Analysis and prediction of the impact of COVID-19 on the global economy

using

FLOSS - R

submitted by

Tanmay Srinath (BMSCE, Bangalore)

under the guidance of

Prof. Radhendushka Srivastava
Department of Mathematics
IIT Bombay

15 October 2021

Contents

Chapter 1

Abstract

COVID-19 has had a profound impact on the lives of each individual. However, for countries as a whole, the economy has taken the hardest hit. This case study aims to analyze the Gross Domestic Product (GDP) and employment data of countries worldwide during the COVID-19 pandemic using statistical tools to predict their GDP loss. Predictions were made using factors like sector-wise GDP loss and sector-wise employment loss by applying multiple linear regression. Each data variable was tested for its significance to the regression model performance, and those with insignificant contributions were removed. Later, the regression results from the remaining variables were analyzed and explained. Then, a neural network was trained over the selected variables' data to compare its prediction results against the linear model. The comparison gave an insight into whether the data was suitable for a linear model or a non-linear model. Finally, the best among the two was chosen based on the prediction error percentage.

Chapter 2

Introduction

Gross Domestic Product or GDP, is a monetary measure of the market value of all the final goods and services produced in a specific period [1]. GDP is often used as a metric for international comparisons and a broad measure of economic progress. It is widely considered one of the most powerful statistical indicators of national development [2]. Therefore, it is natural to study the impact of COVID-19 on a country's GDP to get an overall picture of the damage caused by it. In this case study, multiple linear regression and artificial neural network were used to predict the GDP loss of a country caused due to the pandemic. Each of the proposed statistical methods had its advantage; for example, the white-box linear regression model was beneficial in determining insignificant variables, whereas the neural network had a relatively more minor prediction error.

Chapter 3

Analysis and Results

3.1 Data Collection

The COVID-19 Economic Impact Assessment data was collected for this case study from an online repository known as the ADB Data Library. The ADB Data Library is a platform that hosts publicly available data from the Asian Development Bank. The data obtained contains a measure of the potential economy and sector specific impact of the COVID-19 outbreak [3].

3.2 Data Exploration

The original dataset had the dimension 1566 x 10 and contained the following columns:

- **Economy**: Contains the country name.
- **ADB Country Code**: Contains the country code as assigned by ADB.
- **Sector**: Contains the economic sector from where the data was collected.
- **Country 2018 GDP**: Contains a country's GDP for the year 2018.
- **Scenario**: Contains the scenario based on which the GDP drop is predicted.
- **as % of total GDP**: Contains the GDP loss as a percentage of the total GDP.
- **in $ Mn**: Contains the total income in denominations of $1 million.
- **Employment (in 000)**: Contains total number of people employed in counts of 1000s.
- **as % of sector GDP**: Contains percentage of sector GDP loss.
- **as % of sector employment**: Contains percentage of sector employment loss.

For further examination, the R programming language was used. R is a language and environment for statistical computing and graphics [4]. The dataset was imported into the R environment using the *read_xlsx*() function of the *readxl* [5] package. Then a summary of the dataset was generated using the *skim*() function of the *skimr* [6] package as shown in Figure 3.1.

```
-- Data Summary ------------------------
                           Values
Name                       dforig_large
Number of rows             1576
Number of columns          10
_____________________
Column type frequency:
  character                9
  numeric                  1
_____________________
Group variables            None

-- Variable type: character ------------------------------------------------------------------------------------------------------------------------
# A tibble: 9 x 8
  skim_variable            n_missing complete_rate   min   max empty n_unique whitespace
* <chr>                        <int>         <dbl> <int> <int> <int>    <int>      <int>
1 Economy                          0             1     4    32     0       62          0
2 ADB Country Code                 0             1     3     3     0       62          0
3 Sector                           0             1     4    54     0        8          0
4 Scenario                         0             1    39    65     0        4          0
5 as % of total GDP                0             1     2     6     0      317          0
6 in $ Mn                          0             1     2    12     0     1008          0
7 Employment (in 000)              0             1     1     5     0      380          0
8 as % of sector GDP               0             1     2     6     0      598          0
9 as % of sector employment        0             1     2     6     0      586          0

-- Variable type: numeric --------------------------------------------------------------------------------------------------------------------------
# A tibble: 1 x 11
  skim_variable    n_missing complete_rate     mean       sd    p0    p25     p50      p75     p100 hist
* <chr>                <int>         <dbl>    <dbl>    <dbl> <dbl>  <dbl>   <dbl>    <dbl>    <dbl> <chr>
1 Country 2018 GDP         0             1 1227601. 3105474. 2535. 60972. 330910. 1042173. 20544343. ▇▁▁▁▁
```

Figure 3.1: Summary of the original dataset.

From the summary, it was observed that the dataset contained nine character-type data columns and only one numeric-type data column which was incorrect as the columns *"as % of total GDP", "in $ Mn", "Employment (in 000)", "as % of sector GDP",* and *"as % of sector employment"* must be of numeric type. Therefore, it was necessary to convert the data columns to appropriate format before performing any statistical analysis.

