ASSESSMENT 2: DATA ANALYSIS

Part A: Proposal

36103 Statistical Thinking for Data Science

University Of Technology Sydney

Group Name

Retail Group

Group Members

Leah Nguyen

Tony Tan

Anmol Mittal

Ben McKinnon

Kasun Caldera

Manasa Burli Nagendra

Paul Touhill

Date

September 5, 2021

Contents

1 Introduction

Death from road traffic crashes continue to be major global public health problems across Australia (of Health & Welfare 2021). A road traffic fatality results from a complex interaction of **human**, **technical** and **environmental** factors, and different causes behind road vehicle crashes require different measures to reduce their impacts.

2 Methodology

CRISP-DM methodology will be used for this project which is used in most of the data science projects in the industry (Wijaya 2021). The six steps are as follows:

- Business Understanding
- Data Understanding
- Data Preparation
- Modeling
- Evaluation
- Deployment (Not in the scope of this assignment)

3 Business Understanding

Our group study assesses how strongly the variation between some environmental factors and the increased risk of vehicle crash-related mortality in Victoria, Australia. The findings are relevant in the context of impact-based warnings for both road users, road maintenance and traffic management authorities, as well as rescue forces.

On the one hand, this is relevant for raising the effectiveness of road weather warning services, while on the other hand, it may help rescue and medical services to project the number of emergencies in designated regions in case weather forecasts foresee bad conditions for road transport.

4 Data Understanding

4.1 Accident

Accident datasets are datasets with anonymized information from government of all road accidents in Victoria, Australia from 2006 until 2020 is used (Department of Transport 2006). The data provided allows users to analyse Victorian fatal and injury crash data based on time, location, conditions, crash type, road user type, object hit etc.

Technique: R, EDA

4.2 Weather

Weather datasets contains about daily weather observations describing daily mean, maximum, and minimum temperature and precipitation for the year of 2006 - 2020 from many locations across Australia. We collected the data using the Visual Crossing Weather API to perform a preliminary exploration to determine what, if any, predictive value weather has on crash and injury data when analyzed at the level of incident.

5 Data Preparation

5.1 Merging accident data

The car accident data will be sourced directly from the Victorian Government data website (Appendix 1). Not all files that are extracted from the website will be used in the analysis. After selecting specific files by their file name, they will be loaded into separate variables ready to be merged. The `PERSON` dataset has been selected as the base file because it holds the largest amount of unique information.

5.2 Merging weather data

The visual crossing weather API has a query which can search by location including the use of postcode. Using a collection of all the distinct postcodes from the node dataset, we can then make an API request for the address. As the API response contains a list of records for the date range specified in its query, we can then match each of our accident records with the weather using the postcode for location and the date.

6 Model

6.1 Classification Model – Generalised Linear Model

We will first begin training a logistic regression model. The model will be trained on years 2006 – 2017 and tested on years 2018-2020. A binary variable to identify if there was a fatality in the car accident will be created and set as our target variable. To test several model fits, we will use backward selection method to determine which features should be used in the final model.

7 Evaluation

8 Limitations

We don't have the data for cars driving that didn't get into an accident. Therefore, instead of predicting accidents, we are predicting deaths when accidents do occur. The issue here is we cannot quite research how effective a factor is at preventing deaths, as a death causing accident could have been prevented altogether due to this factor. Yet, on data, this factor may seem to always lead to deaths, because of negligent drivers.

9 Bibliography

Department of Transport (2006), 'Victoria Car Accident Dataset 2006 - 2020'.

URL:

https://vicroadsopendatastorehouse.vicroads.vic.gov.au/opendata/Road_Safety/ACCIDENT.zip

of Health, A. I. & Welfare (2021), Injury in australia: transport injuries, Report, AIHW.

URL: *https://www.aihw.gov.au/reports/injury/transport-injuries*

Wijaya, C. Y. (2021), 'CRISP-DM Methodology For Your First Data Science Project'.

URL: *https://towardsdatascience.com/crisp-dm-methodology-for-your-first-data-science-project-769f35e0346c*

10 Appendix(ces)

10.1 Appendix A: Loading libraries

```
knitr::opts_chunk$set(eval = FALSE)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.3     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(here)
```

```
## here() starts at C:/Users/LEAH NGUYEN/OneDrive/Desktop/GitHub/UTS-STDS2021
```

```
library(httr)
library(rjson)
library(foreign)
```

10.2 Appendix B: Road Accident Data Collection Code

```
knitr::opts_chunk$set(eval = FALSE)

# Set where zip file is to be saved
file_path <- here('data', 'ACCIDENT')
setwd(file_path)

# Download and extract data
url <- 'https://vicroadsopendatastorehouse.vicroads.vic.gov.au/opendata/Road_Safety/ACCIDENT.zi
download.file(url, 'CarAccidentsData.zip')
unzip('CarAccidentsData.zip')

# Place selected files into variables
f <- file.path(file_path,
           c("ACCIDENT.csv","ACCIDENT_EVENT.csv", "ACCIDENT_LOCATION.csv", "ATMOSPHERIC_COND.cs
             'NODE.csv','NODE_ID_COMPLEX_INT_ID.csv','PERSON.csv', 'ROAD_SURFACE_COND.csv',
             'SUBDCA.csv','VEHICLE.csv'))


# Create names for the variables and remove any characters after '.'
names(f) <- gsub(".*/(.*)\\..*", "\\1", f)

# Read files into variables ready for analysis
for (i in 1:length(f)){
    x= read_csv(f[i])
    names(x)<- gsub(' ','_', names(x))
    assign(names(f[i]),x)
    remove(x)
  }
```

