GDP Growth Classifier – Increasing/Decreasing

Introduction 1
Business Opportunity 2
Analytical Framework 2
Analytical Question 2
Data Acquisition 2
Freight Data 2
GDP Data 3
Data Cleansing / Manipulation 3
Data Characteristics 4
Split into Training, Validation, and Test Data Sets 5
Data Normalization 6
Evaluation of Feature Importance 7
Recursive Feature Elimination (RFE) 7
Marginal Effects 8
Comparison of the Feature Analysis Results 10
Machine Learning Methodology 10
Logistic Regression 10
Logistic Regression Results 10
Neural Network 11
Neural Network Results 12
Overall Results 13
Next Steps 13

Introduction

The goal of this project was to evaluate the ability to predict if GDP is increasing or decreasing based on transportation metrics from the U.S. Department of Transportation (DOT) Bureau of Transportation Statistics (BTS). Freight transportation is impacted by many factors that impact GDP such as:

- Consumption of durable and nondurable goods
- Nonresidential 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 cost/expenses sooner to reduce losses.

Analytical Framework

Analytical Question

The goal of this project is to use supervised machine learning to build a classification model that predicts if GDP is increasing or decreasing based on transportation freight metrics. Here, increasing is defined as a positive GDP growth rate as calculated from the previous quarter and decreasing is defined as a negative growth rate. 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 carried out this project to see what level of performance is possible using this simple approach.

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 of 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.

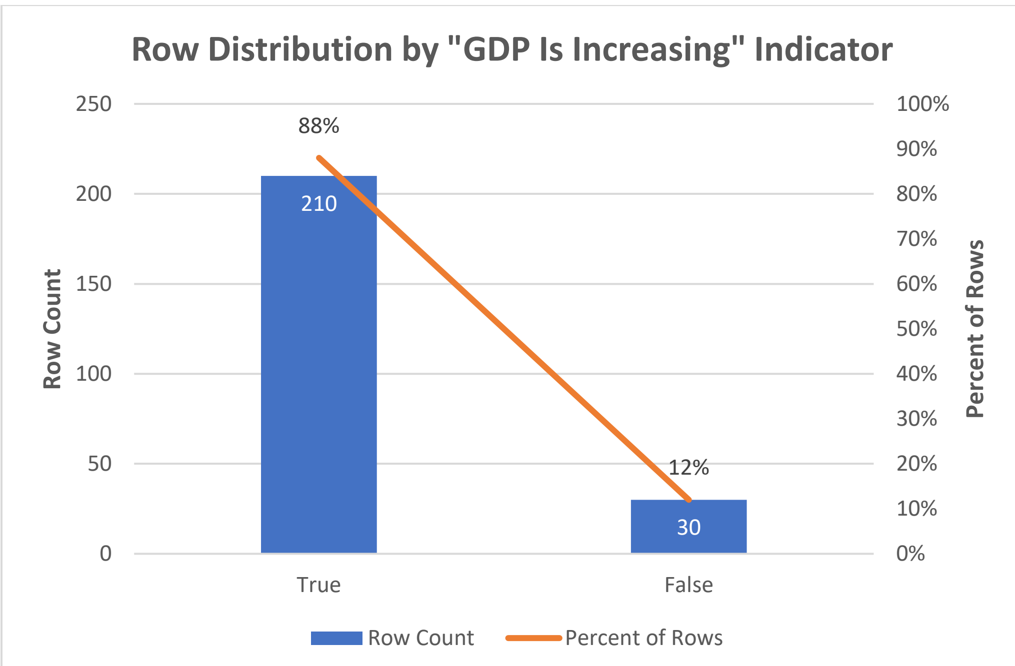
A Boolean indicator was added to the dataset to represent if the GDP was increasing. So, True meaning that GDP growth was positive while False means that GDP growth was negative.

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.

