Economic Growth or Contraction

John Balzani

1/3/2020

Executive Summary:

The Composite Leading Index (CLI) is an index composed of various indicators that are thought to have predictive value for the US economy over the following 6-9 months, with higher values of the index corresponding to better economic outcomes (OECD, 2019). The purpose of this report is to see if the CLI can predict the direction of economic growth in a given quarter for the upcoming 3 quarters, with predictive value above the values of real GDP growth itself. In particular, one of the purposes of this research is to explore if these predictions can be made accurately for the 2008 recession beginning in July of 2008 and for the and current post-2008 recession business cycle. The models employed are a logistic regression model, a k-nearest neighbors model, and a random forest model. An ensemble prediction is then made for whether or not there will be economic growth, with a zero representing negative or no economic growth, and a 1 representing economic growth.

Data from 1982 are used to make lags of variables and the first differences of the variables, data from 1983 to April 2008 are used to estimate the models used, and data from July 2008 to July 2019 are used to evaluate the models. The economic growth data consists of quarterly observations of real US GDP (economic) growth from January 1982 to July 2019. The CLI data consists of monthly observations of the Composite Leading Index from June of 1961 to July of 2019. These are time series data.

This project undertook the following key steps. First, the data were cleaned and explored. Second, the data are tested for stationarity, since time series data should be tested for stationarity. Stationarity means that the "statistical properties are constant over time", and this is required to do much statistical analysis (Nau, 2019). Third, the models were created. Fourth, the models were evaluated with accuracy and F1 score used as the metrics to judge model effectiveness. It is found that the logistic regression is the best out of these models for predicting economic growth or contraction. This model has an accuracy of 93.3% during the test period, which is quite good, and a F1 score of .727. This model outperforms a simple strategy of predicting economic growth for every period, which would give an accuracy of 84.4% for the test period. The other model results were, in order of effectiveness, the random forest model (accuracy 84.4%, F1 score 0), the ensemble model (accuracy 84.4%, F1 score 0), and the knn model (accuracy 82.2%, F1 score 0).

Methods and Analysis:

After importing the data, the data was separated into training and test sets. Since this is time series data, the training and test sets cannot be determined randomly. In order to see whether or not economic growth or contraction can be predicted for the current business cycle which began with the 2008 recession, the data for the 2008 recession starting July 2008 and after is used as the test set.

The GDP growth dataset was briefly reviewed for NA values, outliers, and duplicates. It was found that there are no NA data points, as can be seen below.

```
#check for NAs in data
A191RL1Q225SBEA %>% filter(is.na(DATE))
```

```
## # A tibble: 0 x 2
## # ... with 2 variables: DATE <date>, A191RL1Q225SBEA <dbl>
```

```
A191RL1Q225SBEA %>% filter(is.na(A191RL1Q225SBEA))
```

```
## # A tibble: 0 x 2
## # ... with 2 variables: DATE <date>, A191RL1Q225SBEA <dbl>
```

After checking for duplicates, it was found that there are no duplicate values, as can be seen below.

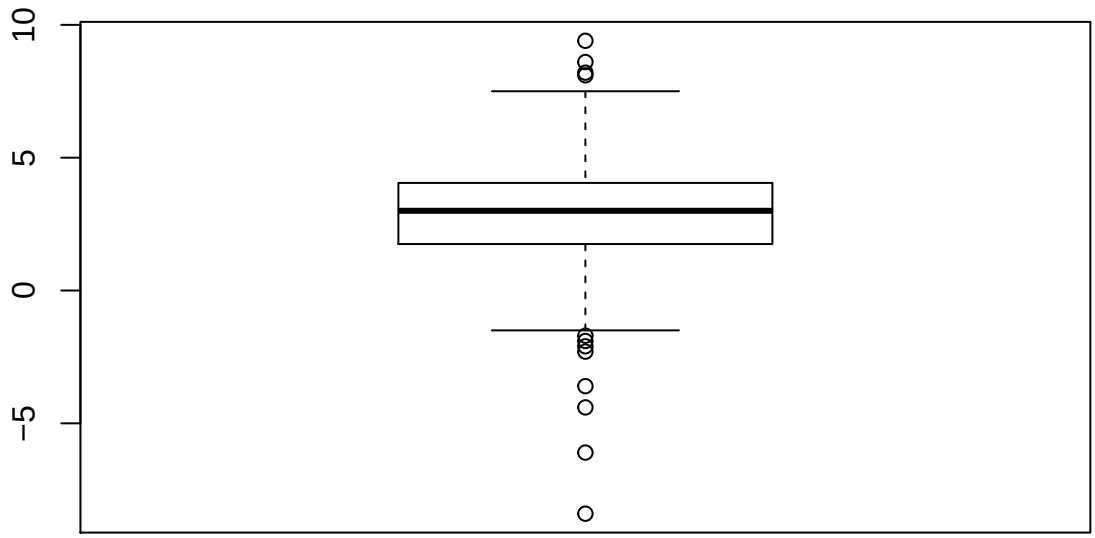
```
#check for duplicates
gdp_duplicates <- A191RL1Q225SBEA[duplicated(A191RL1Q225SBEA), ]
gdp_duplicates
```

```
## # A tibble: 0 x 2
## # ... with 2 variables: DATE <date>, A191RL1Q225SBEA <dbl>
```

The minimum and maximum values of GDP growth were checked in order to see if there are any problematic outliers. The minimum value was -8.4 and the maximum value was 9.4. These are reasonable values, so I next generate a boxplot to visualize the data.

A boxplot of the GDP growth rates was also generated, in order to better visualize the distribution (Figure 1). It can be seen that GDP growth is usually between 0 and 5 percent, and occasionally higher or lower.

Figure 1: Boxplot of GDP Growth



The CLI data was then examined for NA values, duplicates, and outliers.

```
#check for NAs in data
OECDLOLITOAASTSAM %>% filter(is.na(DATE))
```

```
## # A tibble: 0 x 2
## # ... with 2 variables: DATE <date>, OECDLOLITOAASTSAM <dbl>
```

```
OECDLOLITOAASTSAM %>% filter(is.na(OECDLOLITOAASTSAM))
```

```
## # A tibble: 0 x 2
## # ... with 2 variables: DATE <date>, OECDLOLITOAASTSAM <dbl>
```

After checking for duplicates, it was found that there are no duplicate values, as can be seen below.

```
#check for duplicates
cli_duplicates <- OECDLOLITOAASTSAM[duplicated(OECDLOLITOAASTSAM), ]
cli_duplicates
```

```
## # A tibble: 0 x 2
## # ... with 2 variables: DATE <date>, OECDLOLITOAASTSAM <dbl>
```

The minimum and maximum values of the Composite Leading Index were checked in order to see if there are any problematic outliers. The minimum value was 95.569 and the maximum value was 103.348. These are reasonable values, so I next generate a boxplot to visualize the data.

A boxplot of the CLI was also generated, in order to better visualize the distribution (Figure 2). It can be seen that the CLI is usually around 100, and that there are more very low values (below minimum bar) than very high values (above maximum bar). The fatter tail to the downside can also be seen in a histogram (Figure 3).

