Predicting the Australian unemployment rate using machine learning and neural network

Brendan Garner

Abstract

The purpose of this report is to apply a machine learning algorithm and neural network in R using TensorFlow and Keras to predict the unemployment rate in Australia.

Using data from the Australian Bureau of Statistics, both models were trained on data between June 1981 and December 2017 (inclusive) and tested on data between March 2018 – September 2020 (inclusive). The artificial neural network took substantially longer to train but provided more accurate predictions than the random forest model.

It was found that both models did a reasonable job of predicting the unemployment rate for the test data set except for the last two quarters which reflected the economic shock brought on by government restrictions in response to the Covid-19 global pandemic. The models were trained using data from normal business cycles could not be used to predict unemployment during a once in a hundred-year event like Covid-19.

Predictive performance could be improved by including additional variables such as mining investment, construction activity and health care spending. A better model would also consider the data as a time series.

Brief overview of the Australian unemployment rate (1999 – 2020)

Turn of the century

Excessive speculation in internet related companies during the late 1990's and into 2000 caused a stock market bubble in the Nasdaq. The bursting of this bubble and the events of 9/11 caused a shallow recession in the United States between March and November 2001 (Nordhaus, 2002).

In Australia, the Federal Government announced that a Goods and Services Tax (GST) would be introduced on 1 July 2000. This resulted in a substantial bring-forward of housing investment that almost caused a technical recession in the second half of that year. Unemployment rose during this period and continued into 2001, peaking around the same time as the United States emerged from recession.

The mining boom

The rapid urbanisation and industrialisation of China and other emerging economies in Asia drove the global prices for Australia's resource exports higher by more than 300 per cent ($USD) between 2003 and 2011 (Connolly and Orsmond, 2011). The booming mining and construction sectors contributed to roughly 20 per cent of total employment growth up to 2008. Unemployment fell during this period by 2.8 percentage points despite strong immigration and a substantial increase in the participation rate (Borland, 2011).

The Global Financial Crisis (GFC)

The domestic economy was slowing in the first half of 2008 prior to the bankruptcy of Lehman Brothers in September, which resulted in an extreme financial shock and a collapse in commodity prices. The RBA responded by slashing interest rates by a total of 375 basis points within five months. Thanks to the rapid policy response, a lower Australian dollar and strong demand for bulk commodities from China, the downturn in Australia was milder than had been feared (Kearns and Lowe, 2011).

Mining moves from construction to production

Mining investment peaked around 2012, with the transition from a labour-intensive construction phase to a capital-intensive production phase leading to a weaking of labour demand across different industries. Much of the slowing in economic growth and the rise in unemployment can be attributed to the reduction in mining investment (Kent, 2014). During this period, net immigration declined with many workers returning to New Zealand to assist with the reconstruction activity in Christchurch following the earthquake (Kent, 2015).

East coast construction boom

A sustained period of low interest rates resulted in a substantial increase in residential and infrastructure construction on the east coast of Australia. By 2018, construction employment accounted for almost 10 percent of total employment, its highest share since the 1920s. This period also coincided with the introduction of the National Disability Insurance Scheme, leading to a growing number of health-related jobs. By 2018, the unemployment rate had fallen by 1 percent from the peak in October 2014 (Debelle, 2018).

Covid-19

To reduce the spread of Covid-19 in the community, the Australian government introduced a number of restrictions that resulted in a decline of almost 20 percent in hours worked by early May 2020. Unemployment rose to 7.4 percent by June 2020, a 21-year high (Lowe, 2020). By the end of the year, the unemployment rate had fallen to 6.6% (ABS, 2020).

Data preparation

Data wrangling

As a result of the quarterly Job Vacancies Survey (JVS) being suspended, no original estimates were produced for five quarters between August 2008 and August 2009 inclusive. The ABS subsequently modelled the missing data using the Hours Worked Estimates (ABS, 2010) and included these figures in the Treasury Macroeconomic Model (TRYM) model. Missing JVS data were obtained from the TRYM model (ABS, 2009) and these values were used to replace the missing JVS values in the data set.