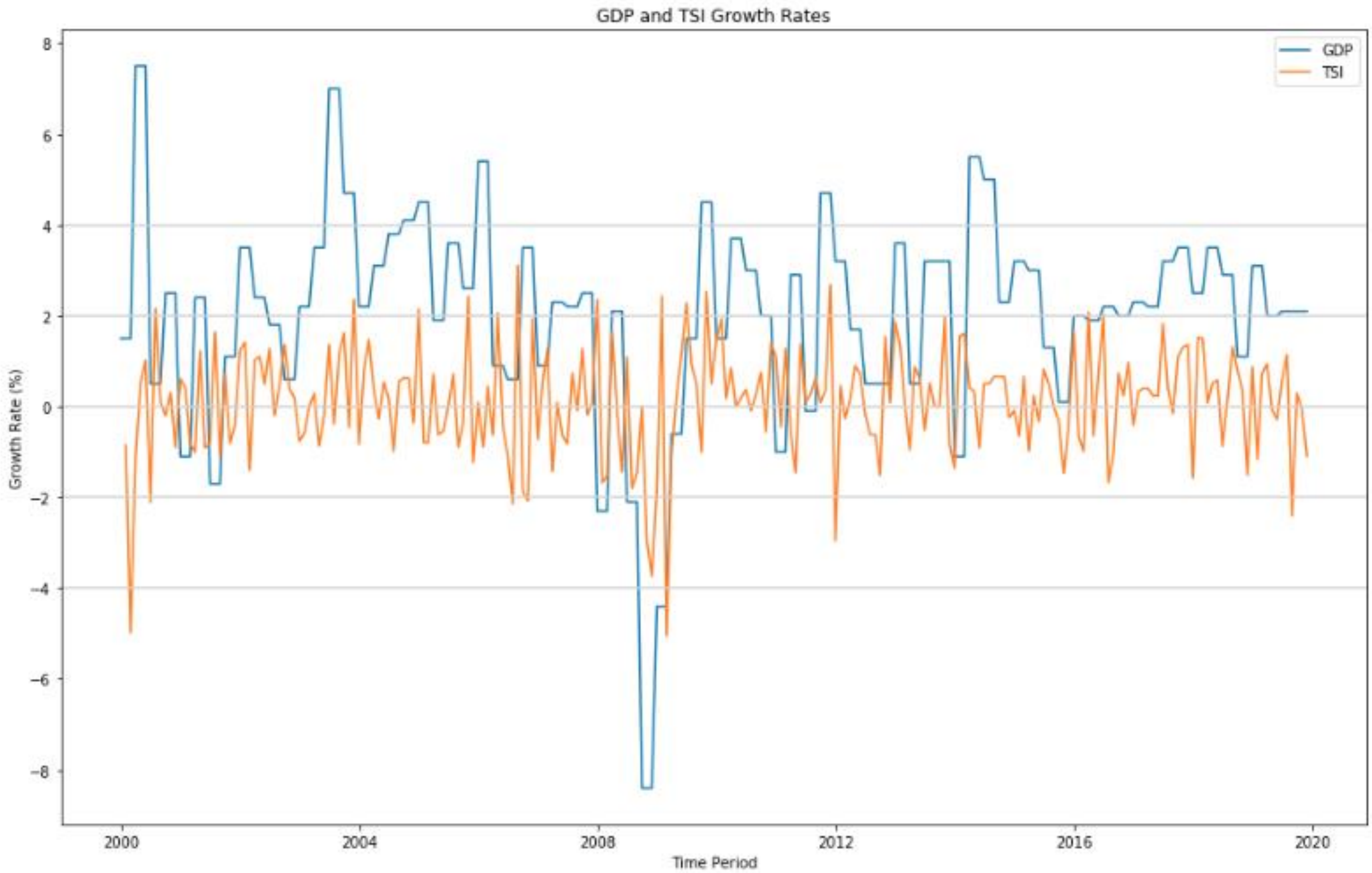
The freight and GDP datasets were then merged together for some exploratory data analysis as described below in the Data Characteristics section.