Figure 2: Boxplot of CLI Data

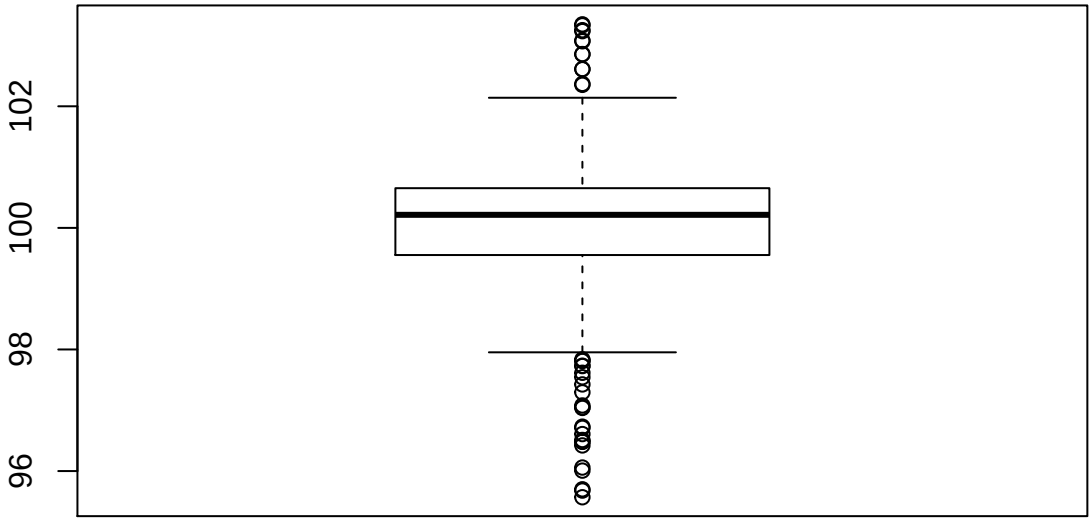
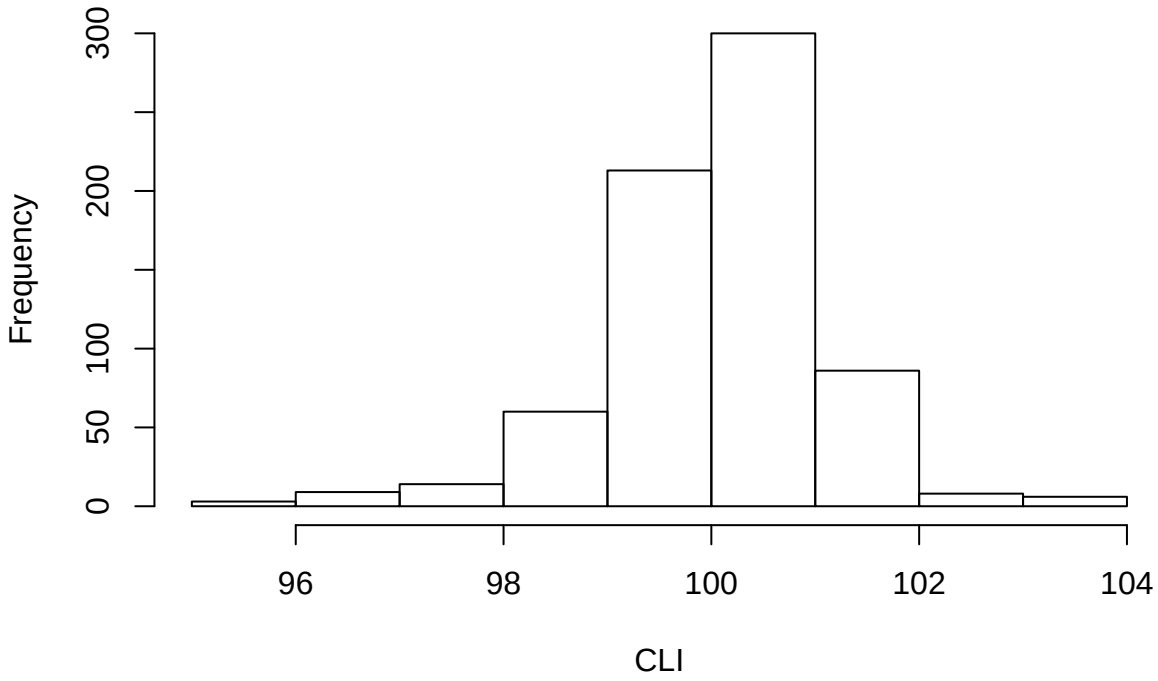


Figure 3: Histogram of CLI Data



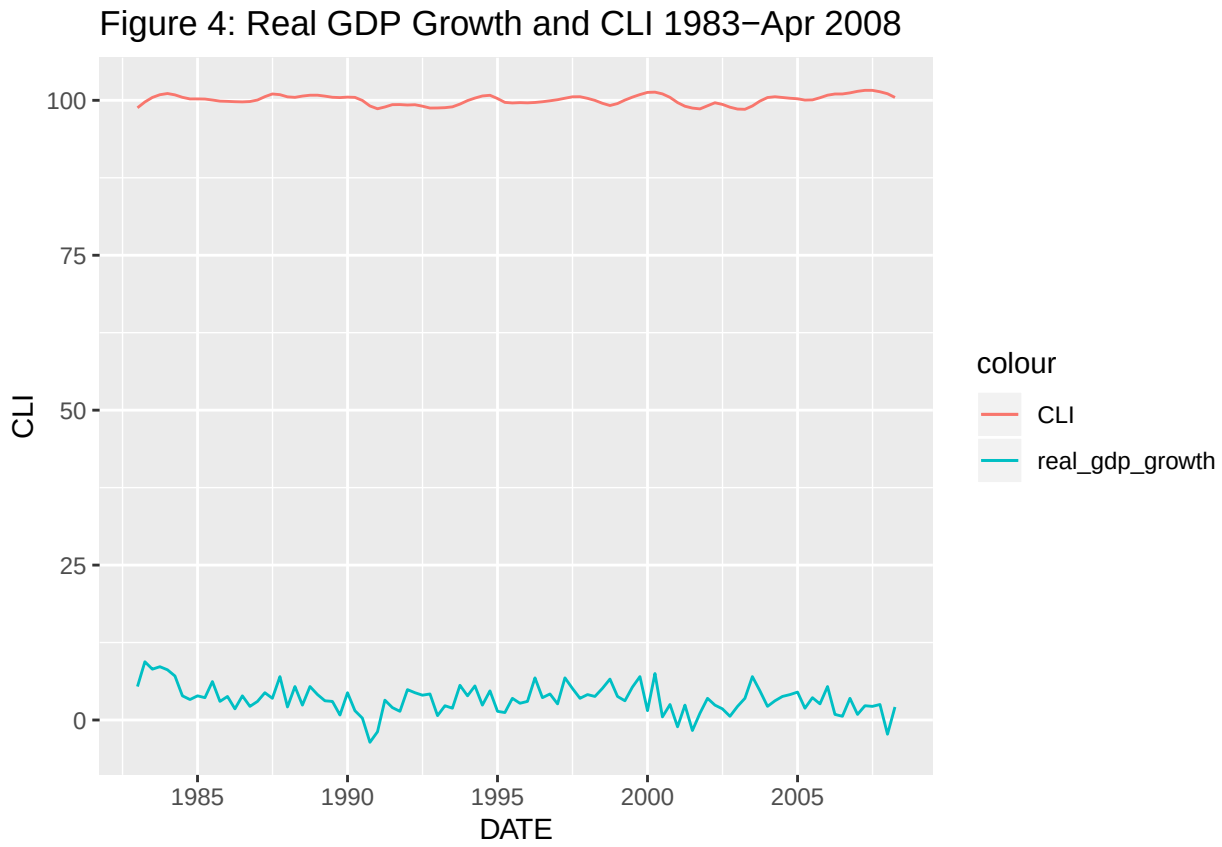
After checking for NAs, duplicates, and outliers, the CLI data is filtered to only include the relevant time period, and only those data points that match the data in the real GDP growth dataset are selected. This is done by creating a variable called MONTH that extracts the month of the observation, then filtering for those months for which quarterly GDP data is available, which are months 1, 4, 7, and 10.

```
#get 1st date of each month
CLI_filtered <- OECDLOLITOAASTSAM %>%
  filter(DATE >= '1982-01-01') %>%
  mutate(MONTH = month(DATE)) %>% #extract month
  filter(MONTH %in% c(1, 4, 7, 10))#filter for months with qtly gdp data
```

