODAL_D11	Count of intermodal units/containers transported for the month
WATERBORNE_D11	Millions of short tons transported on internal U.S. waterways for the month
TRUCK_D11	Monthly truck tonnage index
AIR_RTMFM_D11	Ton miles of freight and mail transported by the air industry for the month
TSI	Monthly freight-only component of the Transportation Services Index as calculated by the BTS (does not include the passenger data)

The Transportation Services Index only has data going back to January of the year 2000.

GDP Data

Quarterly "Real GDP" growth rates were used, which has been adjusted for inflation. Each data point represents the change from the preceding quarter. This data is reported by the U.S. Bureau of Economic Analysis (BEA).

GDP data is available back to 1947.

Data Cleansing / Manipulation

Since the TSI index only has data back to January of the year 2000, this analysis only included data published since then.

The freight data obtained from the US Department of Transportation Bureau of Transportation Statistics is monthly data. The GDP data obtained from the US Department of Commerce Bureau of Economic Analysis is quarterly. In both cases, the data is clean. The only cleansing performed on the data was to drop any recent data where all data elements had not been published yet.

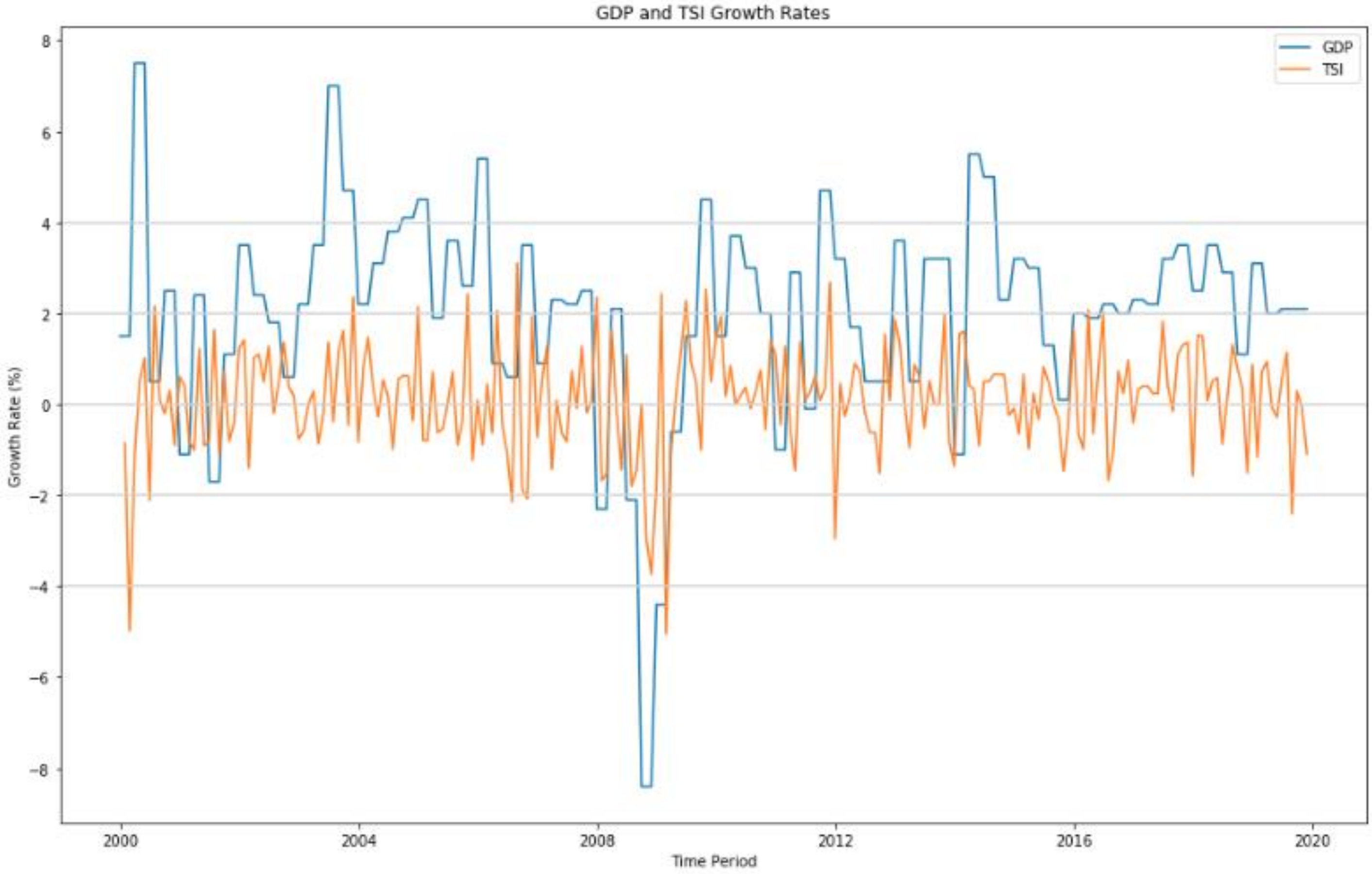
The GDP growth rates were converted from quarterly to monthly by treating the quarterly values as constant values for each month in each quarter.