3.3 Data Reformatting and Cleaning

Since numeric values were incorrectly represented as characters, the following chunk of code was executed to reformat the data:

```
dforig_large$`in $ Mn`=sapply(dforig_large$`in $ Mn`, replaceCommas<-function(x){
  x<-as.numeric(gsub("\\,", "", x))
})
dforig_large[,c(6:10)]=sapply(dforig_large[,c(6:10)],as.numeric)
dforig_large$`Country 2018 GDP`=sapply(dforig_large$`Country 2018 GDP`,as.numeric)
```

After making changes, the column names were replaced with terms that could easily explain the data associated with the respective column.

```
colnames(dforig_large)=c("Country Name", "ADB Country Code", "Sector", "2018 GDP",
    "Scenario", "Total GDP Loss", "In Million", "Employment in 1000s", "Sector GDP
    Loss", "Sector Employment Loss")
```

After changing the column names, the dataset's summary was again generated and analyzed as shown in Figure 3.2.

```
-- Data Summary ------------------------
                           Values
Name                       dforig_large
Number of rows             1576
Number of columns          10
___________________________
Column type frequency:
  character                4
  numeric                  6
___________________________
Group variables            None

-- Variable type: character ------------------------------------------------------------------------------------------------------------------
# A tibble: 4 x 8
  skim_variable    n_missing complete_rate   min   max empty n_unique whitespace
* <chr>                <int>         <dbl> <int> <int> <int>    <int>      <int>
1 Country Name             0             1     4    32     0       62          0
2 ADB Country Code         0             1     3     3     0       62          0
3 Sector                   0             1     4    54     0        8          0
4 Scenario                 0             1    39    65     0        4          0

-- Variable type: numeric --------------------------------------------------------------------------------------------------------------------
# A tibble: 6 x 11
  skim_variable          n_missing complete_rate       mean         sd      p0      p25       p50       p75       p100 hist
* <chr>                      <int>         <dbl>      <dbl>      <dbl>   <dbl>    <dbl>     <dbl>     <dbl>      <dbl> <chr>
1 2018 GDP                       0             1    1227601.   3105474.   2535.   60972.   330910.   1042173.   20544343. ▇▁▁▁▁
2 Total GDP Loss               524             0.668     -1.16       1.60  -14.2     -1.41     -0.51     -0.19        0 ▁▁▁▁▇
3 In Million                   524             0.668  13505.     65453.      0.42   181.     1092.     5071.    1150160. ▇▁▁▁▁
4 Employment in 1000s          568             0.640    483.      2721.      0        8        43       186.      52784  ▇▁▁▁▁
5 Sector GDP Loss              524             0.668     -4.26       4.95  -45.3     -5.35     -2.7      -1.55       -0.04 ▁▁▁▁▇
6 Sector Employment Loss       568             0.640     -4.27       5.03  -61.5     -5.4      -2.70     -1.55       -0.03 ▁▁▁▁▇
```

Figure 3.2: Summary of the reformatted dataset.

It can be observed from the summary that the dataset contained missing values. Those missing values were later removed using the *na.omit()* function as shown below:

```
dforig = na.omit(dforig_large)
```

The cleaned dataset contained 1008 row entries. Now the dataset need to go through various preprocessing steps to make it suitable for analysis.

3.4 Data Preprocessing

To make the data suitable for analysis, the following steps were performed:

- Rephrasing the *Scenario* column string values for easier understanding.

```
dforig$Scenario=sapply(dforig$Scenario, function(x) gsub("Additional impact
    under Longer containment, larger demand shock", "Additional impact (Long
    term)", x))
dforig$Scenario=sapply(dforig$Scenario, function(x) gsub("Additional impact
    under Shorter containment, smaller demand shock", "Additional impact (
    Short term)", x))
dforig$Scenario=sapply(dforig$Scenario, function(x) gsub("Shorter containment
    , smaller demand shock", "Short term effects ", x))
dforig$Scenario=sapply(dforig$Scenario, function(x) gsub("Longer containment,
     larger demand shock", "Long term effects", x))
```

- Rephrasing the *Sector* column string values for easier understanding.

```
dforig$Sector=sapply(dforig$Sector, function(x) gsub("Agriculture, Mining and
     Quarrying", "Agriculture", x))
dforig$Sector=sapply(dforig$Sector, function(x) gsub("Business, Trade,
    Personal, and Public Services", "Business and Trade", x))
dforig$Sector=sapply(dforig$Sector, function(x) gsub("Hotel and restaurants
    and Other Personal Services", "Hotels and Restaurant", x))
dforig$Sector=sapply(dforig$Sector, function(x) gsub("Light/Heavy
    Manufacturing, Utilities, and Construction", "Light/Heavy Manufacturing",
    x))
```

- Removing row entries containing the value *"_ALL"* in the *Sector* column.

```
dfnn=dfnn[-c(which(dfnn$Sector=="_All")),]
```

- Converting character variables to factors.

```
dforig$Sector=as.factor(dforig$Sector)
dforig$Scenario=as.factor(dforig$Scenario)
```

- Selecting relevant columns for data analysis.

```
dfnn=as_tibble(select(dforig,`Percentage of Total GDP Loss`,Sector,Scenario,`
    Percentage of Sector GDP Loss`,`Percentage of Sector Employment Loss`))
```

- Changing the data type of character columns containing numbers to numeric.

```
dfnn$`Percentage of Total GDP Loss`=sapply(dfnn$`Percentage of Total GDP Loss
    `, as.numeric)
```

- Removing row entries with non-negative value for percentage change in GDP to perform an independent analysis.

```
df_pos_gdp=dfnn[which(dfnn$`Total GDP Loss`>=0),]
dfnn=dfnn[-c(which(dfnn$`Total GDP Loss`>=0)),]
```

