Discover necessity for Greenhouse Emission Test on Vehicle in New Zealand, based on datamining in last 15 years Fleets and Emission data

Tian Bai
tbai915

461962597

Table of Content

1.1 Business understanding

New Zealand as a member in the United Nations responding to the 17 sustainable development goals to provide future generations an eligible developing circumstance. The goal 13 mentioned about Climate Crisis which New Zealand arrange many different government agencies working form different ways to improve it. Nowadays, human is facing the climate crisis and the World meteorological organization confirmed “The year 2019 was the second warmest year on record after 2016, according to the World Meteorological Organization’s consolidated analysis of leading international datasets” (WMO confirms 2019 as second hottest year on record, para. 1).

The major department in this GOAL is Ministry for the Environment (MFE) in New Zealand. MFE mentioned the main pathway for improve climate crisis is to control the Greenhouse gas which made by Carbon Dioxide (CO_2), Methane and so on. On one hand, we are able to reduce the existing CO_2 to increase the area of plants in New Zealand. On the other hand, is to control the CO_2 emission. Biofuels still place a key role in human transport, people are able to control the exhaust emission for reduce the CO_2.

The Ministry of Transport apply stricter emission standard on the new imported fleets and also encourage people or commercial to select electric or hybrid fleets. However, there are still have a significant number of used vehicles in New Zealand. The question is, does the used vehicles can maintain the emission standard? There are no other tests will be introduced to the Greenhouse emission test comparing with other countries such as Germany and China. The current smoky exhaust test contained in warrant of fitness (Wof), based on the colour of smoke, and it will not test on Greenhouse Emission.

1.1.1 Study of Business

Hence, a study of this situation is about discovering the pattern emission distribution between different categories of fleets on the last 15 years data in New Zealand. The purpose is The Ministry of Transport currently could estimate the totally fleets emission annual, however, it will be hard to statistic emission information to each single vehicle and they cannot estimate the partitions for different age range of vehicles.

1.1.2 Business success goal

Therefore, the success goals of the study will be introduced with the following:

1. The models could accurate estimate the predict emission values against real values.

2. The models are able to predict partial emission value by input partial group statistics within one fleet group.

Suppose one fleet group able to category by different age ranges, input all the statistics from the whole group will get the totally emission predict value. If input each category statistics will return partial emission predict value. Under accurate models, blocking other fleets and remain a specific type of fleet inputs in one year, able to get a predict value of single type against totally predict value to estimate the emission partition of this type of fleet.

1.2 Assess the situation

For divide fleets emission data by age groups which require the data contains number of fleets, average age of fleets and corresponding CO2 emission. Since New Zealand have a fully vehicle registration system, these data are findable from The Ministry of Transport.

1.2.1 Resource inventory

The initial data collect from Ministry of Transport at page (https://www.transport.govt.nz/mot-resources/vehicle-fleet-statistics/).

Python, PyCharm, Sci-kit learn, Pandas, yellowbrick, matplotlib, numpy, SPSS, PySpark, were used for all three iterations.

1.2.2 Requirements, assumptions and constraints

The requirements are the project participants needs to have some statistical base and in good knowledge of the above-mentioned software and language.

The assumption of this project is:

- As the Euro 5 Emission Standard published in 2009 and Euro 6 Emission published 5 years later, there will be over 50% fleets with age more than 10 years and this group will contribute over 70% CO2 emission of totally fleets.

The constraints for this project are:

- These data should be corresponding and continues.
- For instance, the emission data should be corresponding with the data of fleets during a same time period.

1.2.3 Risks and contingencies

There are risks:

- If some type of vehicles cannot split by age, in this case, if the majority group can split by age range then could dismiss the minor cannot groups.
- Since, in New Zealand the majority type of traveling resources are light fleets, the other less common fleets types, for example, heavy fleets might not able to find the age distribution as its minor amounts. We are using the majority sample of population to estimate the total population, this method is acceptable in Statistic and it is commonly used for population census

ISAS risks:

- SPSS may not be accurate enough, and SPSS might not have many model options to choose from and more customize as the Sci-kit Learn library in Python does.