The percent change for the freight data was calculated, as compared to the previous month for each column. This made the freight data consistent with how the GDP data was structured. All percentage data elements were in the format where for example a value of 10% was stored as 0.10.

The freight and GDP datasets were then merged together for some exploratory data analysis as described below in the Data Characteristics section.

Data Characteristics

The dataset had 240 rows consisting of monthly data from 1/1/2000 through 12/31/2019. The below chart shows the GDP and TSI growth rates since the year 2000. GDP is less noisy since it is only updated quarterly but the TSI is released monthly.



Split into Training, Validation, and Test Data Sets

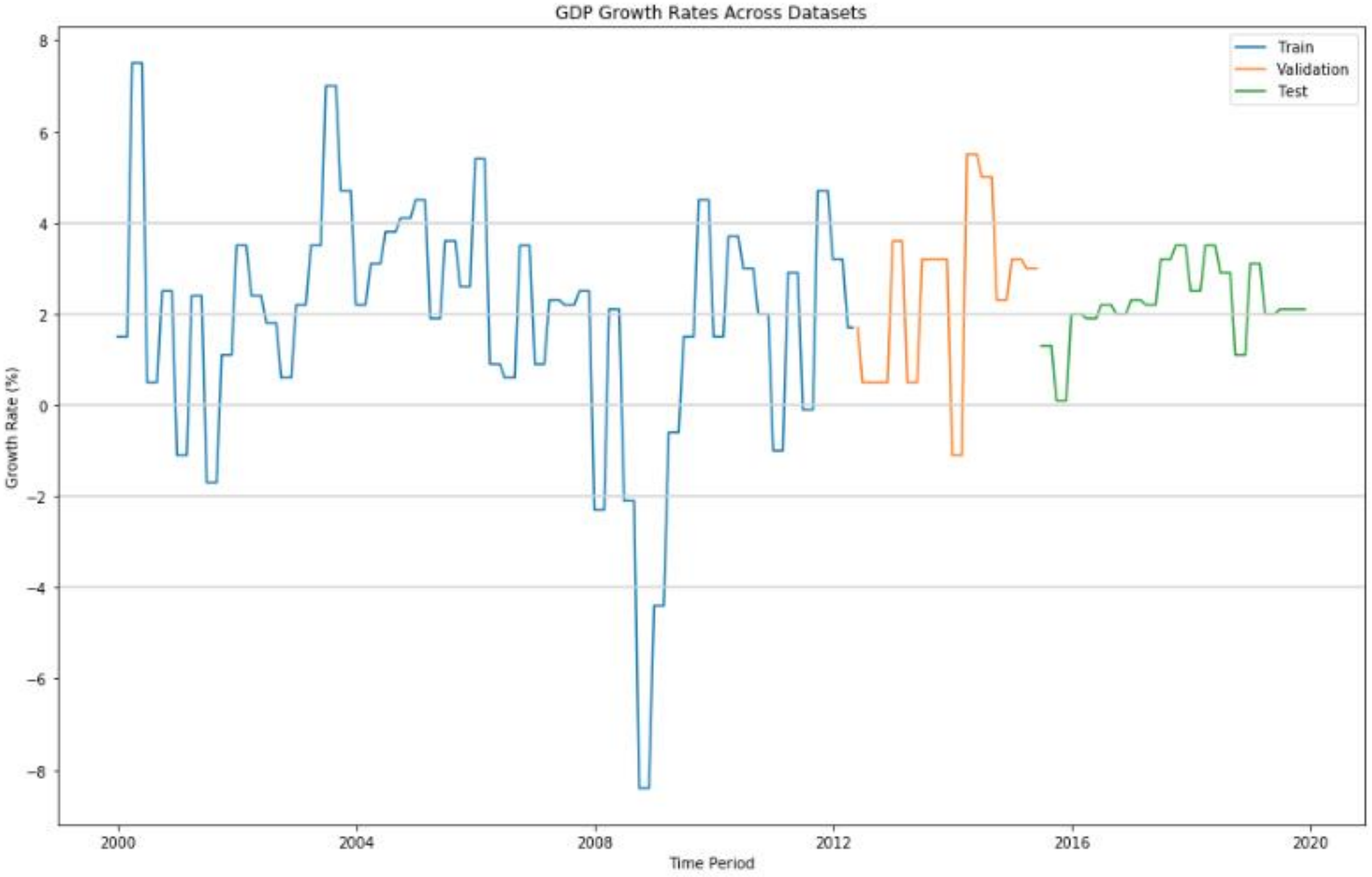
The data was split into training, validation, and test datasets. As stated above, the TSI index only has data back to January of the year 2000. The transportation data is published monthly. There were 240 rows of data available. The calculation of percent changes for the transportation data left the first row of data with no percent change values, so it was dropped. The distribution of the data used is shown below:

- Training: 149 rows
- Validation: 37 rows (20% of the training + validation set)
- Testing: 53 rows (22% of the overall data set)

Since this model was based on events happening over time, these three datasets were extracted without shuffling them first. A plot of the GDP growth rate with each data set identified is shown in the figure below. This shows that the training data has more variance than the validation and test sets.

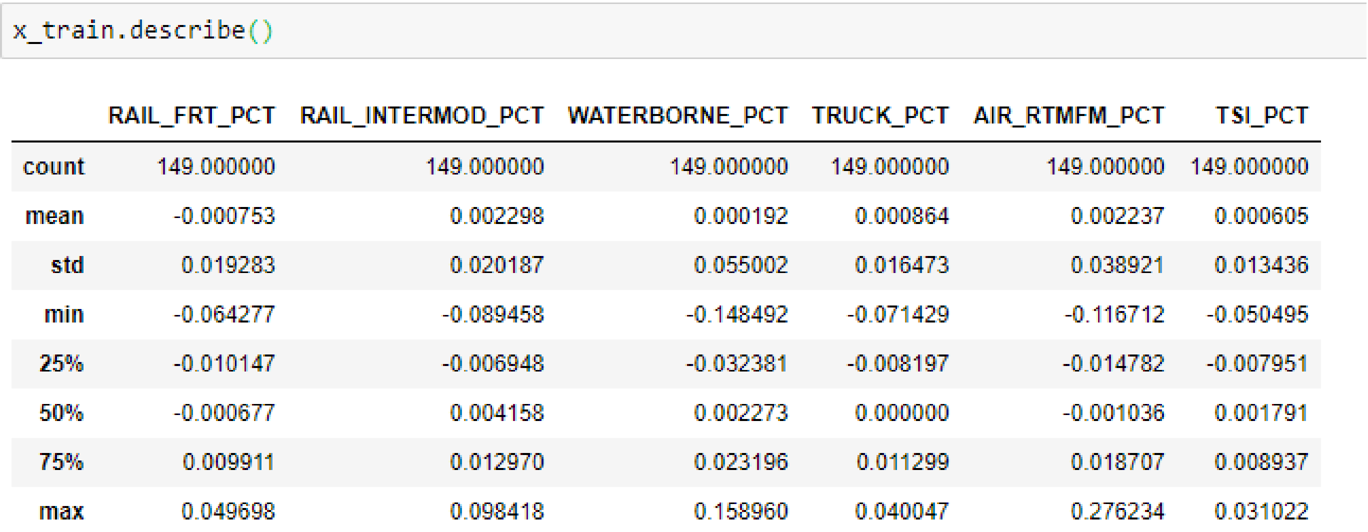
The data characteristics for the GDP quarterly growth rate data for the training set only is shown in the table below.

GDP Data Characteristic	Value
Mean	1.9%
Standard Deviation	2.7%
Min Observed Value	-8.4%
Max Observed Value	7.5%