10.3 Appendix C: Weather Data Collection Using API

```
knitr::opts_chunk$set(cache = TRUE)
# Load postcodes and variables ----
postcodes <- fromJSON(file = "sample.json") # json with list of postcodes
RAPIDAPI_KEY = 'd1d5ff8ef9msh03f3fb1acd367a2p14e523jsnf314364cf14f' # can vary with different c

# Create functions to use ----
request_by_postcode_and_year <- function(postcode, yearStart){
  base_url <- "https://visual-crossing-weather.p.rapidapi.com/"
  path <- "history"
  query_string <- list(
    startDateTime = sprintf('%s-01-01T00:00:00', yearStart),
    aggregateHours = '24',
    location = sprintf('%s,VIC,AUS', postcode),
    endDateTime = sprintf('%s-12-31T00:00:00', yearStart),
    contentType = 'csv',
    shortColumnNames = '0'
  )

  response <- GET(
    url = base_url,
    path = path,
    add_headers(
      'x-rapidapi-host' = 'visual-crossing-weather.p.rapidapi.com',
      'x-rapidapi-key' = RAPIDAPI_KEY
    ),
    query = query_string,
    content_type('application/octet-stream'))

  text <- content(response, "text")
  return(text);
}
```

```r
gather_by_postcodes_and_year <- function(postcodes, year){
  final_df <- data.frame(matrix(ncol = 0, nrow = 0)) # initialize empty data frame
  for(postcode in postcodes){
    csv_response_text = request_by_postcode_and_year(postcode, year)
    df_from_response <- read_csv(csv_response_text)
    final_df <- bind_rows(final_df, df_from_response)
    cat("|") # some feedback on console
    Sys.sleep(1) # go easy on the api just in case
  }
  return(final_df)
}

create_csv <- function(df, file_name){
  if(!dir.exists("data")){
    dir.create("data")
  }

  write_csv(df, sprintf("data/%s", file_name))
}

create_csv_data_for_year <- function(year){
  combined_postcodes_df = gather_by_postcodes_and_year(postcodes, year)

  new_file_name = sprintf("sample_weather_%s.csv", year)
  create_csv(combined_postcodes_df, new_file_name)
}

# MAIN LOGIC STARTS HERE ----

# for(year in 2006:2020){
#   create_csv_data_for_year(year)
# }
```

```
create_csv_data_for_year(2020)
```

10.4 Appendix D: Accident: Merge Datasets

```
knitr::opts_chunk$set(cache = TRUE)
# Base Table
PERSON <- PERSON %>%
          select(-LICENCE_STATE, -PEDEST_MOVEMENT, -POSTCODE, -TAKEN_HOSPITAL, -EJECTED_CODE)

ACCIDENT <- ACCIDENT %>%
              select(-DIRECTORY, -EDITION, -PAGE, -GRID_REFERENCE_X,
                     -GRID_REFERENCE_Y, -POLICE_ATTEND, -ROAD_GEOMETRY)
BASE <- left_join(PERSON, ACCIDENT, by='ACCIDENT_NO') %>%
          left_join(x=., ROAD_SURFACE_COND, by='ACCIDENT_NO')

# Location
LOCATION <- left_join(NODE,ACCIDENT_LOCATION %>%
                        select(ACCIDENT_NO,NODE_ID, ROAD_NAME, ROAD_TYPE, ROAD_TYPE_INT),
                      by='ACCIDENT_NO' )
# Weather
ATMOSPHERIC_COND

# Final Dataset for Analysis
data <- left_join(BASE, LOCATION,by='ACCIDENT_NO') %>%
          left_join(x=.,ATMOSPHERIC_COND,by='ACCIDENT_NO')



data
```

#———————————————————————————————————- # Data Cleaning

#———————————————————————————————————-

```
knitr::opts_chunk$set(cache = TRUE)
data <- data %>%
  mutate(FATAL_ACCIDENT = if_else(NO_PERSONS_KILLED>0,"Y","N")) %>%
  relocate(FATAL_ACCIDENT ,.after = ACCIDENT_NO)  %>%
```

```
  relocate(NO_PERSONS_KILLED ,.after = FATAL_ACCIDENT)
```

10.5 Appendix E: Weather Data: Merge Datatsets

```
knitr::opts_chunk$set(cache = TRUE)
# get column names
df <- read_csv(here('data', 'weather_data', 'weather_2006.csv'))
col_names <- colnames(df)
```

```
# define new column names
added_cols <- c("Postcode", "State", "Country")
final_cols <- paste0(added_cols, col_names)

# create vector of years in dataset
years <- c('2006', '2007', '2008', '2009', '2010', '2011', '2012', '2013',
           '2014', '2015', '2016', '2017', '2018', '2019', '2020A', '2020B')

# create appending_df
final_weather <- data.frame(matrix(nrow = 0, ncol = 27))
# apply column names
names(final_weather) <- final_cols

# loop over years and split postcode column and concat to 1 df
for (year in as.list(years)){
  csv_file <- sprintf('weather_%s.csv', year)

  # Read csv
  weather <- read_csv(here('data', 'weather_data', csv_file))
  # split column
  weather <- weather %>%
    separate(Address, c("Postcode", "State", "Country"), sep = ',')

  final_weather <- rbind(final_weather, weather)
}
```

```
write_csv(final_weather, here('data', 'weather_data', "clean_weather.csv"))
```