- Normalizing the dependent variable, *"Total GDP Loss"*, using logarithmic transformation.

```
nrmls=function(x)
{
  return (log(abs(x)))
}
dfnn$`Total GDP Loss`=sapply(dfnn$`Total GDP Loss`, nrmls)
```

- Logarithmic transformation may output some values as *"Inf"* or infinity. Hence, it is necessary to remove row entries containing infinite value (if any).

```
real_index=which(is.infinite(dfnn$`Percentage of Total GDP Loss`)==FALSE)
dfnn=dfnn[c(real_index),]
```

- Splitting data in a ratio of 3:1 for training and testing respectively, to apply the linear regression model over it.

```
## 75% for training data
smp_size <- floor(0.75* nrow(dfnn))
## Set random seed for reproducibility
set.seed(123)
## Creating a list of random training indices
train_ind <- sample(seq_len(nrow(dfnn)), size = smp_size)
## Segregating training and testing data
train <- dfnn[train_ind, ]
test <- dfnn[-train_ind, ]
```

The data preprocessing resulted in a normally distributed dependent variable as shown in Figure 3.3:

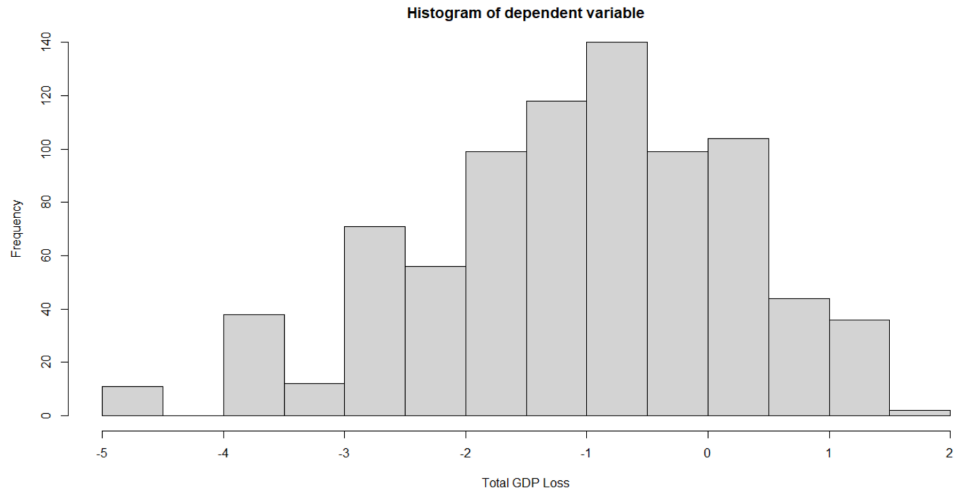

Figure 3.3: Histogram of the dependent variable.

3.5 Data Analysis

3.5.1 Linear Regression

Linear regression is a linear approach for modeling the relationship between a scalar response and one or more explanatory variables (also known as dependent and independent variables respectively). The case of a single explanatory variable is called simple linear regression; for more than one, the process is called multiple linear regression [7]. For the case study, multiple linear regression was utilized.

Original Model

The multiple linear regression model was trained to predict GDP loss, given the sector, scenario and sector-wise GDP loss. The command used to train the model is as follows:

```
1 lm_best=lm(‘Percentage of Total GDP Loss‘~‘Percentage of Sector GDP Loss‘*Sector+
      Scenario,data=train)
```

The following is the summary of the obtained linear model:

```
Call:
lm(formula = `Total GDP Loss` ~ `Sector GDP Loss` * Sector +
    Scenario, data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-3.6064 -0.3783  0.0942  0.4550  2.2762

Coefficients:
                                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                                    -2.182295   0.152073 -14.350  < 2e-16 ***
`Sector GDP Loss`                              -0.165087   0.036889  -4.475 9.11e-06 ***
SectorBusiness and Trade                        1.205159   0.173470   6.947 9.59e-12 ***
SectorHotels and Restaurant                    -0.030008   0.157658  -0.190 0.849108
SectorLight/Heavy Manufacturing                 0.669449   0.173703   3.854 0.000129 ***
SectorTransport services                       -0.047055   0.155710  -0.302 0.762607
ScenarioAdditional impact (Short term)         -0.342538   0.124331  -2.755 0.006044 **
ScenarioLong term effects                      -0.298746   0.102151  -2.925 0.003578 **
ScenarioShort term effects                     -0.244673   0.101781  -2.404 0.016519 *
`Sector GDP Loss`:SectorBusiness and Trade     -0.260295   0.049110  -5.300 1.62e-07 ***
`Sector GDP Loss`:SectorHotels and Restaurant   0.003093   0.038027   0.081 0.935200
`Sector GDP Loss`:SectorLight/Heavy Manufacturing -0.231902   0.051978  -4.462 9.69e-06 ***
`Sector GDP Loss`:SectorTransport services      0.035370   0.037311   0.948 0.343508
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.762 on 609 degrees of freedom
Multiple R-squared:  0.6711,    Adjusted R-squared:  0.6646
F-statistic: 103.5 on 12 and 609 DF,  p-value: < 2.2e-16
```

Figure 3.4: Summary of the obtained multiple linear regression model.

The results of the linear regression model were good but the contribution of some variables was insignificant which can be explained as follows:

- **Hotels and Restaurant (Sector)** - Restrictions on hotel and restaurant services were removed sooner than others, so that people could get food delivered to them as it could reduce the rate of the spread of infection. Hence, the associated data variable was quite useless in predicting the GDP drop.
- **Transport services (Sector)** - During the pandemic, there were restrictions imposed on the travel of citizens, within and outside a country. However, compared to other sectors, transport sector got disrupted only for a short time. Hence, it did not contribute much to the prediction of GDP loss.