Data Normalization

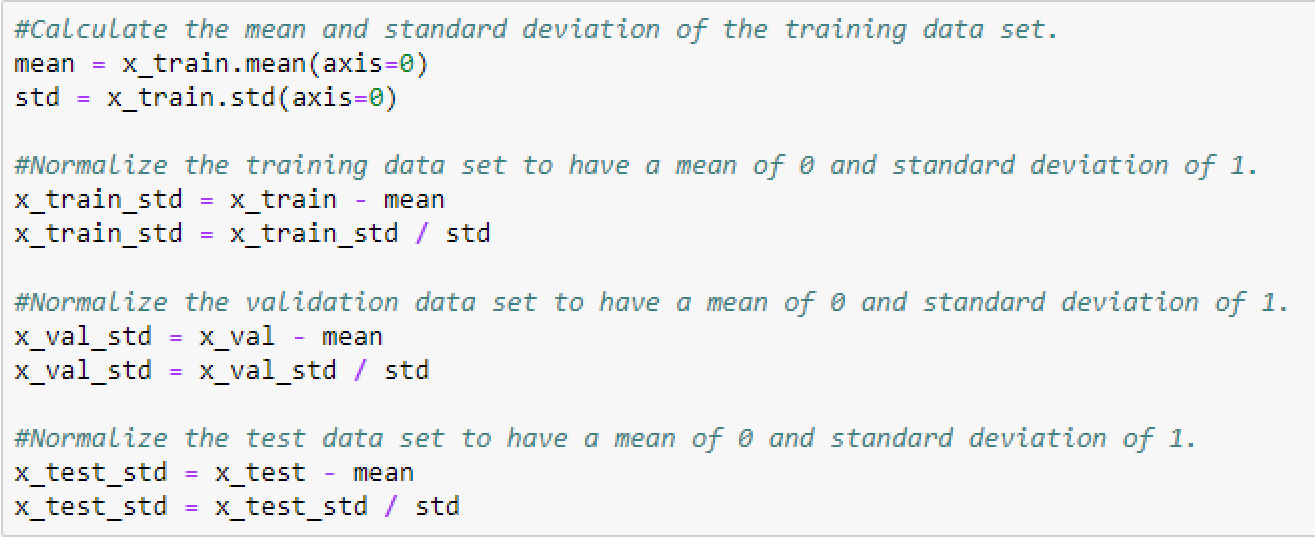
Given that the input data are all percent change values that are formatted such as 20% = 0.20, this means that all input data have small values. The training data had the below characteristics with all values within the range of -0.148 to 0.276.



```
x_train.describe()
```

	RAIL_FRT_PCT	RAIL_INTERMOD_PCT	WATERBORNE_PCT	TRUCK_PCT	AIR_RTMFM_PCT	TSI_PCT
count	149.000000	149.000000	149.000000	149.000000	149.000000	149.000000
mean	-0.000753	0.002298	0.000192	0.000864	0.002237	0.000605
std	0.019283	0.020187	0.055002	0.016473	0.038921	0.013436
min	-0.064277	-0.089458	-0.148492	-0.071429	-0.116712	-0.050495
25%	-0.010147	-0.006948	-0.032381	-0.008197	-0.014782	-0.007951
50%	-0.000677	0.004158	0.002273	0.000000	-0.001036	0.001791
75%	0.009911	0.012970	0.023196	0.011299	0.018707	0.008937
max	0.049698	0.098418	0.158960	0.040047	0.276234	0.031022

While the data already has small values with a mean close to zero, the data was normalized to have a mean of zero and standard deviation of one, based on the mean and standard deviation of the training data, as shown below.



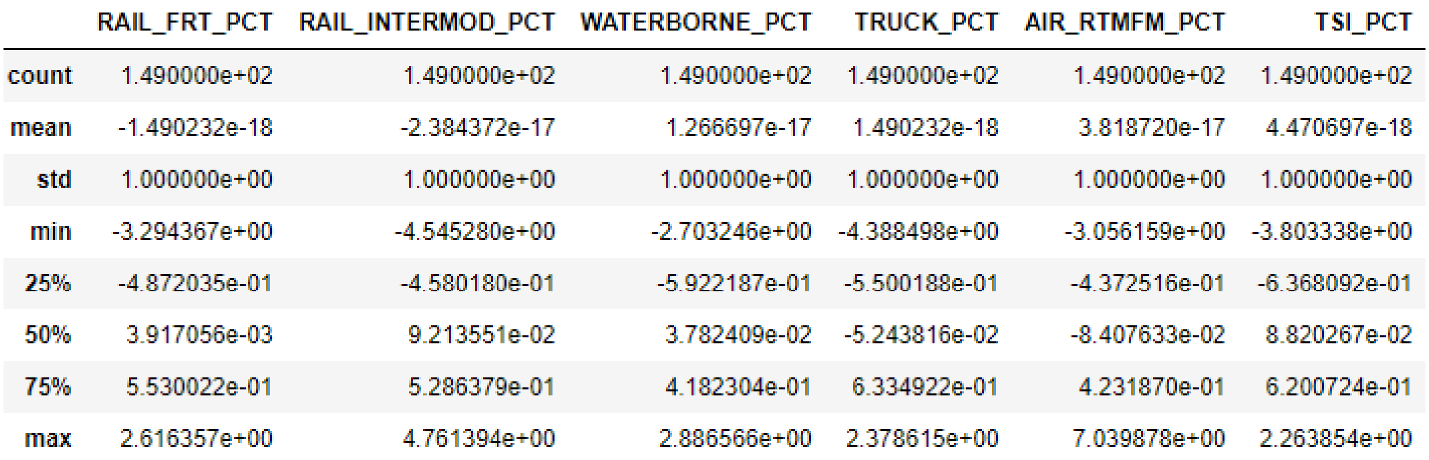
```
#Calculate the mean and standard deviation of the training data set.
mean = x_train.mean(axis=0)
std = x_train.std(axis=0)

#Normalize the training data set to have a mean of 0 and standard deviation of 1.
x_train_std = x_train - mean
x_train_std = x_train_std / std

#Normalize the validation data set to have a mean of 0 and standard deviation of 1.
x_val_std = x_val - mean
x_val_std = x_val_std / std

#Normalize the test data set to have a mean of 0 and standard deviation of 1.
x_test_std = x_test - mean
x_test_std = x_test_std / std
```