OSAS risks:

- Python visualization uses matplotlib, its common visualization is to generate the result as figures, therefore if the result needs to be demonstrated the whole program will need to be paused while demonstrating the output figure.

BSAS risks:

- As the use of git, this make the project go opensource, there are chances that it will occur a number of opensource merges, this require time and efforts to manage the merges or unwanted merges happens.

1.2.4 Benefit analysis

Opensource enables other participants to join the project and this gives a better iteration of the program.

1.3 Datamining objectives

- The model built would able to estimate the Co2 emission from the number of flees and its average ages inputs. The estimated Co2 emission data is combined for multiple fleets type groups with over 80% accuracy.
- Assume the data would inputs many different types of fleets, the success model would still able to estimate parts of the Co2 emission with partial types of inputs with over 80% accuracy.
- In future, if Ministry of Transport considers Co2 emission is important to be considered and collecting more data in detail, there are more chance to obtaining a more accurate model.

1.3.1 Data mining success criteria

- The first success criterion is when input values fleets to models reflect predict values of CO2 emission and it will have a lower residual with real values of CO2 emission thus to count the accuracy, $Real\ Value - Predict\ Value$.
- The second success criterion is the sum of the predict values within different groups should also not have a large residual compare with the predict values, $\sum single\ group\ predict\ value\ -\ real\ value$.

1.3.1.1 Methods for model assessment

- Running the model. Assume after input the features such as numbers of fleets and age of the fleets to the model, we will obtain an estimate Co2 emission. If the input features are all real data from the same year, the obtained estimate Co2 emission will be very close to the true Co2 emission from that year. If all the input features are 0, then then Co2 emission would be 0, as if there are no fleets in New Zealand, the total Co2 emission from fleets would be none.

- To evaluate the model's accuracy. Assume there are 3 features needs to be inputted to the model to estimates Co2 emission, by running the model 3 times with input each feature's value once while inputs the rest 2 features as 0.
  for example:
  - the first run: feature1's real value, 0, 0
  - the second run: 0, feature2's real value,0
  - the third run: 0,0, feature3's real value

  Each run will obtain an estimate Co2 emission, the total sum of these 3 estimated Co2 emission will be very close to the Co2 emission estimates obtained by input all 3 features' value to the modal. By this method, we are able to evaluate the accuracy of the model by comparing the total Co2 emission sum of each run with the estimates of Co2 emission gathered by input all 3 features.

1.3.1.2 Benchmark for evaluating success

- $small\ residuals$: $Real\ Value - Predict\ Value$
- $\sum single\ group\ predict\ value\ -\ real\ value.$
- We are able to evaluate the feature's distribution on the Co2 mission with the same method mentioned above to evaluate the model's accuracy.
  $Co2\ Emission\ obtained\ each\ run / Co2\ Emission\ obtained\ by\ inputs\ all\ 3\ features$

1.3.1.3 Subjective measurements and the arbiter of success

In this study, the measurement of Co2 emission is predicable to the successful model.