Missing values for Estimated Resident Population (Sept 2019 – June 2020) were obtained from the latest National, state and territory population release from the ABS (ABS, 2020). One missing value remained (Sept 2020) which had not been released by the ABS. A value was imputed using the population figure from the previous quarter, with an adjustment equal to the difference between the June and March figures. This small upward adjustment seems reasonable considering the restrictions on international arrivals following the Covid-19 pandemic. The existing population data in the dataset (June 1981 – June 2019) contained values ten times larger than their correct values. These values were adjusted to their correct values.

Descriptive statistics

The data consisted of 158 quarterly observations from June 1981 to September 2020 with one target variable (unemployment_rate) and seven predictor variables. Box plots in R were used to visualise the data distribution of predictor variables that were reported as percentages. Figure 1 highlights that GDP and consumption generally has a tight range with outliers corresponding to the peaks and troughs in the business cycle. The extreme negative outliers represent the extreme financial impact of Covid-19 on the Australia economy in 2020. Follow this visualisation, the data was split into training (June 1981 – December 2017, 147 observations) and testing (March 2018 – September

2020, 11 observations). The data was subsequently scaled by their standard deviations for use in the artificial neural network. The random forest model did not require data scaling.

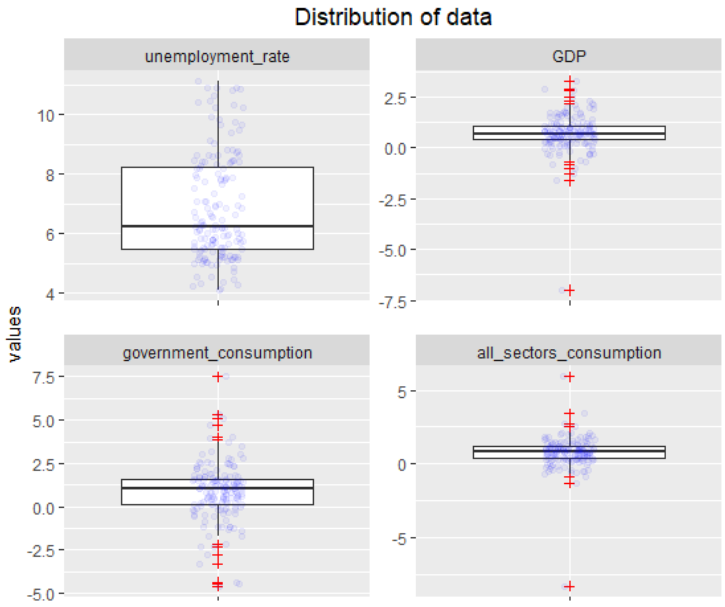

Figure 1. Visualisation of variables reported as percentages.

Machine Learning

The random forest model was chosen to predict the Australian unemployment rate from March 2018 to September 2020. The random forest model can improve performance by building a large collection of de-correlated trees.

The performance of the algorithm can be fine-tuned by adjusting three hyperparameters (min_node_size, mtry and trees). The minimum number of observations to allow the algorithm to continue splitting nodes is known as min_node_size. The number of randomly selected features evaluated when splitting nodes is known as mtry and has an approximate value equal to the number of variables divided by three ($7/3 = 2.3$). Trees refers to the number of de-correlated decision trees that form the random forest. The default setting for repeated cross-validation in the algorithm was left at 10 folds, repeated 10 times. As the data set was small, computation time was not a concern.

The random forest model was fit to the training data. Hyperparameters were adjusted with the effect of these adjustments to the mean squared error (MSE) shown in Figure 2. The MSE measures the error between the prediction and the actual values. The MSE appeared to stabilise at around 950 trees. With this hyperparameter value fixed, the remaining two hyperparameters were adjusted using the TuneControlGrid from the package *mlr*. Using this grid, values between 4 - 6 were tested for mtry and between 2 – 6 for min_node_size. The final values for mtry and min_node_size were set to five and two, respectively. The final MSE for the training data was recorded as **0.1527511**.