The resulting normalized input data is shown below. To achieve the standard deviation of 1.0, we now see values that range from -4.5 to 7.0. This normalization technique resulted in increasing the range of the values and the standard deviation.



	RAIL_FRT_PCT	RAIL_INTERMOD_PCT	WATERBORNE_PCT	TRUCK_PCT	AIR_RTMFM_PCT	TSI_PCT
count	1.490000e+02	1.490000e+02	1.490000e+02	1.490000e+02	1.490000e+02	1.490000e+02
mean	-1.490232e-18	-2.384372e-17	1.266697e-17	1.490232e-18	3.818720e-17	4.470697e-18
std	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
min	-3.294367e+00	-4.545280e+00	-2.703246e+00	-4.388498e+00	-3.056159e+00	-3.803338e+00
25%	-4.872035e-01	-4.580180e-01	-5.922187e-01	-5.500188e-01	-4.372516e-01	-6.368092e-01
50%	3.917056e-03	9.213551e-02	3.782409e-02	-5.243816e-02	-8.407633e-02	8.820267e-02
75%	5.530022e-01	5.286379e-01	4.182304e-01	6.334922e-01	4.231870e-01	6.200724e-01
max	2.616357e+00	4.761394e+00	2.886566e+00	2.378615e+00	7.039878e+00	2.263854e+00

Normalizing may not be of benefit in this situation based on the characteristics of the input features as described above. Further, the non-uniformity of the data between the training, validation, and test sets may lead to issues as a result of normalizing the validation and test sets based on the mean and standard deviation of the training set.

As a result of these concerns, both the raw input data and normalized input data were used to train models and then compare the results.

Machine Learning Methodology

A neural network was built using Keras to predict the GDP growth rate. Since there are only 6 features and a small amount of data, simple neural network structures were evaluated to reduce over fitting. The structure of the neural network was:

- Dense Layer
- Batch Normalization Layer
- Dense Layer
- Batch Normalization Layer
- Dense Layer with 1 unit and linear activation

A grid search was setup to evaluate different combinations of the following:

- Number of units in the dense layers other than the last dense layer
- Type of regularization (L1 and L2)
- Regularization penalty
- Use of normalized input features vs raw input features
- Different combinations of the 6 available input features

For each test case in the grid search, the following steps were performed:

1. Setup each epoch to include all 149 training data points and each test case to run 300 epochs.
2. Train the model with mean absolute error being the performance metric.
3. Measure the mean absolute error on the validation set.
4. Extract the minimum mean absolute error observed for both the training set and validation set for each test case.
5. Store the parameters for the test case and the mean absolute error values in a pandas data frame.

The pandas data frame was then used to evaluate the performance of the test cases.