The q-q (quantile-quantile) plot is a probability plot, which is a graphical method for comparing two probability distributions by plotting their quantiles against each other [8]. The q-q plot of the obtained linear model's residuals is given below:

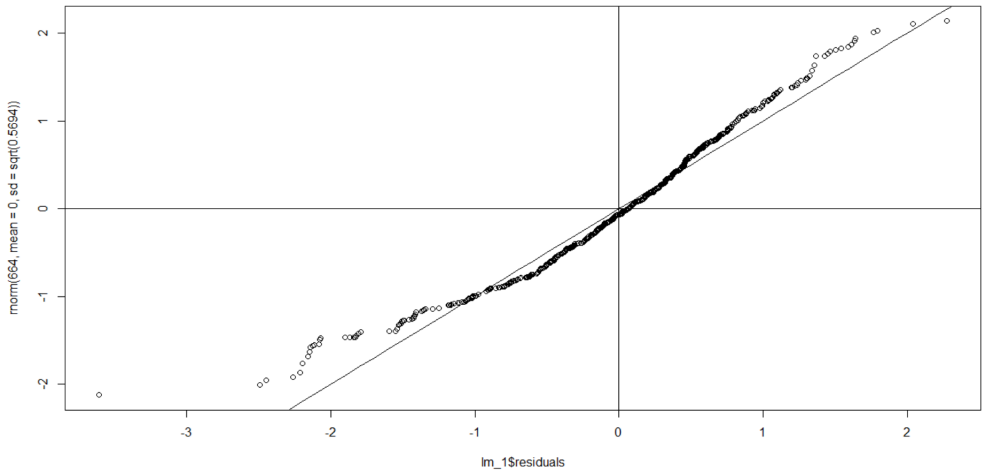

Figure 3.5: q-q plot of residuals.

It can be observed that the residuals approximately follow a normal distribution. Another method to check for normality is performing KS test over the residuals. The KS test is a non-parametric test that is used to decide if a sample comes from a population having a particular distribution [9]. Since the KS test is more precise as compared to the graphical method for testing normality, it was used to verify the previous findings. The test was performed using the following command:

```
LcKS(lm_best$residuals,"pnorm")
```

The resulting *"p-value"* was 2e-04 which confirmed that the residuals were gaussian.

The accuracy of the obtained model was determined by performing the following operations:

- Computing the Root Mean Square Error (RMSE) between predicted values and original test dataset values.
- Computing the minimum and maximum deviation of predicted values from the original values.

All metrics along with the code are shown in Figure 3.6:

```{r warning=FALSE}
RMSE(predlm_best,test$`Total GDP Loss`)
```

```
[1] 1.895851
```

```{r warning=FALSE}
min(abs(predlm_best-test$`Total GDP Loss`))
```

```
[1] 0.004966997
```

```{r warning=FALSE}
max(abs(predlm_best-test$`Total GDP Loss`))
```

```
[1] 9.198139
```

Figure 3.6: Model's accuracy measurement results.

The predictions obtained from the model were visualized by plotting the predicted values (red) over the test dataset values. The visualization indicated that the obtained linear regression model did not accurately fit the given data:

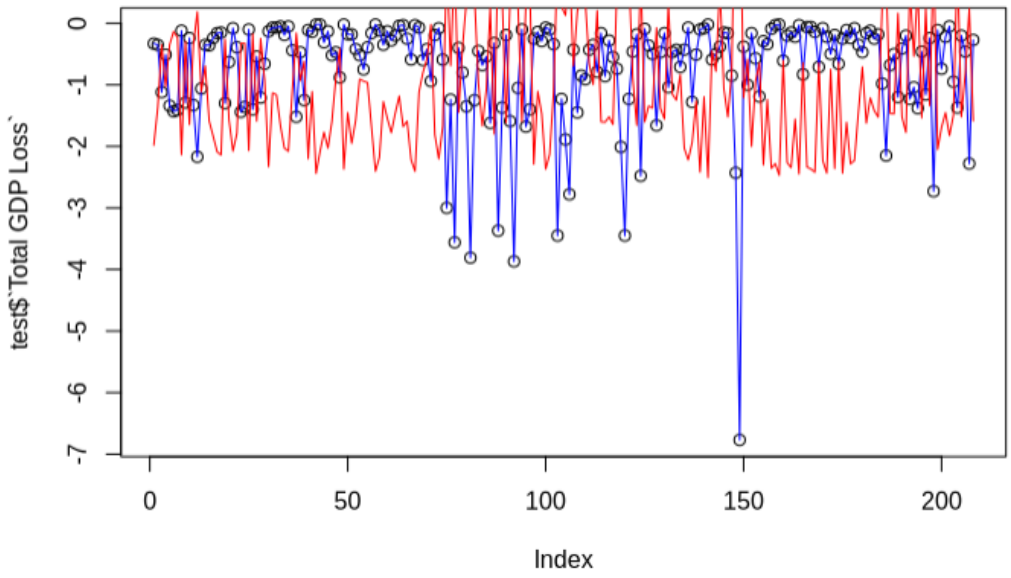

Figure 3.7: Predicted values versus original values.

