

380_Project

Abdallah Al Rahbi

2023-04-23

Front Matter

```
packages <- c(
  "dplyr",
  "readr",
  "tidyr",
  "purrr",
  "broom",
  "magrittr",
  "corrplot",
  "caret",
  "rpart",
  "rpart.plot",
  "e1071",
  "torch",
  "luz",
  "ramify",
  "stringr",
  "janitor",
  "reshape2",
  "ggpubr",
  "ggplot2",
  "keras"
)

# renv::install(packages)
sapply(packages, require, character.only=T)
```

Introduction

Carbon Dioxide (CO₂) emissions are the leading cause of human induced climate change. According to the IPCC, humanity's greenhouse gas emissions have already raised the planet's temperature 1.1 degrees Celsius over pre-industrial levels, with CO₂ in the atmosphere being 50% higher than pre-industrial levels (420 PPM compared with 280 in pre industrial times). Because of this, various countries signed onto the Paris Climate Agreement which aimed to implement policies to curb CO₂ emissions to cap warming at 1.5 degrees Celsius over pre industrial levels. However, whether the necessary policies are being implemented is called into question with CO₂ emissions continuing to rise and the probability of the planet exceeding 1.5 degrees growing over time.

With this in mind, we aim to model CO₂ emissions in the United States using various relevant human activities as predictors. Through this we aim to identify which human activities are the strongest predictors of future CO₂ emissions. In particular, we employed two models: a Neural Network and Extreme Gradient Boosting.

Data Cleaning

```
owid <- read.csv("https://raw.githubusercontent.com/JackBenadon/Stat-380-group-project/main/owid.csv")
goma <- read.csv("https://raw.githubusercontent.com/JackBenadon/Stat-380-group-project/main/goma.csv")
eo <- read.csv("https://raw.githubusercontent.com/JackBenadon/Stat-380-group-project/main/eo.csv")
tran <- read.csv("https://raw.githubusercontent.com/JackBenadon/Stat-380-group-project/main/tran.csv")

zowidyearly1800_2021 <- owid%>%
  select(c(country,year,co2,gdp,population))%>%
  filter(country=="United States")
zowideyearly2005_2022 <- zowidyearly1800_2021%>%
  filter(year>=2005)
goma$DATE <- as.Date(goma$DATE)
goma$year <- as.numeric(format(goma$DATE, "%Y"))
zGOMAYearly2005_2022 <- goma%>%
  filter(str_detect(DATE, "01-01"))
zeosmonthly1973_2022 <- eo[10:610,]
zeosmonthly1973_2022 <- zeosmonthly1973_2022%>%
  row_to_names(row_number = 1)
zeosmonthly1973_2022 <- zeosmonthly1973_2022[-c(1),]
zeosyearly1973_2022 <- zeosmonthly1973_2022%>%
  filter(str_detect(Month, "January"))
zeosyearly2005_2022 <- zeosyearly1973_2022
zeosyearly2005_2022$Month <- gsub("January","",zeosyearly2005_2022$Month)
```

```

zeosyearly2005_2022 <- zeosyearly2005_2022%>%
  filter(Month>=2005)
ztransyearly1990_2015 <- tran %>%
  select(-c(X,X.1,X.2,X.3,X.4,X.5)) %>%
  t() %>%
  as.data.frame() %>%
  row_to_names(row_number = 1) %>%
  clean_names()%>%
  select(-c(2,23))

```

Warning: Row 1 does not provide unique names. Consider running clean_names() after row_to_names().