Data Characteristics

The dataset had 240 rows consisting of monthly data from 1/1/2000 through 12/31/2019. The distribution of the GDP increasing indicator was as shown below. This shows that the data is imbalanced as the GDP growth rate is positive 88% of the time. This imbalance is good for the US economy but may pose some challenges with model training. Class "False" recall will need to be evaluated to determine if the classification prediction results are satisfactory. For starters, the goal will be to beat the 88% accuracy that could be achieved by predicting True for every input sample, while also having the ability to predict "False" cases.



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.

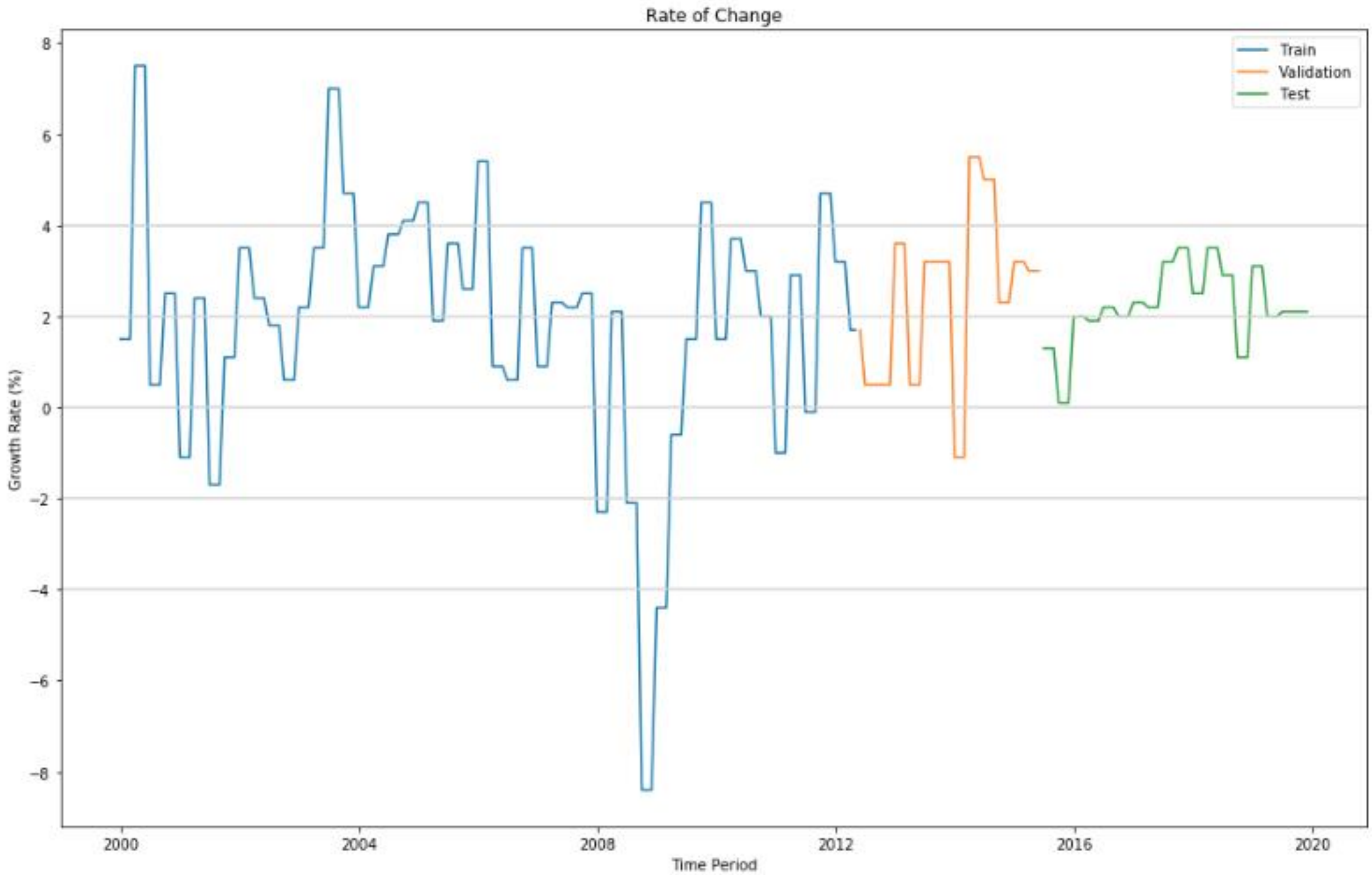


Split into Training, Validation, and Test Data Sets

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 below. This shows that the training data has more variance than the validation and test sets.