Finally, the datasets are combined into one dataset. A variable called gdp_impr is created to represent the status of GDP improvement. This is a binary variable equal to 1 if GDP improves in a given period and is equal to 0 if it does not. At this point, nothing I have done in my exploratory data analysis has given me information about what the relationship in the test set is between economic growth and the CLI.

After creating the combined dataset, the train and test sets are created. Again, 1983 up to July 2008 makes up the training set, and July 2008 and after is the test set.

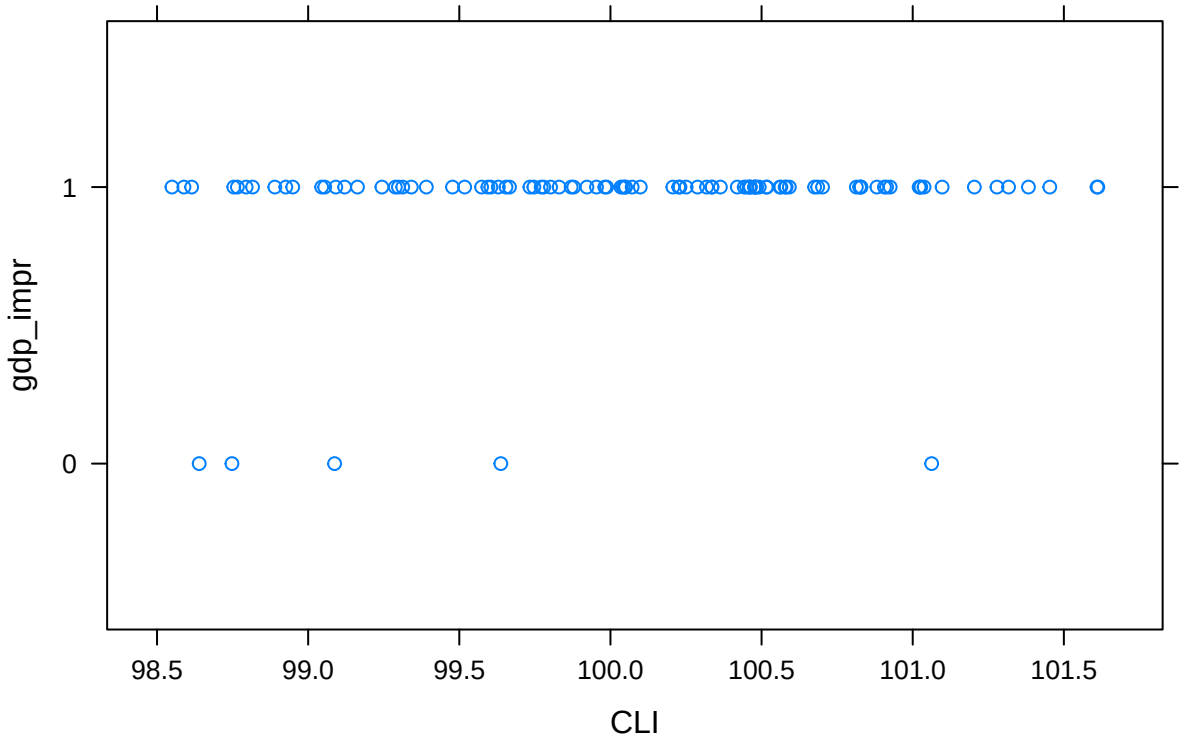
Next, the real GDP growth and CLI data (training set only) are plotted to see how the variables change over time and the relationship between the two (Figure 4).



It can be seen that the time series appear to be stationary over time, and that there may be a relationship between the two, but the relationship is not completely clear.

Next, the gdp_impr variable is plotted against the CLI to see if the relationship can be better visualized (Figure 5). It can be seen that most of the values of 0 for gdp_impr occur at lower values of the CLI, when the CLI is below 100.

Figure 5: gdp_impr vs. CLI



After doing the exploratory data analysis, I next test for stationarity for both real GDP growth and the CLI by using the Augmented Dickey-Fuller test. A 5% level of significance is used for all tests throughout the study. A stationary series provides for a more accurate random forest model, so this testing is necessary since random forest is one of the algorithms tested (Zulkifli, 2019).

Test for Stationarity - Augmented Dickey-Fuller (ADF) Test for Real GDP Growth:

The Augmented Dickey-Fuller test is used to test for stationarity of a data series. To do this test, one regresses the first difference (delta) of the dependent variable on its lagged value (the value of that variable one time period prior to the current one) and also on a certain number of lags of the first difference of the independent variable (van Dijk, Franses, Heij, 2019). The number of lags of the independent variable to test can be determined with the following rule of thumb: Set a maximum value for the lag length, and estimate the test regression with that lag length. If the the absolute value of the last lagged value in the test regression is less than 1.6, then reduce the lag length by one and retest (Ng and Perron, 2001.).

Model for ADF test: delta_real_gdp_growth = alpha_adf_1 + rho*real_gdp_growth_lag + gamma_adf_1*delta_real_gdp_growth_lag + gamma_adf_2*delta_real_gdp_growth_lag2 + gamma_adf_3*delta_real_gdp_growth_lag3 + gamma_adf_4*delta_real_gdp_growth_lag4 + epsilon_adf_1

For GDP growth, I start with the ADF test for 4 lags of delta_real_gdp_growth. This is because GDP growth can affect its value 4 quarters from now, but to affect longer-dated values would be less common.

ADF test for lags 1-4 of delta_real_gdp_growth:

```
##
## Time series regression with "ts" data:
## Start = 2, End = 102
##
## Call:
## dynlm(formula = delta_real_gdp_growth ~ real_gdp_growth_lag +
##     delta_real_gdp_growth_lag + delta_real_gdp_growth_lag2 +
##     delta_real_gdp_growth_lag3 + delta_real_gdp_growth_lag4,
##     data = combined_data_train_ts)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9956 -1.3521 -0.1175  1.2718  4.5747
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  1.76956    0.50070   3.534 0.000634 ***
## real_gdp_growth_lag         -0.53862    0.13594  -3.962 0.000144 ***
## delta_real_gdp_growth_lag   -0.12814    0.13184  -0.972 0.333549
## delta_real_gdp_growth_lag2   0.16013    0.12258   1.306 0.194588
## delta_real_gdp_growth_lag3   0.02429    0.11821   0.205 0.837651
## delta_real_gdp_growth_lag4   0.13147    0.09355   1.405 0.163168
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.055 on 95 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.3914, Adjusted R-squared:  0.3594
## F-statistic: 12.22 on 5 and 95 DF,  p-value: 3.752e-09
```

Conclusion: ADF test should be repeated with lag length of 3, as the absolute value of the t statistic of the last lagged value is less than 1.6.