The findings of the grid search were as follows:

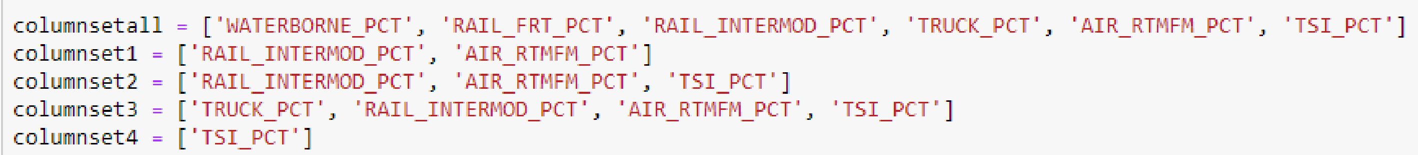
Normalized vs Raw Data

The test cases with the normalized data more commonly had larger mean absolute error values for both the training data and validation data. For example, in one grid search batch that covered 40 test cases, 9 of the 10 worst performing test cases had normalized input data and 7 of the 10 best performing test cases used raw input data. This was while the other parameters in the grid search were equally varied for both normalized and raw input data.

In this scenario with the input features already having small values it does not seem needed to normalize the input data. Two batch normalization layers were used in the neural network.

Combinations of Input Features

The combinations of input features that were evaluated are shown below.



```
columnsetall = ['WATERBORNE_PCT', 'RAIL_FRT_PCT', 'RAIL_INTERMOD_PCT', 'TRUCK_PCT', 'AIR_RTMFM_PCT', 'TSI_PCT']
columnset1 = ['RAIL_INTERMOD_PCT', 'AIR_RTMFM_PCT']
columnset2 = ['RAIL_INTERMOD_PCT', 'AIR_RTMFM_PCT', 'TSI_PCT']
columnset3 = ['TRUCK_PCT', 'RAIL_INTERMOD_PCT', 'AIR_RTMFM_PCT', 'TSI_PCT']
columnset4 = ['TSI_PCT']
```

While not drastic, there were some differences in performance observed in the grid search test cases. A ranking of occurrence of each combination in the top half of best performing test cases is shown below:

- 63% - “columnset2”, “columnset3”, and “columnset4”
- 38% - “columnsetall”
- 25% - “columnset1”

This shows that “columnset2”, “columnset3”, and “columnset4” had better performance than the other column sets. There were no clear factors in the results to separate those three options. Since the number of features is already so small, I decided to choose “columnset3” as the set to use since it has the most features.

Regularization Type and Penalty

The two regularization types that were evaluated were L1 and L2. L2 regularization was seen to need larger penalty values than L1 regularization. When plots of training error and validation error were plotted by epoch number during training, the plots for L2 regularization were seen to be noisier. There also was not as much consistency in the results with L2 regularization. As a result, L1 regularization was chosen.

A penalty value of 0.04 was observed to predominantly be associated with better performing test cases in the grid search test cases, so 0.04 was chosen.

Number of Units in Dense Layers

The grid search included values of 16, 32, and 64 for the number of units in the dense layers besides the last dense layer. Test cases with 64 units were seen to predominantly perform better than the other values, so 64 was chosen.

Results

The summary of the resulting neural network parameters per the discussion above is:

Item	Selected Value
Number of Units in Dense Layers	64
Regularization Type	L1
Regularization Penalty	0.04
Normalized vs Raw Input Data	Raw
Input Feature Set	“TRUCK_PCT”, “RAIL_INTERMOD_PCT”, “AIR_RTMFM_PCT”, “TSI_PCT”

The performance of this model against the data sets was approximately as shown below.

Data Set	Performance (Mean Absolute Error)
Training	1.9%
Validation	1.7%
Test	0.7%

I believe the improved performance in the test set to be a result of there being less variance in that dataset.

The characteristics of the Real GDP quarterly growth rate training data are revisited below for comparison purposes.

GDP Data Characteristic	Value
Mean	1.9%
Standard Deviation	2.7%
Min Observed Value	-8.4%
Max Observed Value	7.5%

As an example, if a value of 1.9% (which is the observed training set mean) is predicted by the model, then the actual GDP growth rate is likely between 0% and 3.8%. This is based on the mean absolute error value that was seen in the training data of 1.9%.

Next Steps

Performance could likely be improved by adding more input features for metrics related to the performance of the economy such as employment data, loan defaults, bankruptcies, etc.

While the year 2000 was the first year that the Transportation Services Index that was reported, the other data from this project may be available back farther in time. More data could improve performance.

For the year 2020 after the first quarter, if new data for the data elements used in this project get pulled in to re-train the model, it will likely have drastic impacts as the GDP is expected to have an extreme decrease.

A sequence model, such as a Long Short-Term Memory model, may also perform better since this is a sequential type of problem and there is likely a tendency for a quarter's GDP growth rate to be in a similar range as compared to the previous quarter.

I would also like to setup a website that can provide GDP growth estimates based on the latest data that has been released and / or input data that users can enter.