Performance of hyperparameter adjustment on the training data set

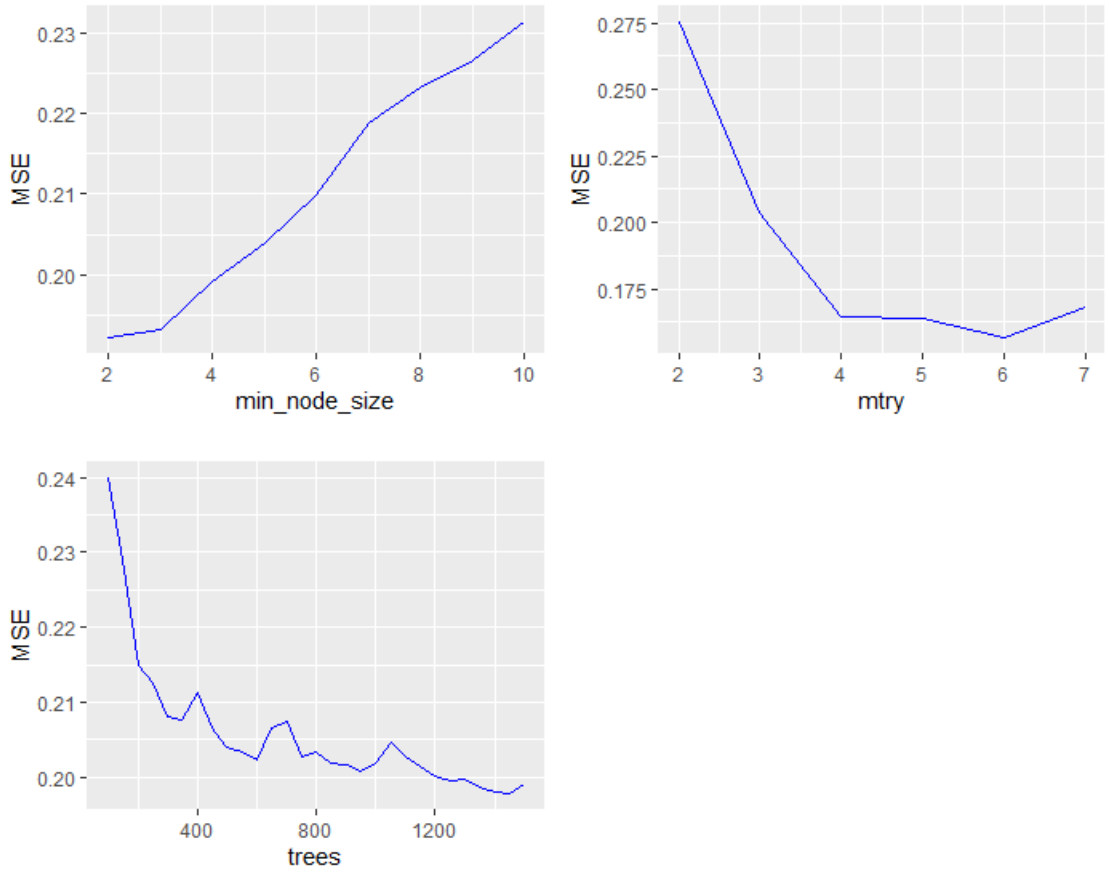

Figure 2. The effect on mean squared error (MSE) of the training data by adjusting each hyperparameter value

Predictive performance of the random forest model on test data

The fitted model was then used to predict the unemployment rate for the test dataset (March 2018 – September 2020), achieving a lower performance with a MSE of **0.1784898** (higher MSE) than the result on the training data set. The predictions and the actual unemployment rate are shown in Table 1.

	Mar-18	Jun-18	Sep-18	Dec-18	Mar-19	Jun-19	Sep-19	Dec-19	Mar-20	Jun-20	Sep-20
Prediction	5.58	5.64	5.56	5.56	5.57	5.55	5.56	5.72	5.56	5.73	5.58
Actual	5.53	5.40	5.20	5.03	5.03	5.20	5.20	5.17	5.20	6.97	7.07
Difference	0.05	0.24	0.36	0.53	0.54	0.35	0.36	0.55	0.36	-1.23	-1.48

Table 1. Performance of the random forest model on the test data set.

Artificial Neural Network

Network structure

The unemployment_rate variable was separated from the train and test data sets, leaving the seven predictor variables (scaled data). These predictor variables form the 7-node input layer for the vanilla artificial neural network. The model features a total of **6 hidden layers** with **7 nodes** per layer, activated by ReLU function. The target variable (unemployment_rate) formed the output layer.