ADF test with lags 1-3 of delta_real_gdp_growth:

```
##
## Time series regression with "ts" data:
## Start = 1, End = 102
##
## Call:
## dynlm(formula = delta_real_gdp_growth ~ real_gdp_growth_lag +
##     delta_real_gdp_growth_lag + delta_real_gdp_growth_lag2 +
##     delta_real_gdp_growth_lag3, data = combined_data_train_ts)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.357 -1.235 -0.223  1.413  5.569
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  1.89679    0.46810   4.052 0.000102 ***
## real_gdp_growth_lag         -0.56110    0.12572  -4.463 2.18e-05 ***
## delta_real_gdp_growth_lag   -0.12893    0.12471  -1.034 0.303802
## delta_real_gdp_growth_lag2   0.16280    0.11993   1.357 0.177777
## delta_real_gdp_growth_lag3   0.01288    0.09569   0.135 0.893184
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.105 on 97 degrees of freedom
## Multiple R-squared:  0.374,  Adjusted R-squared:  0.3482
## F-statistic: 14.49 on 4 and 97 DF,  p-value: 2.611e-09
```

Conclusion: The ADF test should be repeated with lag length 2, as the absolute value of the t statistic of the last lagged value is less than 1.6.

ADF test with lags 1 and 2 of delta_real_gdp_growth:

```
##
## Time series regression with "ts" data:
## Start = 1, End = 102
##
## Call:
## dynlm(formula = delta_real_gdp_growth ~ real_gdp_growth_lag +
##     delta_real_gdp_growth_lag + delta_real_gdp_growth_lag2, data = combined_data_train_ts)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.3245 -1.2418 -0.2145  1.3985  5.5503
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   1.8783     0.4453   4.218 5.49e-05 ***
## real_gdp_growth_lag          -0.5553     0.1174  -4.729 7.57e-06 ***
## delta_real_gdp_growth_lag    -0.1331     0.1202  -1.108    0.271
## delta_real_gdp_growth_lag2    0.1541     0.1006   1.532    0.129
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.094 on 98 degrees of freedom
## Multiple R-squared:  0.3738, Adjusted R-squared:  0.3547
## F-statistic:  19.5 on 3 and 98 DF,  p-value: 5.39e-10
```

Conclusion: The ADF test should be repeated with lag length 2, as the absolute value of the t statistic of the last lagged value is less than 1.6.

ADF test with lag 1 of delta_real_gdp_growth:

```
##
## Time series regression with "ts" data:
## Start = 1, End = 102
##
## Call:
## dynlm(formula = delta_real_gdp_growth ~ real_gdp_growth_lag +
##     delta_real_gdp_growth_lag, data = combined_data_train_ts)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.6824 -1.2279 -0.0702  1.3521  6.2245
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 1.63892    0.41977   3.904 0.000173 ***
## real_gdp_growth_lag        -0.48381    0.10849  -4.460 2.17e-05 ***
## delta_real_gdp_growth_lag -0.24055    0.09824  -2.448 0.016105 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.108 on 99 degrees of freedom
## Multiple R-squared:  0.3589, Adjusted R-squared:  0.3459
## F-statistic: 27.71 on 2 and 99 DF,  p-value: 2.782e-10
```

Conclusion: The t value of real_gdp_growth_lag is -4.46, which is below the critical value of -2.9, so we reject the null hypothesis of non-stationarity of real GDP growth. Real GDP growth is stationary.

Test for Stationarity - Augmented Dickey-Fuller Test for CLI:

Model for ADF test: delta_CLI = alpha_adf_2 + rho1*CLI_lag + beta_adf_1*delta_CLI_lag + beta_adf_2*delta_CLI_lag_2 + beta_adf_3*delta_CLI_lag3 + beta_adf_4*delta_CLI_lag4 + epsilon_adf_2

Note: Starting with ADF test for 4 lags of delta_CLI, since it is most commonly used as a predictor of economic conditions in the following 3 quarters. I will test with 1 extra quarter to be safe.

ADF test with lags 1-4 of delta_CLI:

```
##
## Time series regression with "ts" data:
## Start = 2, End = 102
##
## Call:
## dynlm(formula = delta_CLI ~ CLI_lag + delta_CLI_lag + delta_CLI_lag2 +
##     delta_CLI_lag3 + delta_CLI_lag4, data = combined_data_train_ts)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.40386 -0.09733  0.03398  0.09947  0.38336
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     8.02324    2.84441   2.821  0.00583 **
## CLI_lag        -0.08017    0.02844  -2.819  0.00587 **
## delta_CLI_lag   1.34608    0.09096  14.799  < 2e-16 ***
## delta_CLI_lag2 -1.12394    0.15307  -7.343 7.12e-11 ***
## delta_CLI_lag3  0.69717    0.14576   4.783 6.31e-06 ***
## delta_CLI_lag4 -0.31478    0.09562  -3.292  0.00140 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1655 on 95 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.7874, Adjusted R-squared:  0.7762
## F-statistic: 70.38 on 5 and 95 DF,  p-value: < 2.2e-16
```

Conclusion: ADF test should be repeated with larger lag length, as the absolute value of the t statistic of the last lagged value is greater than 1.6.

ADF test with lags 1-8 of delta_CLI:

```
##
## Time series regression with "ts" data:
## Start = 6, End = 102
##
## Call:
## dynlm(formula = delta_CLI ~ CLI_lag + delta_CLI_lag + delta_CLI_lag2 +
##     delta_CLI_lag3 + delta_CLI_lag4 + delta_CLI_lag5 + delta_CLI_lag6 +
##     delta_CLI_lag7 + delta_CLI_lag8, data = combined_data_train_ts)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.35624 -0.08761  0.01407  0.09182  0.32599
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     9.39082    3.51226   2.674 0.008958 **
## CLI_lag        -0.09391    0.03513  -2.673 0.008968 **
## delta_CLI_lag   1.51785    0.10635  14.273  < 2e-16 ***
## delta_CLI_lag2 -1.51069    0.19704  -7.667 2.34e-11 ***
## delta_CLI_lag3  1.36290    0.25554   5.333 7.53e-07 ***
## delta_CLI_lag4 -1.09549    0.28840  -3.799 0.000269 ***
## delta_CLI_lag5  0.72931    0.28387   2.569 0.011899 *
## delta_CLI_lag6 -0.35907    0.25844  -1.389 0.168262
## delta_CLI_lag7  0.15967    0.18962   0.842 0.402058
## delta_CLI_lag8 -0.01672    0.10739  -0.156 0.876626
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1608 on 87 degrees of freedom
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.7882, Adjusted R-squared:  0.7663
## F-statistic: 35.98 on 9 and 87 DF,  p-value: < 2.2e-16
```

Conclusion: The ADF test should be repeated with lag length 7, as the absolute value of the t statistic of the last lagged value is less than 1.6.