Improved Model 1

The previous attempt showed that certain variables tend to affect the accuracy of the linear model. Therefore, another attempt was made after removing the row entries associated with *"Hotels and Restaurant"* and *"Transport services"* data values of the *"Sector"* column as their contribution to the previously obtained model was

insignificant. The command for generating a linear regression model remained the same. The following is the summary of the newly obtained model:

```
Call:
lm(formula = `Total GDP Loss` ~ `Sector GDP Loss` * Sector +
    Scenario, data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-2.4627 -0.3389  0.0037  0.4263  2.1806

Coefficients:
                                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                                    -2.029224   0.167534 -12.112  < 2e-16 ***
`Sector GDP Loss`                              -0.177075   0.036427  -4.861 1.74e-06 ***
SectorBusiness and Trade                        1.039668   0.159716   6.509 2.51e-10 ***
SectorLight/Heavy Manufacturing                 0.570549   0.163882   3.481 0.000559 ***
ScenarioAdditional impact (Short term)          0.004559   0.157959   0.029 0.976990
ScenarioLong term effects                      -0.390089   0.119868  -3.254 0.001243 **
ScenarioShort term effects                     -0.400104   0.130090  -3.076 0.002260 **
`Sector GDP Loss`:SectorBusiness and Trade     -0.271501   0.044979  -6.036 3.90e-09 ***
`Sector GDP Loss`:SectorLight/Heavy Manufacturing -0.225487   0.048148  -4.683 4.00e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6996 on 363 degrees of freedom
Multiple R-squared:  0.6988,    Adjusted R-squared:  0.6921
F-statistic: 105.3 on 8 and 363 DF,  p-value: < 2.2e-16
```

Figure 3.8: Summary of the improved linear regression model.

From the above model summary, it can be observed that *"Scenario Additional impact (Short term)"* is an insignificant variable. It is understandable as short term impact may not affect GDP much. The q-q plot of the model's residuals is shown below:

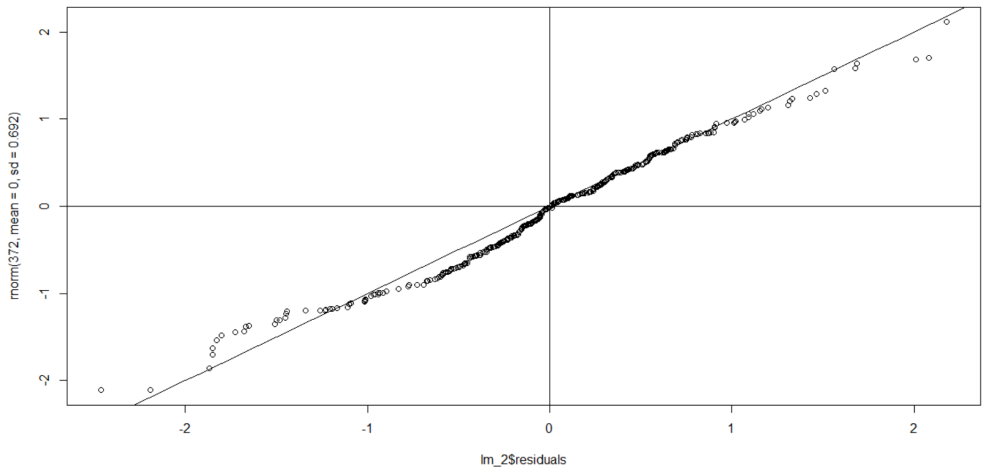

Figure 3.9: q-q plot of residuals.

All metrics associated with the improved model's accuracy along with the code are shown in Figure 3.10:

```r
RMSE(predlm_sgnf,test$`Total GDP Loss`)
```

```
[1] 0.7697049
```

```r
min(abs(predlm_sgnf-test$`Total GDP Loss`))
```

```
[1] 0.0002347793
```

```r
max(abs(predlm_sgnf-test$`Total GDP Loss`))
```

```
[1] 2.266151
```

Figure 3.10: Improved model's accuracy measurement results.

The predictions of the improved model are shown below:

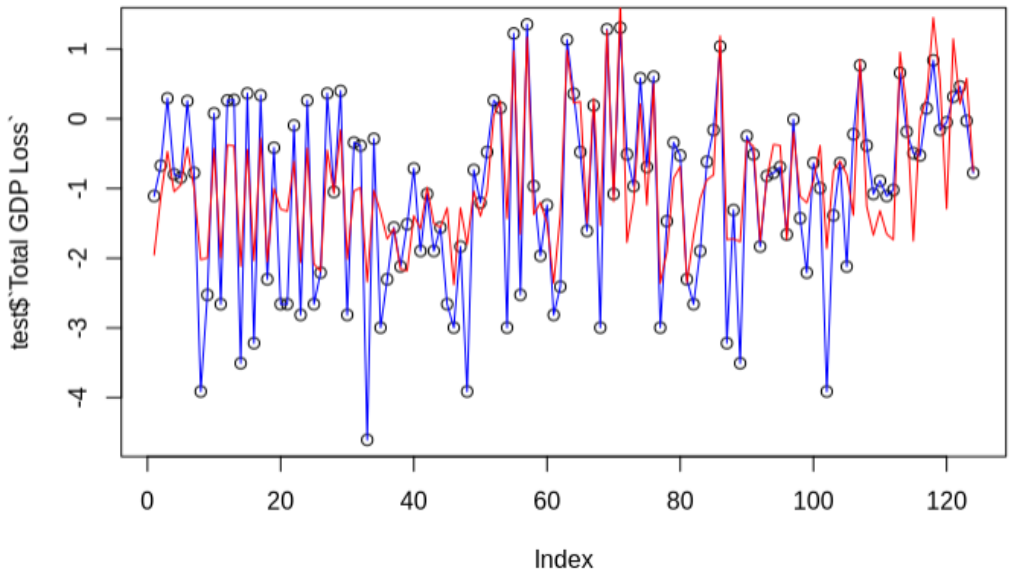

Figure 3.11: Improved model's predicted values versus original values.