Explotory Data Analysis

OWID

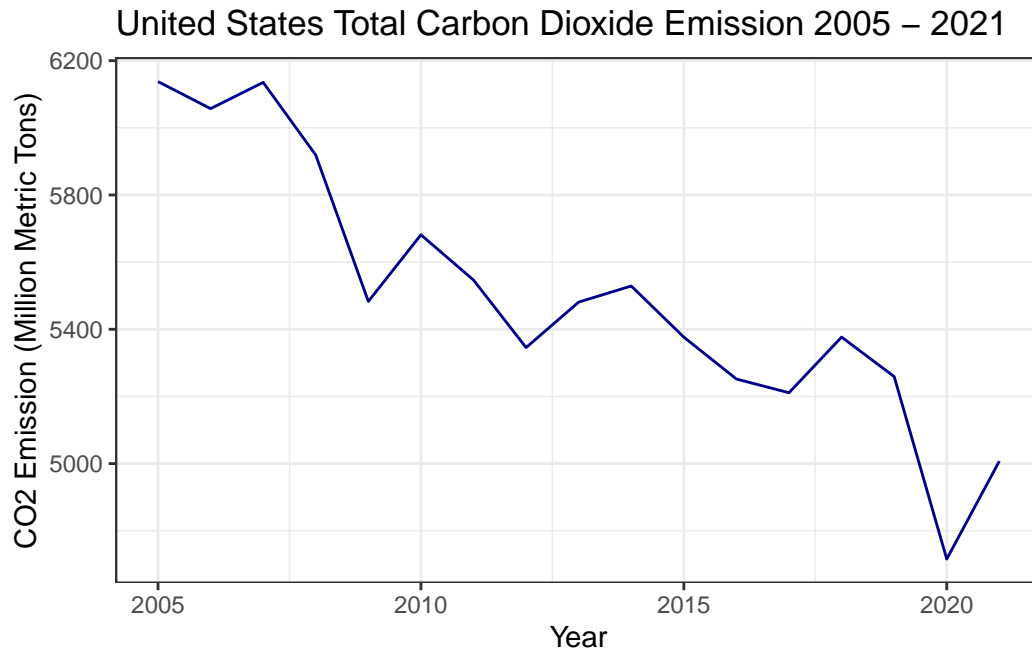
We produced a line graph for the total yearly carbon dioxide emission in the United States from 2005 to 2021.

From the graph, we found that the Carbon Dioxide emission is in steadily declining since 2005, which is a direct result of efforts made on reducing co2 emission since the recognition of global warming. There is a slight increase in emission in 2021. This aligns with the “Build America Buy America Act”, a legislation made in the same year, stating that all construction material used for infrastructure must be manufactured in the united states. Also, after COVID-19, global trade was greatly hindered, and the need for reliable goods produced domestically increased.

```

zzlowid2005_2021<- ggplot(zowideyearly2005_2022,aes(x=year,y=co2))+geom_line(color="darkbl
zzlowid2005_2021

```



GOMA

We produced a line graph for the Gross Manufacturing Output in the United States from 2005 to 2021.

From the graph, we found that the gross manufacturing output have increased roughly 55% since 2005, with slight dips in 2008, 2014, and 2020. This aligns with the global financial crisis, the energy crisis, and the COVID-19 pandemic respectively. It is speculated that this is due to the Manufacturing Enterprise Integration Act of 2002, which promoted the rise of smart manufacturing, and later digital manufacturing and robotic use. Data suggests that this act eliminated jobs, but increased the production capacity by a large margin.

```
zzlgoma2005_2022 <- ggplot(zGOMAYearly2005_2022,aes(x=year,y=GOMA))+geom_line(color="red")
zzlgoma2005_2022
```