ADF test with lags 1-7 of delta_CLI:

```
##
## Time series regression with "ts" data:
## Start = 5, End = 102
##
## Call:
## dynlm(formula = delta_CLI ~ CLI_lag + delta_CLI_lag + delta_CLI_lag2 +
##     delta_CLI_lag3 + delta_CLI_lag4 + delta_CLI_lag5 + delta_CLI_lag6 +
##     delta_CLI_lag7, data = combined_data_train_ts)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.35706 -0.08784  0.01417  0.09271  0.33126
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     9.75431    3.26063   2.992  0.00359 **
## CLI_lag        -0.09754    0.03261  -2.991  0.00359 **
## delta_CLI_lag   1.52319    0.10468  14.552  < 2e-16 ***
## delta_CLI_lag2 -1.50500    0.18916  -7.956 5.34e-12 ***
## delta_CLI_lag3  1.36163    0.24354   5.591 2.45e-07 ***
## delta_CLI_lag4 -1.07646    0.25406  -4.237 5.51e-05 ***
## delta_CLI_lag5  0.71591    0.24371   2.937  0.00421 **
## delta_CLI_lag6 -0.33872    0.18413  -1.840  0.06917 .
## delta_CLI_lag7  0.13930    0.10513   1.325  0.18856
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1592 on 89 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.7883, Adjusted R-squared:  0.7693
## F-statistic: 41.44 on 8 and 89 DF,  p-value: < 2.2e-16
```

Conclusion: The ADF test should be repeated with lag length 6, as the absolute value of the t statistic of the last lagged value is less than 1.6.

ADF test with lags 1-6 of delta_CLI:

```
##
## Time series regression with "ts" data:
## Start = 4, End = 102
##
## Call:
## dynlm(formula = delta_CLI ~ CLI_lag + delta_CLI_lag + delta_CLI_lag2 +
##     delta_CLI_lag3 + delta_CLI_lag4 + delta_CLI_lag5 + delta_CLI_lag6,
##     data = combined_data_train_ts)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38329 -0.08343  0.00964  0.09349  0.35136
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     8.67206    3.08559   2.811  0.00606 **
## CLI_lag        -0.08670    0.03086  -2.810  0.00607 **
## delta_CLI_lag   1.49128    0.10076  14.800  < 2e-16 ***
## delta_CLI_lag2 -1.43846    0.18305  -7.858 7.52e-12 ***
## delta_CLI_lag3  1.21500    0.21529   5.644 1.87e-07 ***
## delta_CLI_lag4 -0.90347    0.22006  -4.105 8.79e-05 ***
## delta_CLI_lag5  0.48536    0.17486   2.776  0.00669 **
## delta_CLI_lag6 -0.13295    0.10295  -1.291  0.19982
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1593 on 91 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.7874, Adjusted R-squared:  0.771
## F-statistic: 48.14 on 7 and 91 DF,  p-value: < 2.2e-16
```

Conclusion: The ADF test should be repeated with lag length 5, as the absolute value of the t statistic of the last lagged value is less than 1.6.

ADF test with lags 1-5 of delta_CLI:

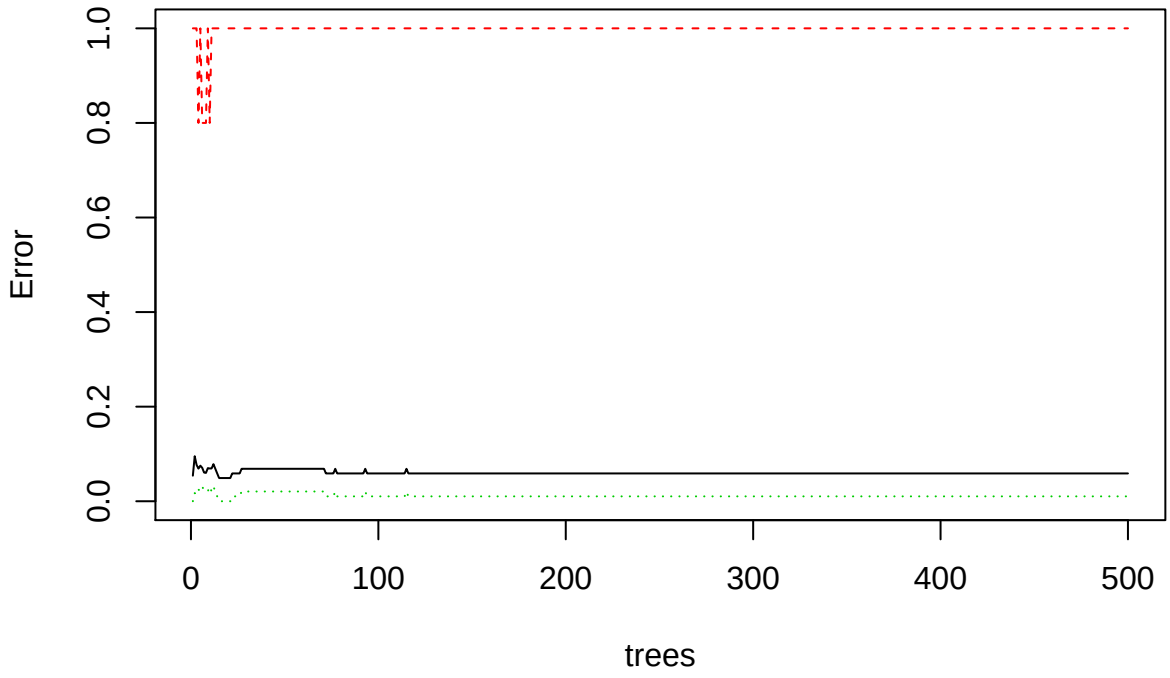
```
##
## Time series regression with "ts" data:
## Start = 3, End = 102
##
## Call:
## dynlm(formula = delta_CLI ~ CLI_lag + delta_CLI_lag + delta_CLI_lag2 +
##     delta_CLI_lag3 + delta_CLI_lag4 + delta_CLI_lag5, data = combined_data_train_ts)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.37213 -0.07876  0.01006  0.09500  0.38067
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    10.12158    2.87280   3.523 0.000663 ***
## CLI_lag        -0.10120    0.02873  -3.523 0.000664 ***
## delta_CLI_lag   1.46598    0.09752  15.033  < 2e-16 ***
## delta_CLI_lag2 -1.32568    0.16081  -8.244 1.05e-12 ***
## delta_CLI_lag3  1.08252    0.18711   5.785 9.66e-08 ***
## delta_CLI_lag4 -0.70906    0.15716  -4.512 1.88e-05 ***
## delta_CLI_lag5  0.30051    0.09787   3.070 0.002802 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.159 on 93 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.7933, Adjusted R-squared:   0.78
## F-statistic:  59.5 on 6 and 93 DF,  p-value: < 2.2e-16
```

The t stat of CLI_lag is -3.523, which is below the critical value of -2.9, so we reject the null hypothesis of non-stationarity. CLI is stationary.