Predictive performance of the model on the training data set

The model achieved a mean absolute error (MAE) of 0.2678184.

Predictive performance of the model on the test data set

The artificial neural network performed quite well on most of the test data except for the final two quarterly where model performance was very poor. The model achieved a MAE of 0.2626425.

	Mar-18	Jun-18	Sep-18	Dec-18	Mar-19	Jun-19	Sep-19	Dec-19	Mar-20	Jun-20	Sep-20
Prediction	5.00	5.18	5.14	5.09	4.99	5.10	5.14	5.34	5.25	5.47	4.94
Actual	5.53	5.40	5.20	5.03	5.03	5.20	5.20	5.17	5.20	6.97	7.07
Difference	-0.54	-0.22	-0.06	0.06	-0.04	-0.10	-0.06	0.17	0.05	-1.49	-2.12

Table 2. Performance of the artificial neural network on the test data set – with six hidden layers each with 7 neurons.

Varying number of neurons

The model featuring three layers (input, output and one hidden layer) was fit to the training data ten times with 7 neurons (number of neurons in input layer) and ten times with 14 neurons. The MAE values of each epoch were averaged and plotted in Figure 3 which shows the network with 7 neurons exhibiting a slightly slowly MAE.

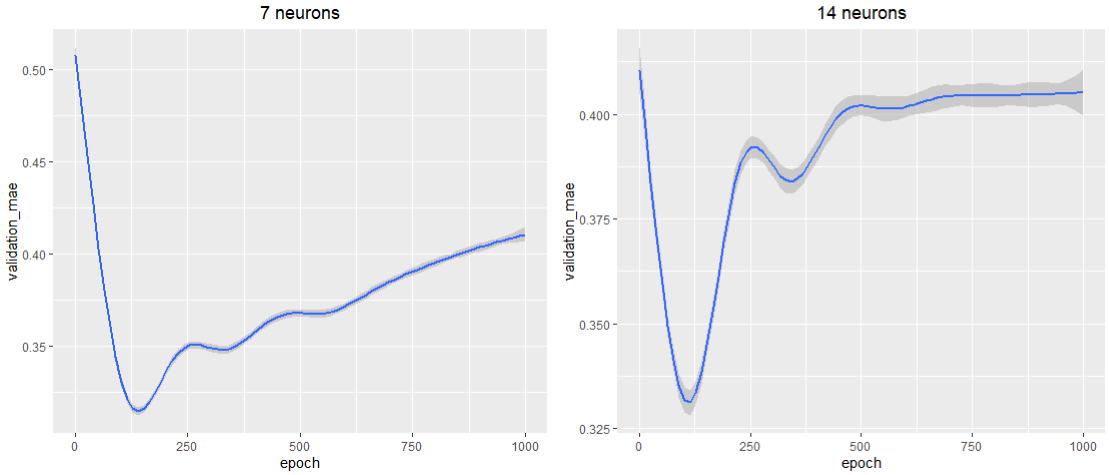

Figure 3. ANN performs slightly better with 7 neurons.

Varying number of layers

The model was set to 7 neurons and fit to the training data three times with five hidden layers for 1000 epochs. The model was subsequently fit three times with three hidden layers for 300 epochs. The MAE values were averaged and plotted in Figure 4. Another model was fit with eight hidden layers for 250 epochs with a considerably higher error rate (MAE).