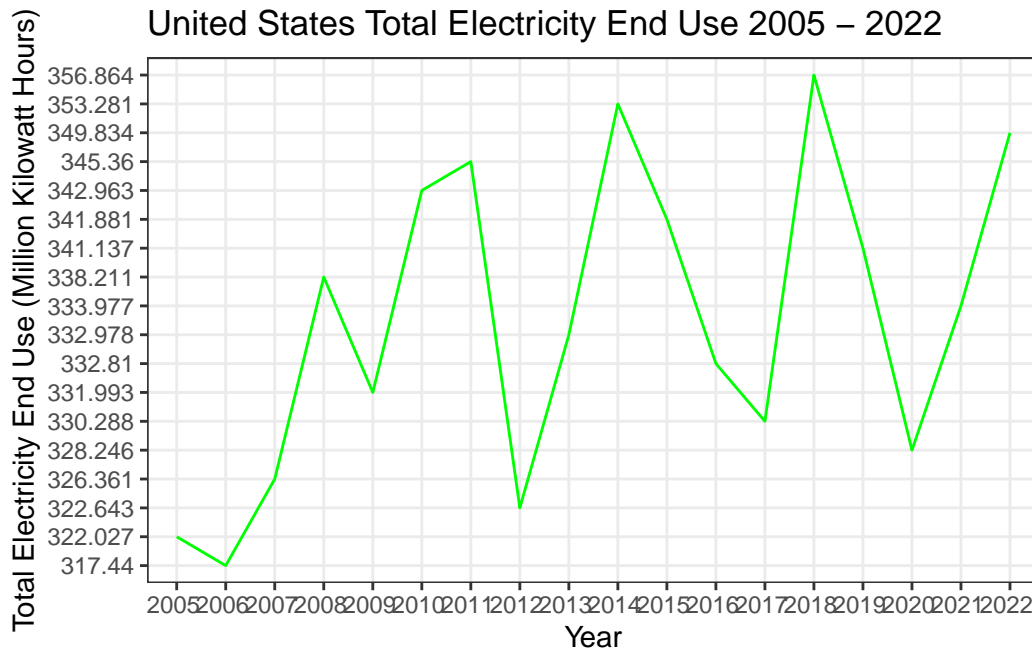


EO

We produced a line graph for the Gross Manufacturing Output in the United States from 2005 to 2021.

From the graph, we can see that there is no obvious trend observed for this variable, as the measure fluctuates between 317 million and 350 million kilowatt hours over the observed period. CO2 emission from the electricity sector is still extremely relevant, since that despite tremendous amount of investments made in clean energy, 60% of electricity generated in the US are still from traditional sources like natural gas, coal and petroleum.

```
colnames(zeosyearly2005_2022)[12] <- "ElectricityEndUseTotal"
zzleosyearly2005_2022eu <- ggplot(zeosyearly2005_2022, aes(x=Month,y = ElectricityEndUseTo
zzleosyearly2005_2022eu
```



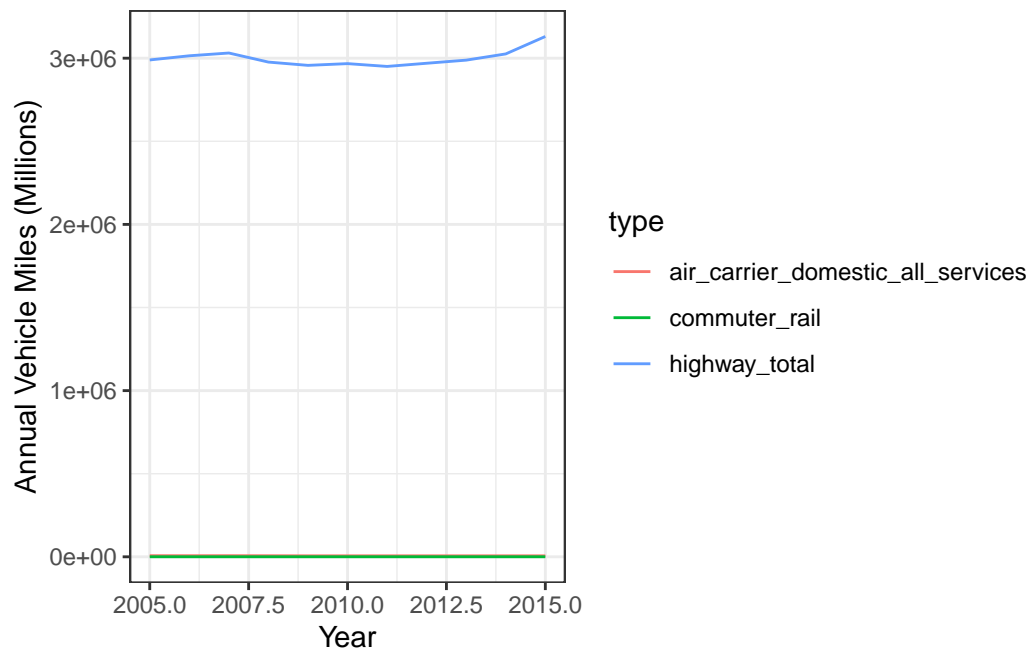
Transportation

We produced a line graph for the Gross Manufacturing Output in the United States from 2005 to 2021.