Model Creation:
Next I created a logistic regression model for lags 1, 2, and 3 of the CLI and also lags 1, 2, and 3 of GDP growth. However, the algorithm did not converge when lag 2 of GDP growth was included, so this was excluded. Since we will also be obtaining predictions from 2 other algorithms and creating an ensemble prediction, I think the exclusion of this lag 2 variable will still obtain a good prediction.

A knn algorithm is then created to model economic growth or contraction, and this model is next optimized for the best value of k. Values between 2 and 25 were tested, and a value of 2 was found to be the optimal k. Finally, a random forest model is created. In order to get an idea of how many trees to have in my random forest model, I generated a plot of the model, from which we can see that the algorithm converges after no more than 100 trees (Figure 6). The random forest algorithm with 100 trees is then optimized for the best value of mtry. Mtry values between 1 and 6 were tested, and the optimal value was found to be 1.

Figure 6: Initial RF Model Error vs. Number of Trees



Results: After testing the performance against the test set for each model, we can see that logistic regression is the best-performing model, with an accuracy of 93.3% and a F1 score of 0.727. This model had perfect specificity, while having the best sensitivity at 0.57, with the positive class being a "0", or economic contraction for all models.

The second-best performing models were the ensemble prediction and the random forest models, with an accuracy of 84.4% and an F1 score of 0. It is interesting to note that these models did not predict a single period of contraction and had a sensitivity of 0, so there is room for improvement. In order to give a tiebreaker between these two models, I would rank the random forest model over the ensemble, as it gives the same results but is much simpler than the complex ensemble model.

The knn model was the worst-performing model, with an accuracy of 82.2% and a F1 score of 0. This model had good specificity (.947) but a sensitivity of 0, so again there is room for improvement.

Below are shown the confusion matrices for the models.
Logistic Regression model:

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0  4  0
##          1  3 38
##
##                Accuracy : 0.9333
##                  95% CI : (0.8173, 0.986)
##     No Information Rate : 0.8444
##     P-Value [Acc > NIR] : 0.0653
##
##                   Kappa : 0.6925
##
##  Mcnemar's Test P-Value : 0.2482
##
##             Sensitivity : 0.57143
##             Specificity : 1.00000
##          Pos Pred Value : 1.00000
##          Neg Pred Value : 0.92683
##              Prevalence : 0.15556
##          Detection Rate : 0.08889
##    Detection Prevalence : 0.08889
##       Balanced Accuracy : 0.78571
##
##        'Positive' Class : 0
##
```

KNN Model:

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0  0  1
##          1  7 37
##
##                Accuracy : 0.8222
##                  95% CI : (0.6795, 0.92)
##     No Information Rate : 0.8444
##     P-Value [Acc > NIR] : 0.7409
##
##                   Kappa : -0.0405
##
##  Mcnemar's Test P-Value : 0.0771
##
##             Sensitivity : 0.00000
##             Specificity : 0.97368
##          Pos Pred Value : 0.00000
##          Neg Pred Value : 0.84091
##              Prevalence : 0.15556
##          Detection Rate : 0.00000
##    Detection Prevalence : 0.02222
##       Balanced Accuracy : 0.48684
##
##        'Positive' Class : 0
##
```

Random Forest Model:

```
## Warning in confusionMatrix.default(factor(y_hat_rf_optimized), reference =
## factor(combined_data_test$gdp_impr)): Levels are not in the same order for
## reference and data. Refactoring data to match.
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0  0  0
##          1  7 38
##
##                Accuracy : 0.8444
##                  95% CI : (0.7054, 0.9351)
##     No Information Rate : 0.8444
##     P-Value [Acc > NIR] : 0.59907
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : 0.02334
##
##             Sensitivity : 0.0000
##             Specificity : 1.0000
##          Pos Pred Value :    NaN
##          Neg Pred Value : 0.8444
##              Prevalence : 0.1556
##          Detection Rate : 0.0000
##    Detection Prevalence : 0.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : 0
##
```

Ensemble:

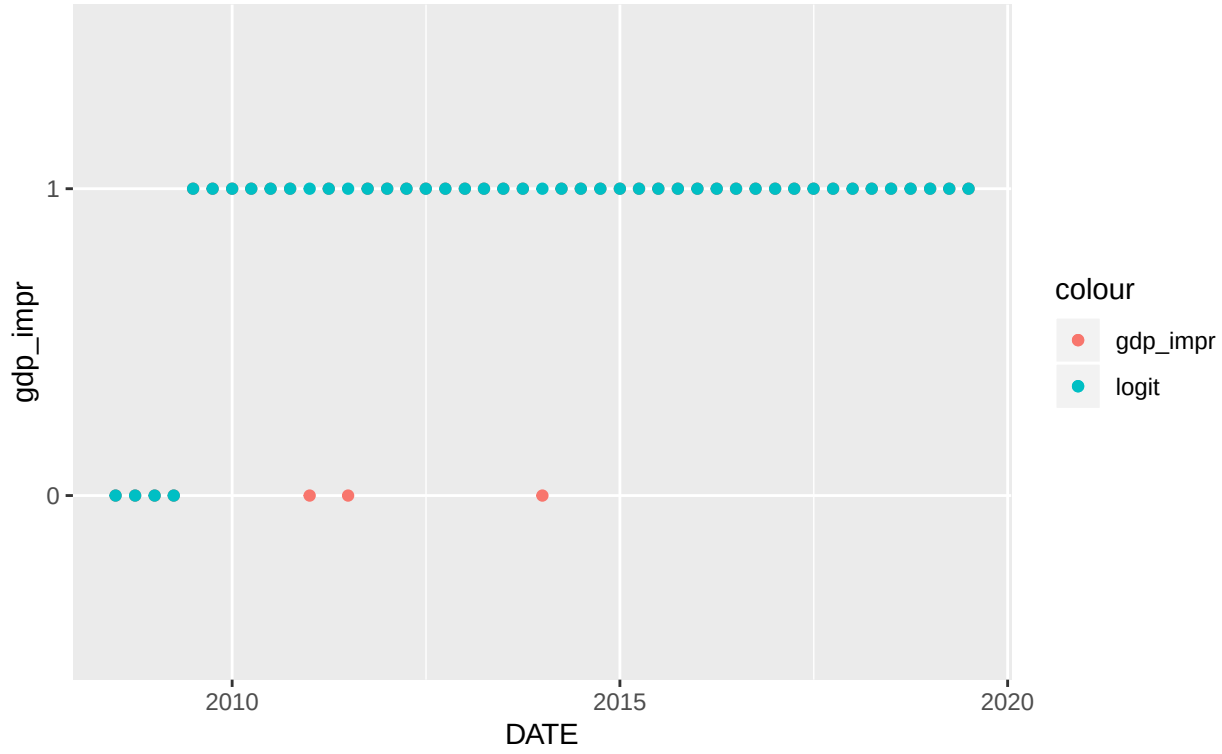
```
## Warning in confusionMatrix.default(factor(y_hat_ensemble), reference =
## factor(combined_data_test$gdp_impr)): Levels are not in the same order for
## reference and data. Refactoring data to match.
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0  0  0
##          1  7 38
##
##                Accuracy : 0.8444
##                  95% CI : (0.7054, 0.9351)
##     No Information Rate : 0.8444
##     P-Value [Acc > NIR] : 0.59907
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : 0.02334
##
##             Sensitivity : 0.0000
##             Specificity : 1.0000
##          Pos Pred Value :    NaN
##          Neg Pred Value : 0.8444
##              Prevalence : 0.1556
##          Detection Rate : 0.0000
##    Detection Prevalence : 0.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : 0
##
```