Compared to the previous model, the improved model produced far better results. However, it was necessary to check if removing the last insignificant variable will produce better results or not.

Improved Model 2

The final linear regression model was trained after removing the only remaining insignificant variable from the dataset. Following is the summary of the final model:

```
Call:
lm(formula = `Total GDP Loss` ~ `Sector GDP Loss` * Sector +
    Scenario, data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-2.34449 -0.37810  0.02369  0.45598  2.26262

Coefficients:
                                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                                     -2.15116    0.16939 -12.699  < 2e-16 ***
`Sector GDP Loss`                               -0.22387    0.03753  -5.965 6.55e-09 ***
SectorBusiness and Trade                         1.28251    0.17729   7.234 3.57e-12 ***
SectorLight/Heavy Manufacturing                  0.78689    0.18648   4.220 3.20e-05 ***
ScenarioLong term effects                       -0.48264    0.11780  -4.097 5.32e-05 ***
ScenarioShort term effects                      -0.47219    0.12831  -3.680 0.000274 ***
`Sector GDP Loss`:SectorBusiness and Trade      -0.21389    0.04786  -4.469 1.09e-05 ***
`Sector GDP Loss`:SectorLight/Heavy Manufacturing -0.17390    0.05157  -3.372 0.000838 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.718 on 316 degrees of freedom
Multiple R-squared:  0.7134,    Adjusted R-squared:  0.7071
F-statistic: 112.4 on 7 and 316 DF,  p-value: < 2.2e-16
```

Figure 3.12: Summary of the final linear regression model.

The q-q plot of the final model's residuals is shown below:

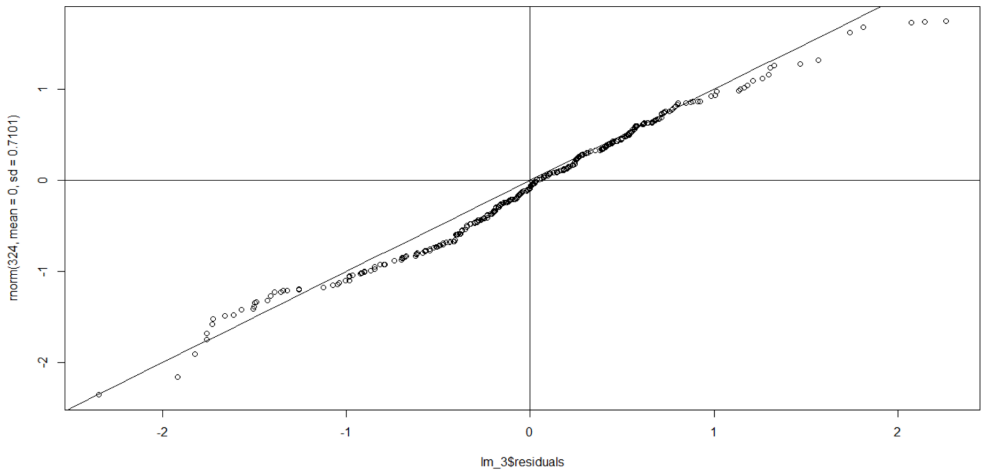

Figure 3.13: q-q plot of the final model's residuals.

The residuals follow a normal distribution indicating improvement in comparison to previous attempts. Now, it is necessary to check the residuals for homoscedasticity. In statistics, a sequence of random variables is homoscedastic if all its random variables have the same finite variance. Uneven variances may lead to biased and skewed results [10]. To check for homoscedasticity, the plot of the square of residuals was examined as shown below:

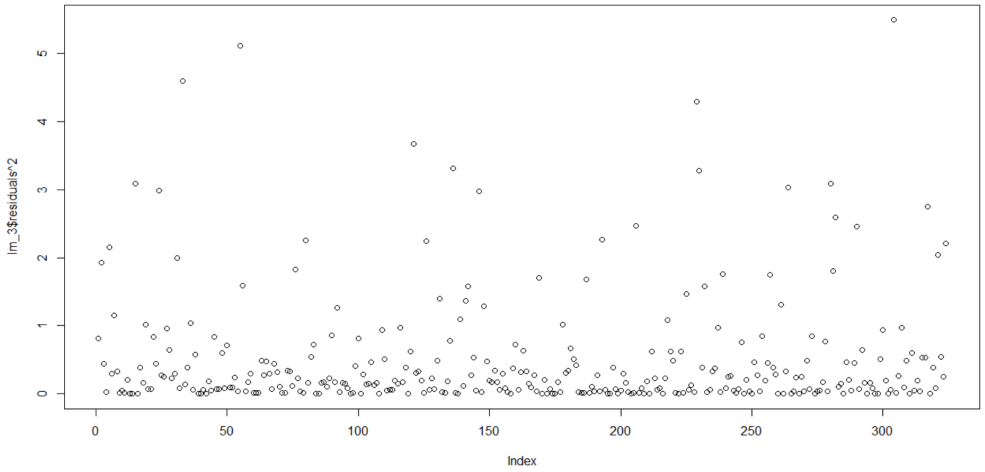

Figure 3.14: Squared residual plot of the final model.