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.

Evaluation of Feature Importance

The below methods were used to evaluate the features in this exercise.

Recursive Feature Elimination (RFE)

Recursive feature elimination is a procedure that recursively trains a model, evaluates the importance of each feature, and then repeats after removing additional features. I applied this approach to both the raw input data and after the data had been normalized as I wanted to see if the results would be any different.

For the raw training data, the features in order of importance were:

AIR_RTMFM_PCT
RAIL_INTERMOD_PCT
TSI_PCT
TRUCK_PCT
WATERBORNE_PCT
RAIL_FRT_PCT

Resulting from:

```
# Recursive Feature Elimination (RFE) for training set
from sklearn.feature_selection import RFE
rfelogreg = LogisticRegression()
selector = RFE(rfelogreg, 1)
selector = selector.fit(x_train, y_train_class)
print(selector.support_)
print(selector.ranking_)
```

```
[False False False False  True False]
[4 2 6 5 1 3]
```

On the standardized input data, the order was slightly different, with the items in red being the ones that were in different order than above:

AIR_RTMFM_PCT
RAIL_INTERMOD_PCT
RAIL_FRT_PCT
TSI_PCT
WATERBORNE_PCT
TRUCK_PCT

Marginal Effects

Marginal effects are defined as:

> Marginal effects tell us how a dependent variable (outcome) changes when a specific independent variable (explanatory variable) changes. Other covariates are assumed to be held constant. (https://www.statisticshowto.com/marginal-effects/)

The below operation was performed on the raw training set, which shows that the highlighted input variables have p-values of less than 0.05 (5%). This indicates that these variables are statistically significant in the logistic regression model that was created. For example, a p-value of 0.0020 for RAIL_INTERMOD_PCT means that this variable has a 0.2% probability of having impacted the model that was analyzed based on chance alone.

```
import statsmodels.api as sm
logit_model=sm.Logit(y_train_class, x_train)
resultlogit=logit_model.fit()
print(resultlogit.summary2())
```

```
Optimization terminated successfully.
         Current function value: 0.627405
         Iterations 6
                          Results: Logit
=====================================================================
Model:                Logit             Pseudo R-squared:  -0.326
Dependent Variable:   gdp_is_increasing AIC:               198.9668
Date:                 2020-06-18 21:52  BIC:               216.9904
No. Observations:     149               Log-Likelihood:    -93.483
Df Model:             5                 LL-Null:           -70.510
Df Residuals:         143               LLR p-value:       1.0000
Converged:            1.0000            Scale:             1.0000
No. Iterations:       6.0000
---------------------------------------------------------------------
                   Coef.   Std.Err.    z    P>|z|    [0.025   0.975]
---------------------------------------------------------------------
RAIL_FRT_PCT      -11.4630  13.6142 -0.8420 0.3998  -38.1464  15.2203
RAIL_INTERMOD_PCT  36.3123  11.7634  3.0869 0.0020   13.2565  59.3682
WATERBORNE_PCT     -1.0022   5.2069 -0.1925 0.8474  -11.2075   9.2031
TRUCK_PCT          10.0182  43.2962  0.2314 0.8170  -74.8407  94.8771
AIR_RTMFM_PCT      14.7219   7.1881  2.0481 0.0406    0.6335  28.8103
TSI_PCT             2.4778  65.9910  0.0375 0.9700 -126.8623 131.8178
=====================================================================
```