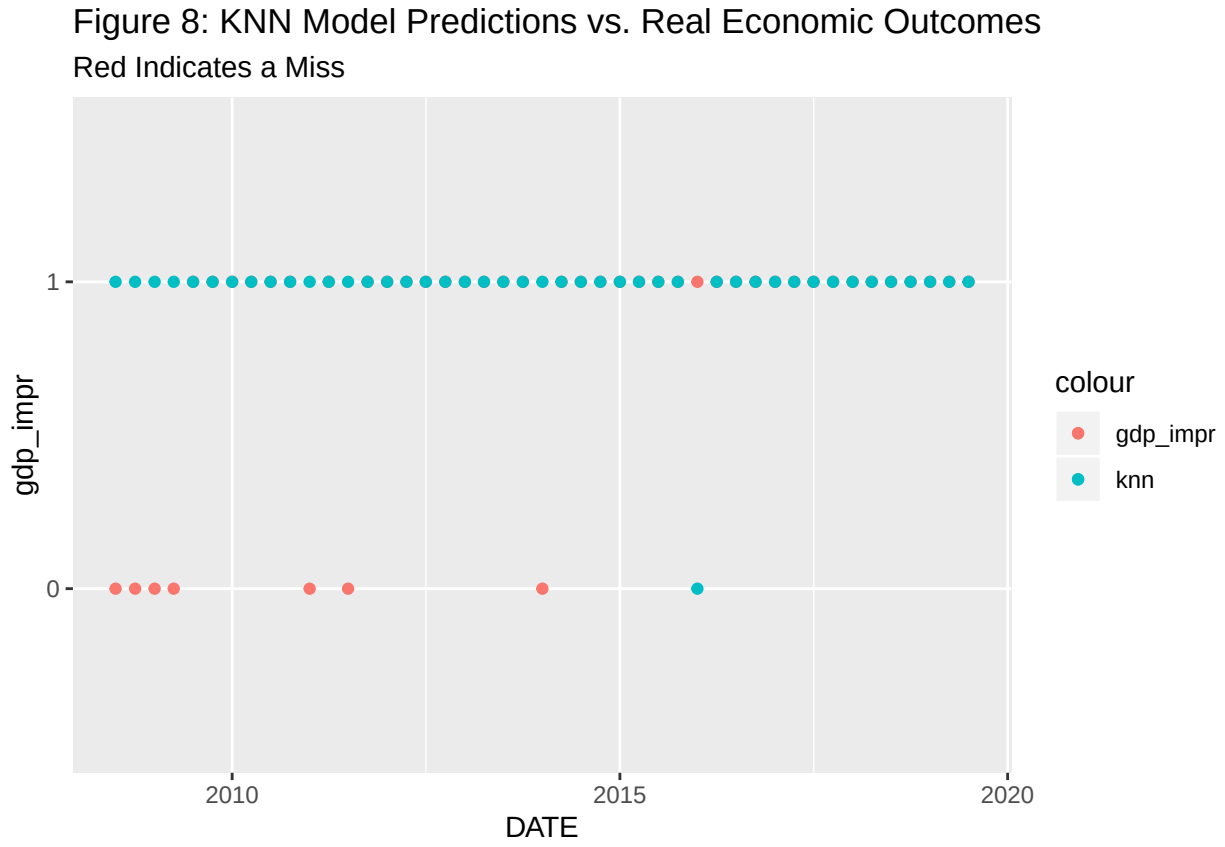
Below are shown plots of the predictions against real economic outcomes for each model. Logistic Regression (Figure 7):