The above plot was shown to mentor for examination and he concluded that the plot indicates heteroscedasticity. To remove it, an Artificial Neural Network (ANN) model was later implemented over the dataset associated with the final model. The final linear regression model's accuracy is shown below:

```{r warning=FALSE}
RMSE(predlm_sgnf_2,test$`Total GDP Loss`)
```

```
[1] 0.7906823
```

```{r warning=FALSE}
min(abs(predlm_sgnf_2-test$`Total GDP Loss`))
```

```
[1] 0.002623887
```

```{r warning=FALSE}
max(abs(predlm_sgnf_2-test$`Total GDP Loss`))
```

```
[1] 2.162498
```

Figure 3.15: Final model's accuracy measurement results.

While there was no improvement in the RMSE of the model, the maximum difference between the predicted and the actual values got reduced, thus improving fit. The predictions of the final model are shown below:

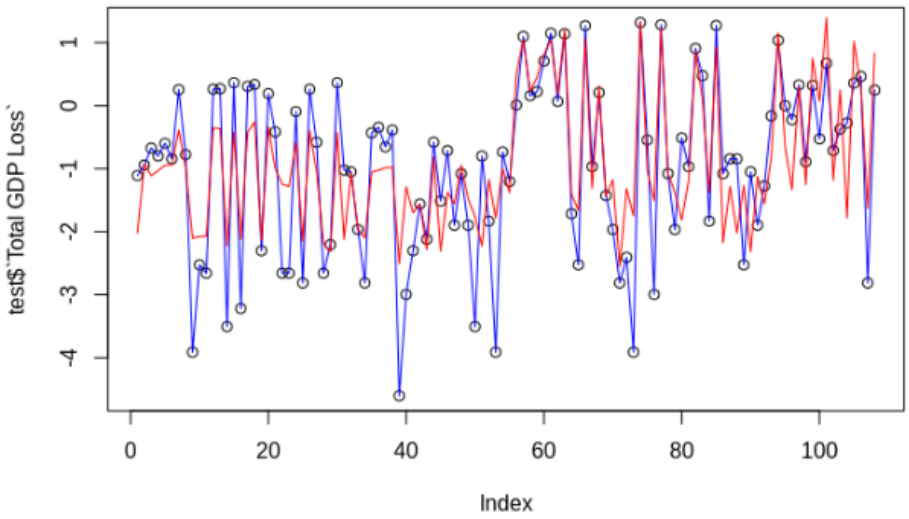

Figure 3.16: Final model's predicted values versus original values.

The final model is the best among all. However, RMSE can be further improved by making use of an ANN model as described in the following section.

3.5.2 Artificial Neural Network (ANN)

An artificial neural network takes an input vector of p variables $X = (X1, X2, \ldots, X_p)$ and builds a non-linear function $f(X)$ to predict the response Y [11]. The *nnet* [12] package of R was used to create the ANN model. The number of hidden units were set to six after numerous experiments. To ensure reproducibility of the results, the command to train an ANN was made to run 1000 times with the iteration number set as the random seed. Finally, the best ANN model with RMSE lower than that of the final linear regression model was selected and saved in the form of an *.rds* file for later use. The code to train and save an ANN model is shown below:

```
ANN_results <- NULL
for(i in 1:1000)
{
  set.seed(i)
  nnet_gdp=nnet(`Total GDP Loss`~`Sector GDP Loss`+Sector+Scenario,train,size=6,
      maxit=2000,linout=TRUE, abstol=0.000001,trace=FALSE)
  pred_nnet=predict(nnet_gdp,test)
  new_rm=RMSE(pred_nnet,test$`Total GDP Loss`)
  ANN_results <- rbind(ANN_results,c(i,new_rm))
}
ANN_selected=ANN_results[which(as.data.frame(ANN_results)$V2 == min(as.data.frame(
    ANN_results)$V2)),]
ANN_selected
set.seed(ANN_selected[1])
nnet_gdp=nnet(`Total GDP Loss`~`Sector GDP Loss`+Sector+Scenario,train,size=6,maxit
    =2000,linout=TRUE, abstol=0.000001,trace=FALSE)
pred_nnet=predict(nnet_gdp,test)
new_rm=RMSE(pred_nnet,test$`Total GDP Loss`)
saveRDS(nnet_gdp,"Best_NNET_GDP.rds")
best_rmse=new_rm
```

The best ANN model had the random seed set to 819 and its accuracy measurements are as follows:

```{r}
best_rmse
```

```
[1] 0.5474239
```

```{r}
min(abs(pred_nnet-test$`Total GDP Loss`))
```

```
[1] 0.002074419
```

```{r}
max(abs(pred_nnet-test$`Total GDP Loss`))
```

```
[1] 1.991076
```

Figure 3.17: ANN model's accuracy measurement results.