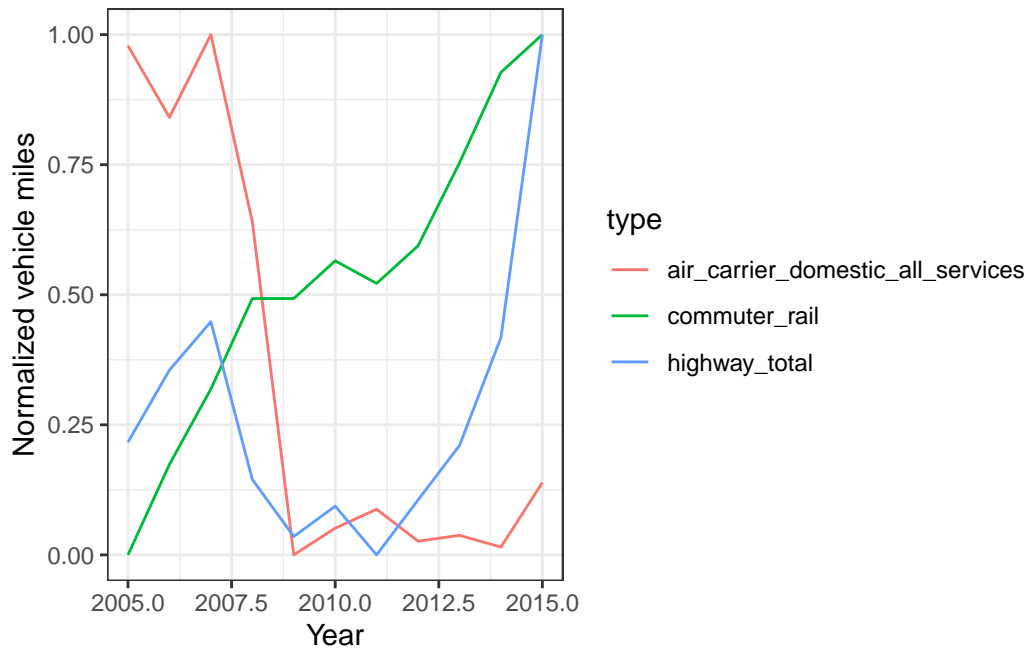
Finally we looked at the vehicle usage which is measure in million miles. We normalized the data for uniformity in comparison, as the magnitude of data differ significantly between the different sectors. We found that over the ten year observation period, air travel millage dropped 16%. In the mean time, travel by rail increased 25%, and total highway millage increased 5%. We speculate that this is not related to 911, instead, it is due to the poor customer service provided by airlines, and increased airline fares, as air travel millage peaked in 2007, at a 20% increase from 2001.

```
colnames(ztransyearly1990_2015)[1] <- "Year"
z2transyearly1990_2015 <- ztransyearly1990_2015 %>%
  pivot_longer(!Year, names_to = "type", values_to = "value")%>%
  filter(type %in% c("air_carrier_domestic_all_services", "highway_total", "commuter_rail"))
  mutate(Year = as.numeric(Year), value)
zztransyearly2005_2015 <- z2transyearly1990_2015%>%
  filter(Year>=2005)%>%
  mutate(value = gsub(",", "", value))%>%
  mutate(value=gsub("\\(R\\) ", "", value))%>%
```

```
mutate(value=as.numeric(value))
ggplot(zztransyearly2005_2015, aes(x=Year, y=value, color=type,group=type))+geom_line()+th
```



```
zztransyearly2005_2015 <- zztransyearly2005_2015%>%
  group_by(type)%>%
  mutate(value=((value - min(value))/(max(value)-min(value))))
ggplot(zztransyearly2005_2015, aes(x=Year,y=value,color=type,group=type))+geom_line()+them
```



```

zztranshighway <- zztransyearly2005_2015 %>%
  filter(type=="Highway, total")
zztransair <- zztransyearly2005_2015 %>%
  filter(type=="Air carrier, domestic, all services")
zztransrail <- zztransyearly2005_2015 %>%
  filter(type=="Commuter rail")

```

```

#loading the data
data = read.csv("https://raw.githubusercontent.com/JackBenadon/Stat-380-group-project/main")
#removing NA values
data <- data %>%
  na.omit()
#removing the (R) in the commuter.rail column
data$Commuter.rail[11] <- 372
#removing the commas in Highway..total because its special
data$Highway..total <- gsub(",", "", data$Highway..total)
# making the all the columns into numeric variables
data$Commuter.rail <- as.numeric(sub(",", "", data$Commuter.rail))
data$Air.carrier..domestic..all.services <- as.numeric(sub(",", "", data$Air.carrier..domestic..all.services))

```



```

data$Highway..total <- as.numeric(data$Highway..total)
#making sure the transport data is only one column
data$transport <- data$Air.carrier..domestic..all.services + data$Commuter.rail + data$Hig
#duplicating data object so both models function properly
data2 <- data

```

Modelling part 1:

Finally, we took two approaches to modelling. Firstly, we constructed a two layer neural network, consisting of 64 nodes in first layer and 32 nodes in the second layer, with MSE loss and ReLU activations. Ultimately, our goal was to use this model to identify which variables were most important for our analysis. We first built the full model with all four predictors to see how it performed. However, we then repeatedly trained the model excluding one predictor at a time then saw how each models loss was affected. Our idea was that the model in which loss was most impacted would indicate that the given excluded variable was most important to predicting CO2

```

#Cleaning for model 1

#selecting variables
data = data %>%
  select(year,co2,transport, ElectricityEndUseTotal, GOMA, gdp)
#Creating predictor and response variables
xdata <- data %>%
  select(transport, ElectricityEndUseTotal, GOMA, gdp)
corrdata <- data %>%
  select(co2,transport, ElectricityEndUseTotal, GOMA, gdp)

ydata <- data %>%
  select(co2)
#Normalizing variables
mean <- apply(data, 2, mean)
std <- apply(data, 2, sd)
std<-case_when(std==0 ~ 1,
               std !=0 ~ std)

meanx <- apply(xdata, 2, mean)
stdx <- apply(xdata, 2, sd)
stdx<-case_when(stdx==0 ~ 1,
                stdx !=0 ~ stdx)

```

```

meany <- apply(ydata, 2, mean)
stdy <- apply(ydata, 2, sd)
stdy<-case_when(stdy==0 ~ 1,
                stdy !=0 ~ stdy)

data <- scale(data, center = mean, scale = std)
xdata <- scale(xdata,center = meanx, scale = stdx)
ydata = scale(ydata,center = meany, scale = stdy)

set.seed(380)
#Full Model 4 inputs
model = keras_model_sequential() %>%
  layer_dense(units=64, activation="relu", input_shape=4) %>%
  layer_dense(units=32, activation = "relu") %>%
  layer_dense(units=1, activation="linear")

#Loss Testing 3 inputs:
modelt = keras_model_sequential() %>%
  layer_dense(units=64, activation="relu", input_shape=3) %>%
  layer_dense(units=32, activation = "relu") %>%
  layer_dense(units=1, activation="linear")

model %>% compile(
  loss = "mse",
  optimizer = "adam",
  metrics = list("mean_absolute_error")
)

modelt %>% compile(
  loss = "mse",
  optimizer = "adam",
  metrics = list("mean_absolute_error")
)

```

```
model %>% summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
dense_2 (Dense)	(None, 64)	320
dense_1 (Dense)	(None, 32)	2080
dense (Dense)	(None, 1)	33

Total params: 2,433

Trainable params: 2,433

Non-trainable params: 0

```
x = as.matrix(data[1:11,3:6])

xNoElec = as.matrix(data[1:11,3:6][,-2])

xNoMan = as.matrix(data[1:11,3:6][,-3])

xNoTrans = as.matrix(data[1:11,3:6][,-1])

xNoGDP = as.matrix(data[1:11,3:6][,-4])

y = as.matrix(data[1:11,2])

#Full Model loss
model %>% fit(x, y, epochs = 100, verbose = 1)

#Testing models
modelt %>% fit(xNoElec, y, epochs = 100, verbose = 1)
#0.0407

modelt %>% fit(xNoMan, y, epochs = 100, verbose = 1)
#0.0196

modelt %>% fit(xNoTrans, y, epochs = 100, verbose = 1)
#0.0317
```

```

modelt %>% fit(xNoGDP, y, epochs = 100, verbose = 1)
#0.017

#historyFull <- model %>% fit(x, y, epochs = 100, verbose = 1)
#historyT <- modelt %>% fit(xNoMan, y, epochs = 100, verbose = 1)

```

Looking at the loss performance for each of our models, we see that the model that exclude electricity has the highest loss value, followed by the model without transportation. That electricity followed by transportation are the most important predictors of Carbon Dioxide emissions.

Modeling Part 2

The second model we used was an Extreme Gradient Boost model, also known as XGBoost. This model works by first generating a layer of decision tree models and averaging them out to minimize the variance associated with the decision tree model type. After that, it attempts to make a new layer of decision trees that tries to account for as much of the previous layer's error as possible. It continues to do this process, using a loss function and gradient descent. We chose this model because it fit our particular problem pretty well: it can account for time series data relatively easily, it is one of the models that is very popular in real data science currently, it wasn't too hard to work with, and it has built-in feature importance generation, allowing us to easily compare which predictors had the greatest effect on the model.