Figure 7: Logit Model Predictions vs. Real Economic Outcomes

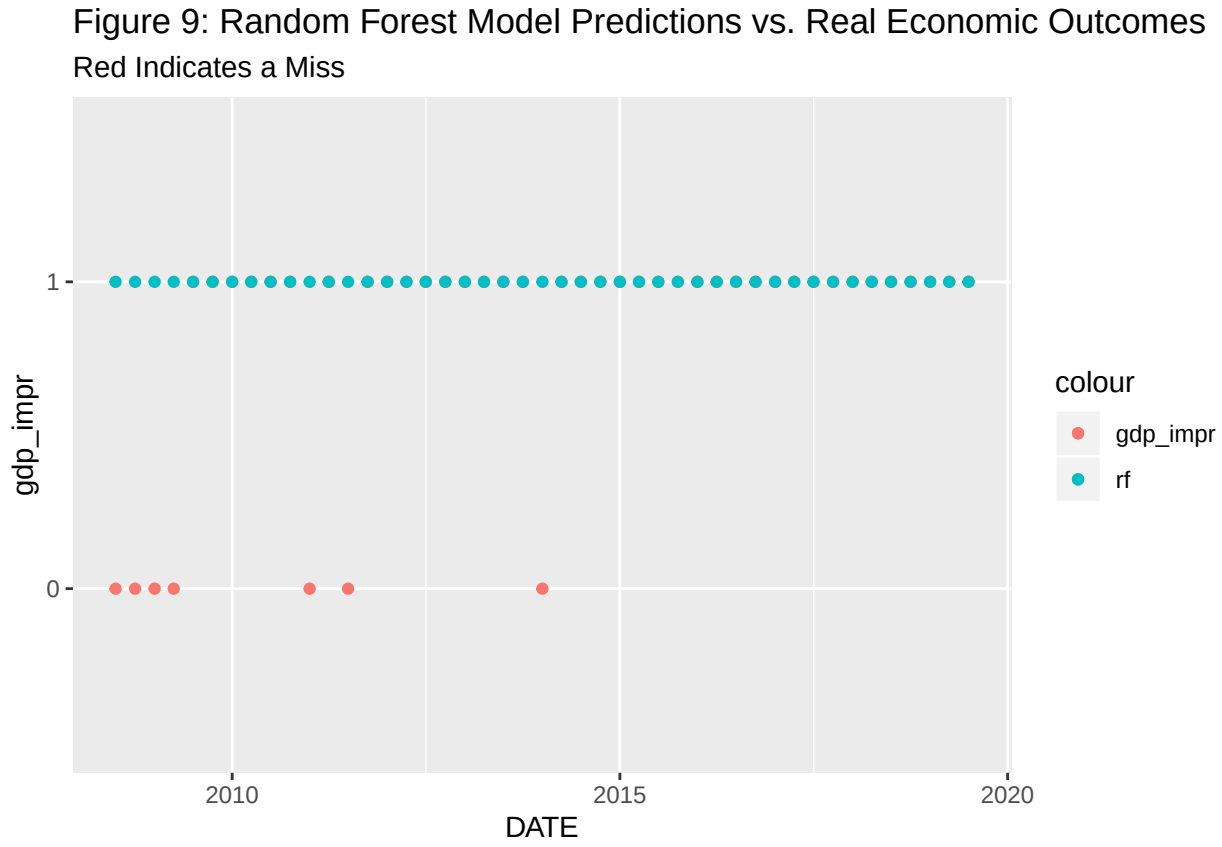
Red Indicates a Miss



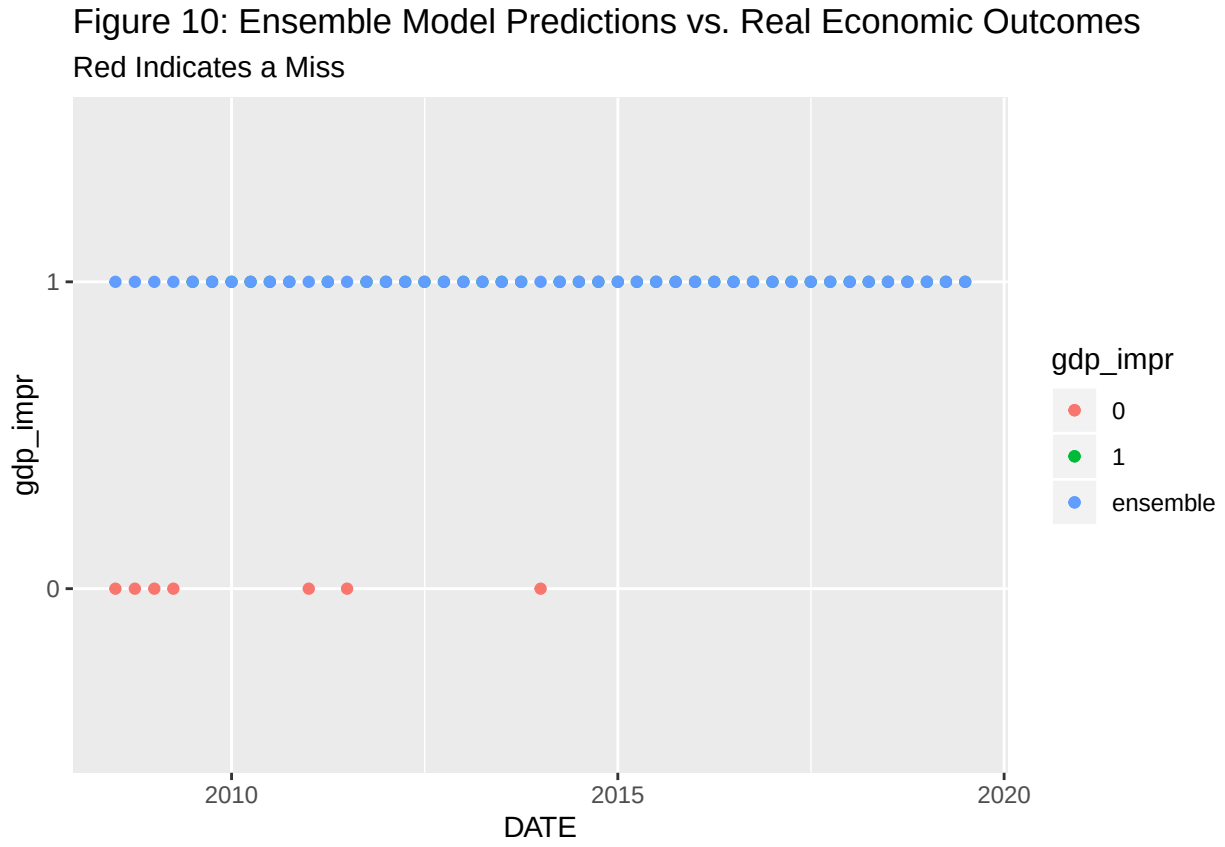
KNN Model (Figure 8):



Random Forest (Figure 9):



Ensemble (Figure 10):



Conclusion:

Models based on the lagged values of CLI and real GDP growth were developed using the training portion of the dataset, and these were tested on the test set. The best-performing model was found to have an accuracy of 0.933 and an F1 score of .73. This indicates that the model did a good job of predicting whether or not there would be economic growth from July 2008 - July 2019. This is better than a strategy of simply assuming that there will be economic growth in every period, which would yield an accuracy of 84.4%.

While the model accuracies are acceptable, there are limitations to these models. The logistic regression model did not converge when lag 2 of real GDP growth was included in the model, and while the logistic regression still performed very well, it would be interesting to think about what the results might be if this variable had been included. A potential area for future work involves using a regularized logistic regression model instead in order to prevent overfitting. Another limitation of this research was the relatively poor accuracy of the knn model on the test set, which may have been due to overtraining since I selected a low value of k. I selected a low value of k in order to be able to detect economic contraction, which is relatively uncommon. However, a minimum value of 3 instead of 2 may give better results on the test set as an area for future research. The random forest model also was not as accurate as I hoped, and the F1 scores of both the random forest model and the knn model were zero, so there is further research to be done with these models on a larger data set.

References:

Ben Hamner, and Michael Frasco. Metrics: Evaluation Metrics for Machine Learning (version R package version 0.1.4), 2018. https://CRAN.R-project.org/package=Metrics.

"Composite Leading Indicators (CLI) Frequently Asked Questions (FAQs)." OECD, 2019. Dijk, Dick van, Philip Hans Franses, and Christiaan Heij. "Lecture 6.3 on Time Series: Specification and Estimation." n.d.

Dowle, Matt, and Arun Srinivasan. Data.Table: Extension of "Data.Frame" (version R package version 1.12.4), 2019. https://CRAN.R-project.org/package=data.table.

Grolemund, Garrett, and Hadley Wickham. "Dates and Times Made Easy with Lubridate." Journal of Open Source Software 40, no. 3 (2011): 1–25.

Hess, Florian. Using R for Introductory Econometrics, 2016. http://www.URfIE.net.

Irizarry, Rafael. Knn, 2019. ———. Logistic Regression, 2019. ———. Overtraining and Oversmoothing, n.d. ———. Random Forests, 2019.

Kuhn, Max. Caret: Classification and Regression Training (version R package version 6.0-84), 2019. https://CRAN.R-project.org/package=caret.

Liaw, A, and M Weiner. "Classification and Regression by Random Forest." R News, 2002.

Nau, Robert. "Stationarity and Differencing," 2019. https://people.duke.edu/~rnau/411diff.htm.

Ng, Serena, and Pierre Perron. "Lag Length Selection and the Construction of Unit Root Tests with Good Size and Power." Econometrica 69, no. 6 (2001): 1519–54.

R Core Team. R: A Language and Environment for Statistical Computing. Vienna, Austria, 2019. https://www.R-project.org/.

Sarkar, Deepayan, and Felix Andrews. LatticeExtra: Extra Graphical Utilities Based on Lattice (version R package version 0.6-29), 2019. https://CRAN.R-project.org/package=latticeExtra.

Wickham et al. "Welcome to the Tidyverse." Journal of Open Source Software 4, no. 43 (2019): 1686.

Wickham, Hadley, Jim Hester, and Romain Francois. Readr: Read Rectangular Text Data (version 1.3.1), 2018. https://CRAN.R-project.org/package=readr.

Zeileis, A. Dynlm: Dynamic Linear Regression (version 0.3-6), 2019. https://CRAN.R-project.org/package=dynlm.

Zulkifli, Hafidz. "Multivariate Time Series Modeling Using Random Forest." Towards Data Science, March 31, 2019. https://towardsdatascience.com/multivariate-time-series-forecasting-using-random-forest-2372f3ecbad1.