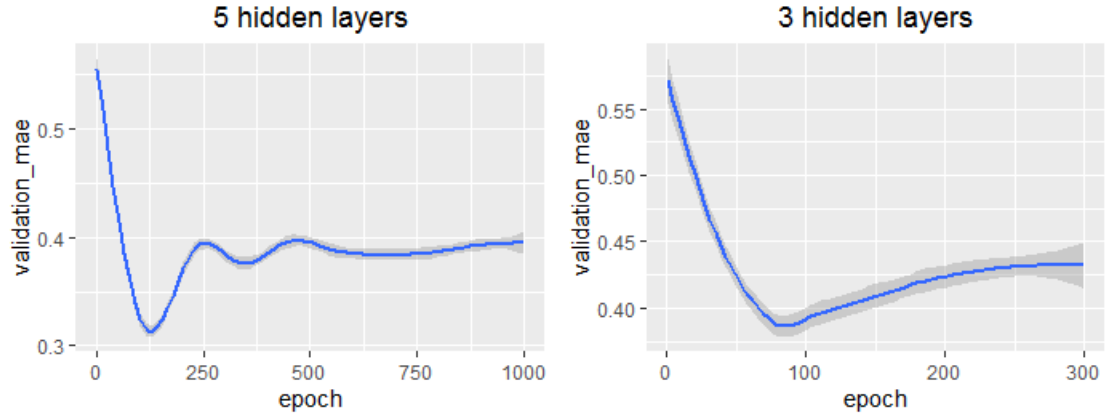

Figure 4. ANN performance improves by increasing 3 hidden layers to 5. Performance drops off when hidden layers are increased to eight.

Comparison between models

The artificial neural network achieved a higher accuracy than the random forest model on most of the test data but performed noticeably worse during the final two Covid-19 affected quarters in 2020.

The time taken to train the artificial neural network is considerably higher as the ANN undertakes substantially more calculations. Choosing the right number of neurons and layers took many hours. The tuning of random forest hyperparameters while still slow seemed a more straightforward.

Performance can be improved in an artificial neural network by adjusting the number of hidden layers and neurons. These neurons jointly implement a complex nonlinear mapping from input to the output (Montavon et. al, 2018). The inner workings are commonly seen to be a "black box" and can be difficult to interpret and explain.

The random forest is more intuitive with changes made directly to the parameters affecting the model output.

Suggestions for improvement

Despite best efforts, the random seed was unable to be correctly set for the artificial neural network (in TensorFlow and Keras). This introduced variability in the model output that hampered the fine-tuning of neurons and layers. The model was fit to the training data with a four-fold cross validation that was repeated and averaged. This repetition slowed down the process and subtracted from the time available for model tuning.

Both models performed poorly when predicting the unemployment rate for the final two quarters in the test data set (June and September 2020). The sudden impact of the government's Covid-19 restrictions on the unemployment rate could not be accurately modelled with the data supplied. The final quarter in particular was difficult to model. This is likely due to the healthy rise in GDP (3.3%) and an improvement in job vacancies from the previous quarter that provided a false appearance of an economy which would display a lower unemployment rate. Job vacancies are a leading economic indicator and will impact the unemployment rate in the future, whereas the unemployment rate is a lagging economic indicator which represents past economic conditions. As time series were not considered in the analysis, the model is not able to adjust the unemployment rate prediction given the higher number in the previous quarter.

The brief overview of the Australian unemployment rate highlighted some of the major national and international events that have impacted economic conditions in Australia since the turn of the century. The major economic factors that have helped shape labour market trends should be included in the model. Mining investment and construction activity should be included to better model employment in those industries. Likewise, health care expenditure should be included to cover a large and growing industry in Australia.

Gradual changes in conditions are easier to model than extreme events like a global pandemic. A model trained using data from normal business cycles cannot be used to predict unemployment during a once in a hundred-year event like Covid-19.

References

Australian Bureau of Statistics. 2009. 1364.0.15.003 – *Modellers' Database, Dec 2009.* Retrieved from: https://www.abs.gov.au/AUSSTATS/abs@.nsf/DetailsPage/1364.0.15.003Dec%202009?OpenDocument

Australian Bureau of Statistics. 2010. 6354.0 - *Job Vacancies, Australia, Feb 2010 Explanatory Notes.*

Australian Bureau of Statistics. 2010. *1504.0 – Methodological News, Dec 2010. Dealing with a Break in Series – Job Vacancies.*

Australian Bureau of Statistics. 2020. *Labour Force, Australia.* Retrieved from: https://www.abs.gov.au/statistics/labour/employment-and-unemployment/labour-force-australia/dec-2020

Australian Bureau of Statistics. 2020. *National, state and territory* population. Retrieved from: https://www.abs.gov.au/statistics/people/population/national-state-and-territory-population/latest-release#data-download

Borland, J. 2011. *The Australian Labour Market in the 2000s. In: The Australian Economy in the 2000s,* Reserve Bank of Australia.

Connolly, E., and Orsmond D. 2011. *The Mining Industry: From Bust to Boom. In: The Australian Economy in the 2000s.* Reserve Bank of Australia.