For the standardized training set, the below marginal effects results were obtained. These results show that only RAIl_INTERMOD_PCT was statistically significant.

```
# Marginal effects for normalized training set
import statsmodels.api as sm
logit_model=sm.Logit(y_train_class, x_train_std)
resultlogit=logit_model.fit()
print(resultlogit.summary2())
```

```
Optimization terminated successfully.
         Current function value: 0.659362
         Iterations 5
                         Results: Logit
=================================================================
Model:              Logit             Pseudo R-squared: -0.393
Dependent Variable: gdp_is_increasing AIC:              208.4899
Date:               2020-06-22 20:59  BIC:              226.5135
No. Observations:   149               Log-Likelihood:   -98.245
Df Model:           5                 LL-Null:          -70.510
Df Residuals:       143               LLR p-value:      1.0000
Converged:          1.0000            Scale:            1.0000
No. Iterations:     5.0000
-----------------------------------------------------------------
                   Coef.  Std.Err.    z    P>|z|   [0.025 0.975]
-----------------------------------------------------------------
RAIL_FRT_PCT      -0.0826   0.2536 -0.3257 0.7446 -0.5796 0.4144
RAIL_INTERMOD_PCT  0.4238   0.2142  1.9788 0.0478  0.0040 0.8435
WATERBORNE_PCT    -0.1690   0.2797 -0.6043 0.5456 -0.7172 0.3792
TRUCK_PCT         -0.1879   0.6951 -0.2703 0.7869 -1.5502 1.1744
AIR_RTMFM_PCT      0.3419   0.2661  1.2844 0.1990 -0.1798 0.8635
TSI_PCT            0.3646   0.8642  0.4219 0.6731 -1.3292 2.0584
=================================================================
```

Comparison of the Feature Analysis Results

The marginal effects analysis indicated that only the RAIL_INTERMOD_PCT and AIR_RTMFM_PCT inputs are statistically significant.

The ranking order by the Recursive Feature Elimination algorithm is shown below with the differences highlighted in red. All three approaches agree that RAIL_INTERMOD_PCT and AIR_RTMFM_PCT are 2 most impactful features in the input data.

Raw Training Set	**Normalized Training Set**
AIR_RTMFM_PCT	AIR_RTMFM_PCT
RAIL_INTERMOD_PCT	RAIL_INTERMOD_PCT
TSI_PCT	WATERBORNE_PCT
RAIL_FRT_PCT	TSI_PCT
TRUCK_PCT	RAIL_FRT_PCT
WATERBORNE_PCT	TRUCK_PCT

Since this exercise involved a small number of features and since there was a small amount of data, multiple combinations of different features were evaluated using both raw and normalized data.

Machine Learning Methodology

Logistic Regression

I implemented a logistic regression model using sklearn. I performed a grid search over different combinations of input features, normalized/non-normalized inputs, class weights, and regularization factors.

Each model iteration was trained and then evaluated against a validation data set. Each set of results were then written as a new row to a pandas dataframe.

For each model, the below metrics were recorded. The output of class 0, which means that "GDP is increasing" is false, is a less frequently occurring result in the historical data. This was the motivation to include the Class 0 Recall in the metrics that were recorded.

- Training Accuracy
- Training Class 0 Recall
- Training Combined Score = Training Accuracy + Training Class 0 Recall
- Validation Accuracy
- Validation Class 0 Recall
- Validation Combined Score = Validation Accuracy + Validation Class 0 Recall

Results were evaluated for both the training and validation data.

The validation set only had 3 values of gdp_is_increasing = 0, so this did not provide much opportunity to test the class 0 recall. The training set had 27 values of gdp_is_increasing = 0, so the training set was used more to evaluate the class 0 recall.

Logistic Regression Results

The results of the grid search operation were examined based on the highest class 0 recall training score, training accuracy, and based on the training accuracy and class 0 recall added together. Observations are summarized in the table below.

Type of Results Analysis	Training Accuracy	Training Class 0 Recall
Maximum Class 0 Recall	60%	78%
Maximum Accuracy	82%	0% - 4%
Maximum Combo of Class 0 Recall + Accuracy	56% - 67%	67% - 78%

These results show that there were not any models in the grid search that met the performance requirements.