```
library(xgboost)
```

Warning: package 'xgboost' was built under R version 4.2.3

Attaching package: 'xgboost'

The following object is masked from 'package:dplyr':

```
slice
```

```

data2 <- as.data.frame(data2)

train <- data2[1:9,]
test <- data2[10:11,]

```

```

train_Dmatrix <- train %>%
  dplyr::select(transport,gdp,GOMA,ElectricityEndUseTotal) %>%
  as.matrix()

pred_Dmatrix <- test %>%
  dplyr::select(transport,gdp,GOMA,ElectricityEndUseTotal) %>%
  as.matrix()

targets <- train$co2

#Cross-validation
library(caret)

xgb_trcontrol <- trainControl(
  method = "cv",
  number = 5,
  allowParallel = TRUE,
  verboseIter = FALSE,
  returnData = FALSE
)

#Building parameters set
xgb_grid <- base::expand.grid(
  list(
    nrounds = c(5,10,50, 100,200,500,1000),
    max_depth = c(1:6),
    colsample_bytree = 1,
    eta = 0.5,
    gamma = 0,
    min_child_weight = 1,
    subsample = 1)
)

#Building the model
model_xgb <- caret::train(
  train_Dmatrix,targets,
  trControl = xgb_trcontrol,
  tuneGrid = xgb_grid,
  method = "xgbTree",
  nthread = 10,

```

```

    verbosity=0
  )

```

Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
There were missing values in resampled performance measures.

```

model_xgb$bestTune

```

```

nrounds max_depth eta gamma colsample_bytree min_child_weight subsample
6      500      1 0.5      0              1              1              1

```

```

#Making the variables used in forecast object
fitted <- model_xgb %>%
  stats::predict(train_Dmatrix) %>%
  stats::ts(start = c(2005,1),frequency = 1)

ts_co2 <- ts(targets,start=c(2005,1),frequency=1)
forecast_xgb <- model_xgb %>% stats::predict(pred_Dmatrix)
forecast_ts <- ts(forecast_xgb,start=c(2014,1),frequency=1)

#Preparing forecast object
forecast_co2 <- list(
  model = model_xgb$modelInfo,
  method = model_xgb$method,
  mean = forecast_ts,
  x = ts_co2,
  fitted = fitted,
  residuals = as.numeric(ts_co2) - as.numeric(fitted)
)
class(forecast_co2) <- "forecast"

#The function to convert decimal time label to wanted format
library(lubridate)

```

Warning: package 'lubridate' was built under R version 4.2.3

Attaching package: 'lubridate'

The following objects are masked from 'package:base':

date, intersect, setdiff, union

```
date_transform <- function(x) {format(date_decimal(x), "%Y")}  
#Making a time series variable for observed data  
observed_values <- ts(test$co2,start=c(2014,1),frequency=1)  
  
#Plot forecasting  
library(ggplot2)  
library(forecast)
```

Warning: package 'forecast' was built under R version 4.2.3

Registered S3 method overwritten by 'quantmod':

```
method      from  
as.zoo.data.frame zoo
```

Attaching package: 'forecast'

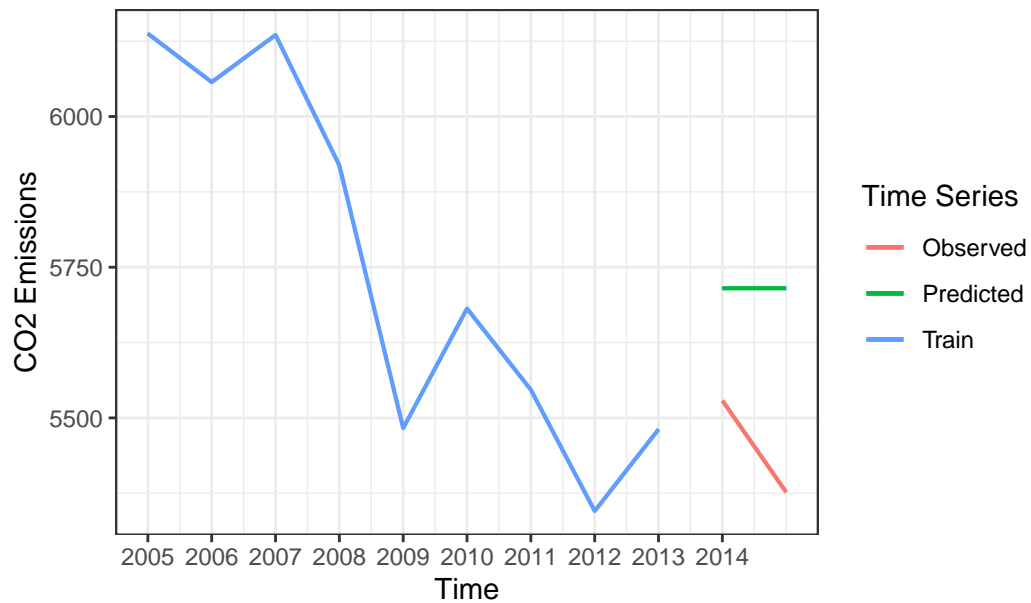
The following object is masked from 'package:ggpubr':

gghistogram