Debelle, G. *The State of the Labour Market.* 2018. Citi 10th Annual Australia and New Zealand Investment Conference. Retrieved from: https://www.rba.gov.au/speeches/2018/sp-dg-2018-10-17.html

Kearns, J., and Lowe P. 2011. *Australia's Prosperous 2000s: Housing and the Mining Boom. In: The Australian Economy in the 2000s,* Reserve Bank of Australia.

Kent, C. 2014. *Cyclical and structural changes in the Labour Market.* Address on Labour Market Developments, hosted by The Wall Street Journal. Reserve Bank of Australia. Retrieved from: https://www.rba.gov.au/speeches/2014/sp-ag-160614.html

Kent, C. 2015. *Adjustments in the Labour Market.* Reserve Bank of Australia. Address to the Economic Society of Australia (QLD) Luncheon. Retrieved from: https://www.rba.gov.au/speeches/2015/sp-ag-2015-08-14.html

Montavon, G., Samek, W., and Müller, K. 2018. *Methods for interpreting and understanding deep neural networks.* Digital Signal Processing (73) 1-15.

Nordhaus, W., D. 2002. *The Mildest Recession: Output, Profits, and Stock Prices as the U.S. Emerges from the 2001 Recession.* National Bureau of Economic Research. Working Paper 8938.

Lowe, P. 2020. *COVID-19, the Labour Market and Public Sector Balance Sheets.* Address to the Anika Foundation. Reserve Bank of Australia. Retrieved from: https://www.rba.gov.au/speeches/2020/sp-gov-2020-07-21.html