The ANN model gave better results as compared to the final linear regression model. The squared residual plot associated with the ANN model is shown below:

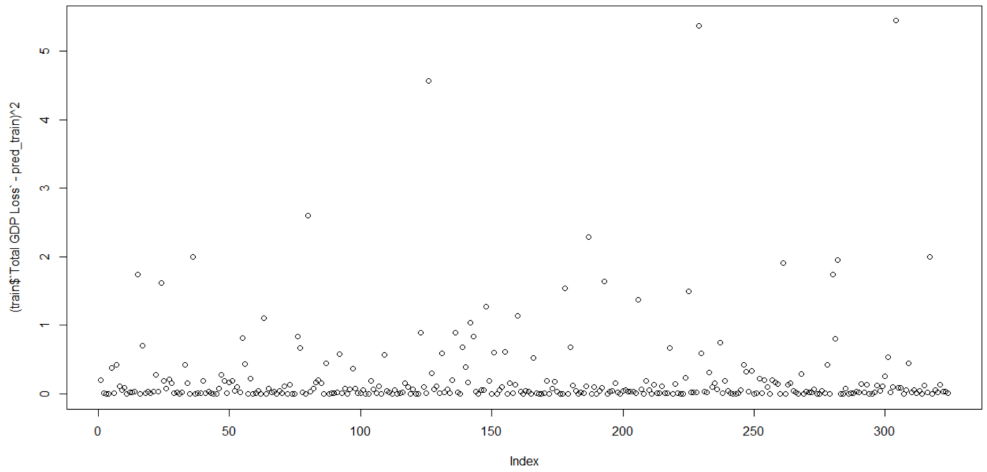

Figure 3.18: Squared residuals plot associated with the neural network model.

The above plot was reviewed by mentor and he concluded that the plot seems to indicate approximate homoscedasticity. The model fits the test data very well, as shown by the following figure:

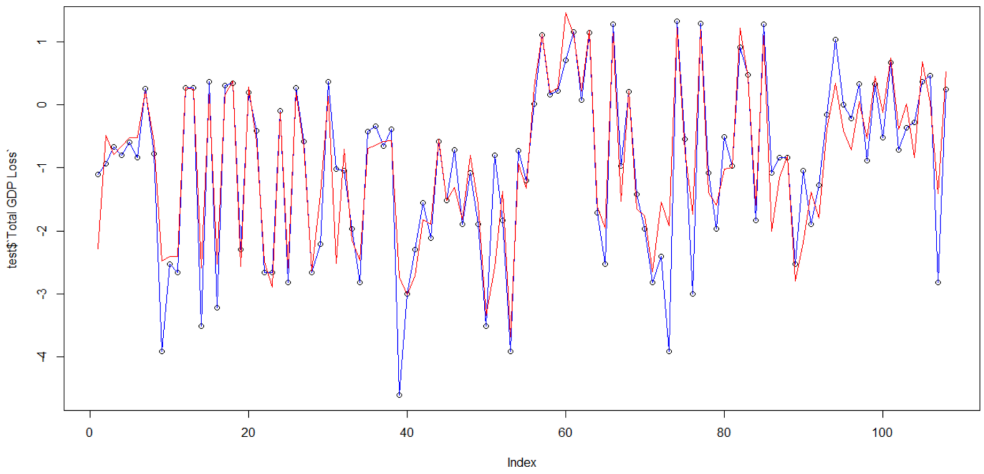

Figure 3.19: ANN model's predicted values versus original values.

Chapter 4

Conclusion

The case study attempted to explore the impact of COVID-19 on the GDP of various countries. The obtained statistical models have accurately predicted the GDP loss. Besides estimating the underlying pattern of the given data, the linear regression models also helped discover insignificant variables in the dataset. Later, an artificial neural network model was also trained for better accuracy and fit. As the final linear regression model predicted the actual values with a minor error. It seems like a good choice for predicting the GDP loss as it has the advantage of being a white-box approach. On the other hand, a neural network may provide better results but it is a black-box approach.

References

[1] T. Callen, "Gross domestic product: An economy's all." [Online]. Available: https://www.imf.org/external/pubs/ft/fandd/basics/gdp.htm

[2] H. Wickham, *The Power of a Single Number - A Political History of GDP*. Columbia University Press, 2016. [Online]. Available: https://cup.columbia.edu/book/the-power-of-a-single-number/9780231175104

[3] "Covid-19 economic impact assessment template." [Online]. Available: https://data.adb.org/dataset/covid-19-economic-impact-assessment-template

[4] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2021. [Online]. Available: https://www.R-project.org/

[5] H. Wickham and J. Bryan, *readxl: Read Excel Files*, 2019, r package version 1.3.1. [Online]. Available: https://CRAN.R-project.org/package=readxl

[6] E. Waring, M. Quinn, A. McNamara, E. Arino de la Rubia, H. Zhu, and S. Ellis, *skimr: Compact and Flexible Summaries of Data*, 2021, r package version 2.1.3. [Online]. Available: https://CRAN.R-project.org/package=skimr

[7] D. A. Freedman, *Statistical Models: Theory and Practice*. Cambridge University Press, 2009.

[8] R. Gnanadesikan and M. B. Wilk, "Probability plotting methods for the analysis of data," *Biometrika*, vol. 55, no. 1, pp. 1–17, 1968.

[9] H. W. Lilliefors, "On the kolmogorov-smirnov test for normality with mean and variance unknown," *Journal of the American Statistical Association*, vol. 62, no. 318, pp. 399–402, 1967. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/01621459.1967.10482916

[10] K. Yang, J. Tu, and T. Chen, "Homoscedasticity: an overlooked critical assumption for linear regression," *General Psychiatry*, vol. 32, no. 5, p. e100148, Oct. 2019. [Online]. Available: https://doi.org/10.1136/gpsych-2019-100148

[11] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning with Applications in R (Second Edition)*, 2021.

[12] W. N. Venables and B. D. Ripley, *Modern Applied Statistics with S*, 4th ed. New York: Springer, 2002, iSBN 0-387-95457-0. [Online]. Available: https://www.stats.ox.ac.uk/pub/MASS4/