```
autoplot(forecast_co2)+  
  autolayer(forecast_co2$mean,series="Predicted",size=0.75) +  
  autolayer(forecast_co2$x,series = "Train",size=0.75 ) +  
  autolayer(observed_values,series = "Observed",size=0.75) +  
  scale_x_continuous(labels =date_transform,breaks = seq(2005,2014,1) ) +  
  guides(colour=guide_legend(title = "Time Series")) +  
  ylab("CO2 Emissions") + xlab("Time") +  
  ggtitle("") +  
  theme_bw()
```

Scale for x is already present.

Adding another scale for x, which will replace the existing scale.

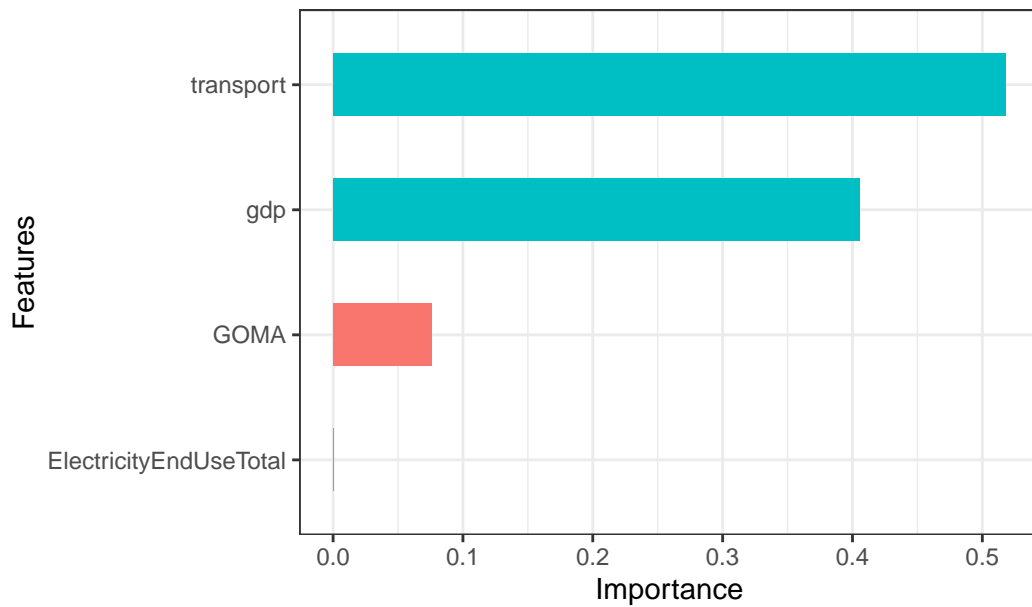


```
#Feature importance
library(Ckmeans.1d.dp)
```

Warning: package 'Ckmeans.1d.dp' was built under R version 4.2.3

```
xgb_imp <- xgb.importance(
  feature_names = colnames(train_Dmatrix),
  model = model_xgb$finalModel)

xgb.ggplot.importance(xgb_imp, n_clusters = c(2)) +
  ggtitle("") +
  theme_bw() +
  theme(legend.position="none")
```

```
xgb_imp$Importance
```

```
[1] 0.5180465303 0.4056273088 0.0756505753 0.0006755855
```

XGBoost Results

Unfortunately, our second model does a poor job predicting our data, as seen in the time series line graph above. We believe this is due to the model being trained on such a small dataset, as this type of machine learning is best suited for huge datasets with many more data points than we had here. Regardless, it indicates that transportation is our most important predictor, followed by GDP.