As mentioned, the class value of 1 which means that the GDP is increasing, occurred 88% of the time. So, 88% accuracy could be obtained by predicting 1 for every input sample. The goal is to at least beat 88% accuracy while also having the ability to predict class 0 occurrences.

These results show that a logistic regression model is not able to achieve this performance level with the input features that were selected.

Neural Network

I built a neural network using Keras to see what performance it could provide compared to logistic regression. 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 sigmoid activation to produce a binary output

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 or L2) to apply to the dense layers other than the last dense layer
- Regularization penalty
- Use of normalized input data vs raw input data
- Different combinations of the 6 available input features
- Class weight

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

1. Train the model with mean absolute error being the performance metric.
2. Measure the mean absolute error on the validation set.
3. Extract the minimum mean absolute error observed for both the training set and validation set for each test case.
4. 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 parameters/configurations that were selected based on the grid search were:

Item	Selected Value
Number of Units in Dense Layers	16
Regularization Type	L2
Regularization Penalty	0.001
Normalized vs Raw Input Data	Raw

Input Feature Set	'RAIL_FRT_PCT', 'RAIL_INTERMOD_PCT', 'WATERBORNE_PCT', 'TRUCK_PCT', 'AIR_RTMFM_PCT', 'TSI_PCT'
Class Weight	{1: 1, 0: 5}

Neural Network Results

The overall accuracy for the 3 test sets were Training = 97%, Validation = 89%, and Test = 89%. The below outputs also show that the Test set unfortunately had 0 support for class 0, meaning that there were no instances of class 0 in the Test set data. There were 27 instances of class 0 in the Training set and 3 in the Validation set. These reports show that 29 out of the 30 class 0 instances were correctly identified by the model.

The biggest challenge with this project is the limited amount of data. As a result, the Training data had to be relied on to evaluate the ability to identify class 0 scenarios.

Additional metrics showing the performance of this model against the data sets is below, from the sklearn classification report. The results for the training set were:

```
Train              precision    recall  f1-score   support

           0       0.84      1.00      0.92        27
           1       1.00      0.96      0.98       122

    accuracy                           0.97       149
   macro avg       0.92      0.98      0.95       149
weighted avg       0.97      0.97      0.97       149
```

The results for the validation set were:

```
Val                precision    recall  f1-score   support

           0       0.40      0.67      0.50         3
           1       0.97      0.91      0.94        34

    accuracy                           0.89        37
   macro avg       0.68      0.79      0.72        37
weighted avg       0.92      0.89      0.90        37
```

And the results for test set were:

```
Test               precision    recall  f1-score   support

           0       0.00      0.00      0.00         0
           1       1.00      0.89      0.94        53

    accuracy                           0.89        53
   macro avg       0.50      0.44      0.47        53
weighted avg       1.00      0.89      0.94        53
```

Overall Results

As mentioned, the class value of 1 which means that the GDP is increasing, occurred 88% of the time. So, 88% accuracy could be obtained by predicting 1 for every input sample. The goal for this project is to at least beat 88% accuracy while also having the ability to predict class 0 occurrences.

The logistic regression model was not able to achieve an acceptable level of both accuracy and class 0 recall, with accuracy levels no higher than 82%. These accuracy levels came with low class 0 recall scores.

The neural network model was able to achieve satisfactory performance, as summarized in this table:

Data Set	Accuracy	Class 0 Recall
Training	97%	100%
Validation	89%	67%*
Test	89%	0%**

* The validation set only had 3 class 0 instances, with 2 of the 3 correctly identified.

** The test set did not have any class 0 instances.

This classifier project was an experiment to see if transportation freight data has significance as a leading indicator for the GDP growth rate. I believe that the neural network model results show that this is the case.

Next Steps

Since this project showed that transportation freight data has significance related to predicting at least some level of GDP growth, an interesting next step is to build a regression model to predict the GDP growth rate.

This would be interesting to do first just based on the same data that was used in this project, but then to see how much more accuracy can be obtained by bringing in other types of data.