Appendix

```
library(readxl)

library(dplyr)

library(caret)

library(ranger)

library(mlr)

library(ggplot2)
```

```
library(tensorflow)
tf$random$set_random_seed(1234)

library(keras)
use_session_with_seed(1234)

##########################################
# Only required once
#install python packages
library(reticulate)
#create a new environment
conda_create("r-reticulate")
#install tensorflow
conda_install("r-reticulate", "tensorflow")
##########################################

# Question 2a) - prepare data
# Import the spreadsheet into an R data frame
# Skip the first row and use the second row as headers
col_names <- array(read_excel('AUS_Data.xlsx', skip=1, n_max = 1, col_names = FALSE))
data <- data.frame(read_excel('AUS_Data.xlsx', skip = 2, col_names = FALSE))
colnames(data) <- col_names

# Abbreviate the column names
colnames(data) <- c("period", "unemployment_rate", "GDP",
                    "government_consumption", "all_sectors_consumption",
                    "terms_of_trade", "CPI", "job_vacancies", "population")

# Move the period column to the right of the data frame
# Change column to date format
```

```
data <- data %>% select(-period, period)
data$period <- as.Date(data$period, origin="1899-12-30")

# Question 2b) - descriptive statistics
# Use box plot to visualise the data distribution for each variable
data_stacked <- stack(data[1:4])
ggplot(data_stacked, aes(x="", y=values)) +
  ggtitle ("Distribution of data") +
  theme(plot.title = element_text(hjust = 0.5),
        axis.title.x=element_blank()) +
  geom_boxplot(outlier.color="red", outlier.shape=3) +
  geom_jitter(width=0.1, alpha=0.05, color="blue") +
  facet_wrap( ~ ind, scales="free")

# Question 3. Apply one supervised ML algorithm - random trees
# Split data: train (June 81 - Dec 2017), test (Mar 18 - Sep 20)
train <- subset(data, period < as.Date("2018-03-01"))
test <- subset(data, period >= as.Date("2018-03-01"))

# Remove period data
train <- subset(train, select = -(period))
test <- subset(test, select = -(period))

# Define task and learner
train.task <- makeRegrTask(data = train, target = "unemployment_rate")
learner <- makeLearner("regr.ranger")

set.seed(1234)
# Choose resampling strategy and define grid
rdesc <- makeResampleDesc("RepCV", folds = 10, reps = 10)
ps <- makeParamSet(makeDiscreteParam("mtry", 5),
```

```
            makeDiscreteParam("importance", "permutation"),
            makeDiscreteParam("min.node.size", 2),
            makeDiscreteParam("num.trees", 950))

# Tune parameters
res = tuneParams(learner, train.task, rdesc, par.set = ps,
            control = makeTuneControlGrid())

# Train on dataset using best hyperparameters
lrn = setHyperPars(makeLearner("regr.ranger"), par.vals = res$x)
m <- mlr::train(lrn, train.task)

# Make predictions for unemployment rate
cat("OOB prediction error (MSE) on training data is ", m$learner.model$prediction.error)
pred_test_RF <- predict(m$learner.model, data = test)
print(pred_test_RF$predictions)
print(test$unemployment_rate)

# Implement artificial neural network
# Apply normalisation to data
mean <- apply(train, 2, mean)
std <- apply(train, 2, sd)
train <- scale(train, center = mean, scale = std)
test <- scale(test, center = mean, scale = std)

# Separate unemployment rate as target
train_targets <- subset(train, select = unemployment_rate)
train <- subset(train, select = -(unemployment_rate))
test_targets <- subset(test, select = unemployment_rate)
test <- subset(test, select = -(unemployment_rate))
```

```
# Build neural network
build_model <- function() {
  k_clear_session()
  set.seed(1234)
  tf$random$set_random_seed(1234)
  model <- keras_model_sequential() %>%
    layer_dense(units = 7, activation = "relu",
                input_shape = dim(train)[[2]]) %>%
    layer_dense(units = 7, activation = "relu") %>%
    layer_dense(units = 7, activation = "relu") %>%
    layer_dense(units = 7, activation = "relu") %>%
    layer_dense(units = 7, activation = "relu") %>%
    layer_dense(units = 7, activation = "relu") %>%
    layer_dense(units = 7, activation = "relu") %>%
    layer_dense(units = 1)

  model %>% compile(
    optimizer = "rmsprop",
    loss = "mse",
    metrics = c("mae")
  )
}

k <- 4
indices <- sample(1:nrow(train))
folds <- cut(indices, breaks = k, labels = FALSE)
num_epochs <- 1000
all_mae_histories <- NULL
for (i in 1:10) {
  for (i in 1:k) {
    cat("processing fold #", i, "\n")
```

```
    # Prepare the validation data: data from partition # k
    val_indices <- which(folds == i, arr.ind = TRUE)
    val_data <- train[val_indices,]
    val_targets <- train_targets[val_indices]

    # Prepare the training data: data from all other partitions
    partial_train_data <- train[-val_indices,]
    partial_train_targets <- train_targets[-val_indices]

    # Build the Keras model (already compiled)
    model <- build_model()

    # Train the model (in silent mode, verbose=0)
    history <- model %>% fit(
      partial_train_data, partial_train_targets,
      validation_data = list(val_data, val_targets),
      epochs = num_epochs, batch_size = 1, verbose = 0
    )
    mae_history <- history$metrics$val_mean_absolute_error
    all_mae_histories <- rbind(all_mae_histories, mae_history)
  }
}

# compute the average of the per-epoch MAE scores for all folds
average_mae_history <- data.frame(
  epoch = seq(1:ncol(all_mae_histories)),
  validation_mae = apply(all_mae_histories, 2, mean)
)

# Plot validation scores
```

```
ggplot(average_mae_history, aes(x = epoch, y = validation_mae)) + geom_smooth() +
  ggtitle ("14 neurons") +
  theme(plot.title = element_text(hjust = 0.5))

mea_test <- 0
mea_train <- 0
average_predictions <- 0
k <- 50
# Fit final model and make predictions on test data
for (i in 1:k) {
  k_clear_session()
  set.seed(1234)
  tf$random$set_random_seed(1234)
  model <- build_model()
  set.seed(1234)
  tf$random$set_random_seed(1234)
  model %>% fit(train, train_targets,
                epochs = 300, batch_size = 16, verbose = 0)
  result_test <- model %>% evaluate(test, test_targets)
  mea_test <- mea_test + result_test$mean_absolute_error
  result_train <- model %>% evaluate(train, train_targets)
  mea_train <- mea_train + result_train$mean_absolute_error

  predictions <- model %>% predict(test)
  predictions_unscaled <- t(predictions) * std[1] + mean[1]
  average_predictions <- average_predictions + predictions_unscaled
}

cat("average mea_train is", mea_train/k)
cat("average mea_test is", mea_test/k)
cat("average predicitons are", average_predictions/k)
```