1.3.1.4 Successful deployment of model results as a part of data mining success

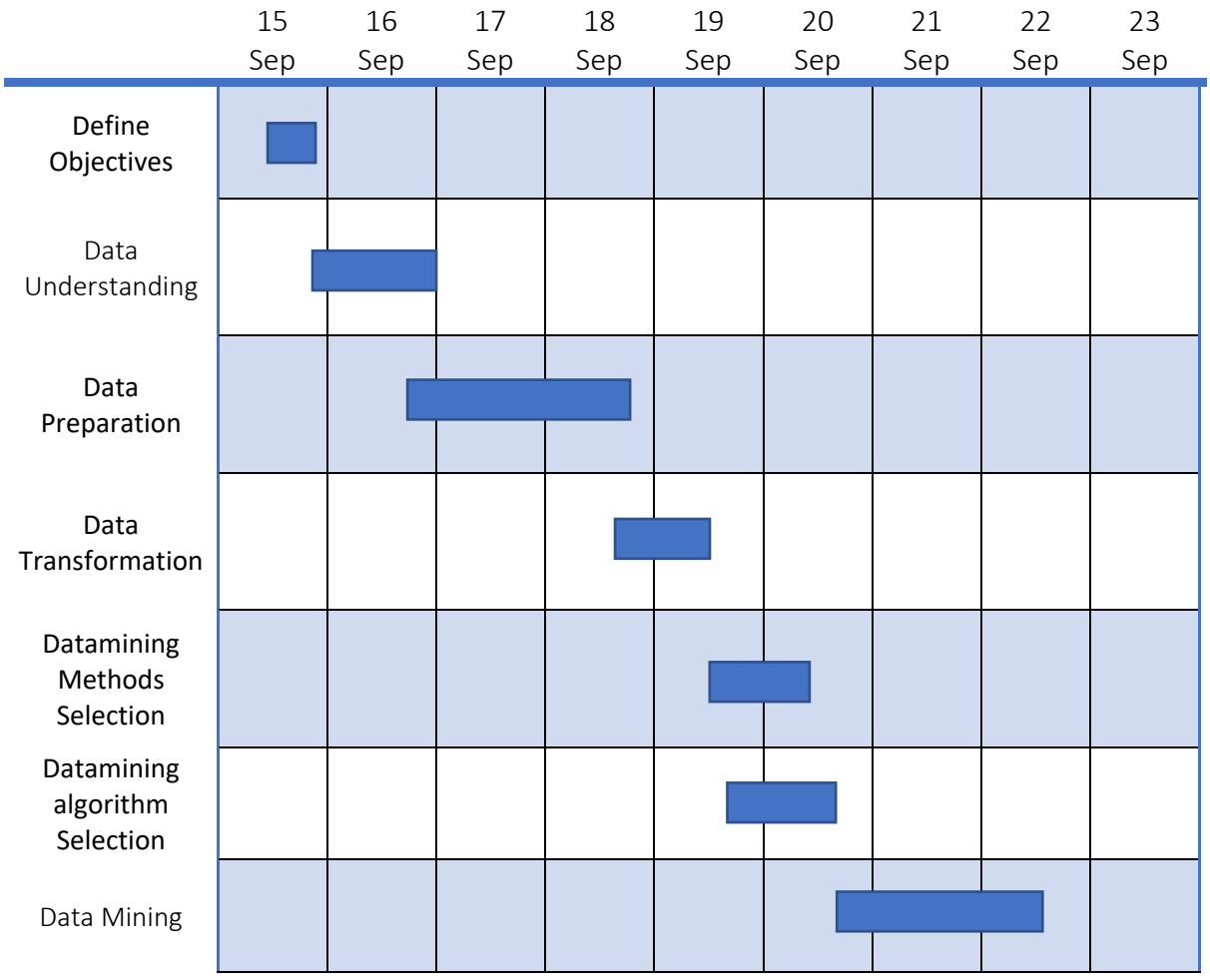
The reason is the input values might contain different fleets groups values, when input each group's values separately from total fleets group in the models, the predict value output of that one group against the total groups predict value will reflect the CO2 contribution for this group, $single\ group\ predict\ value / total\ groups\ predict\ value$.

1.4 Project plan

Phase	Time	Risks
Iteration1 Proposal	24th July – 7th Aug (2 weeks)	• Unable to find suitable data
Iteration2 ISAS	7th Aug – 28th Aug (3 weeks)	• SPSS may not be accurate enough • SPSS might not have many model options to choose from and more customize as the Sci-kit Learn

		library in Python does.
Iteration3 OSAS	28th Aug – 2th Oct (6 weeks)	• When the result needs to be demonstrated the whole program will need to be paused while demonstrating the output figure
Iteration4 BDAS	9th Oct – 23th Oct (2 weeks)	• Require time and efforts to manage the merges or unwanted merges happens
Research paper	23th Oct – 30th Oct (1 week)	

1.4.1 Gantt chart

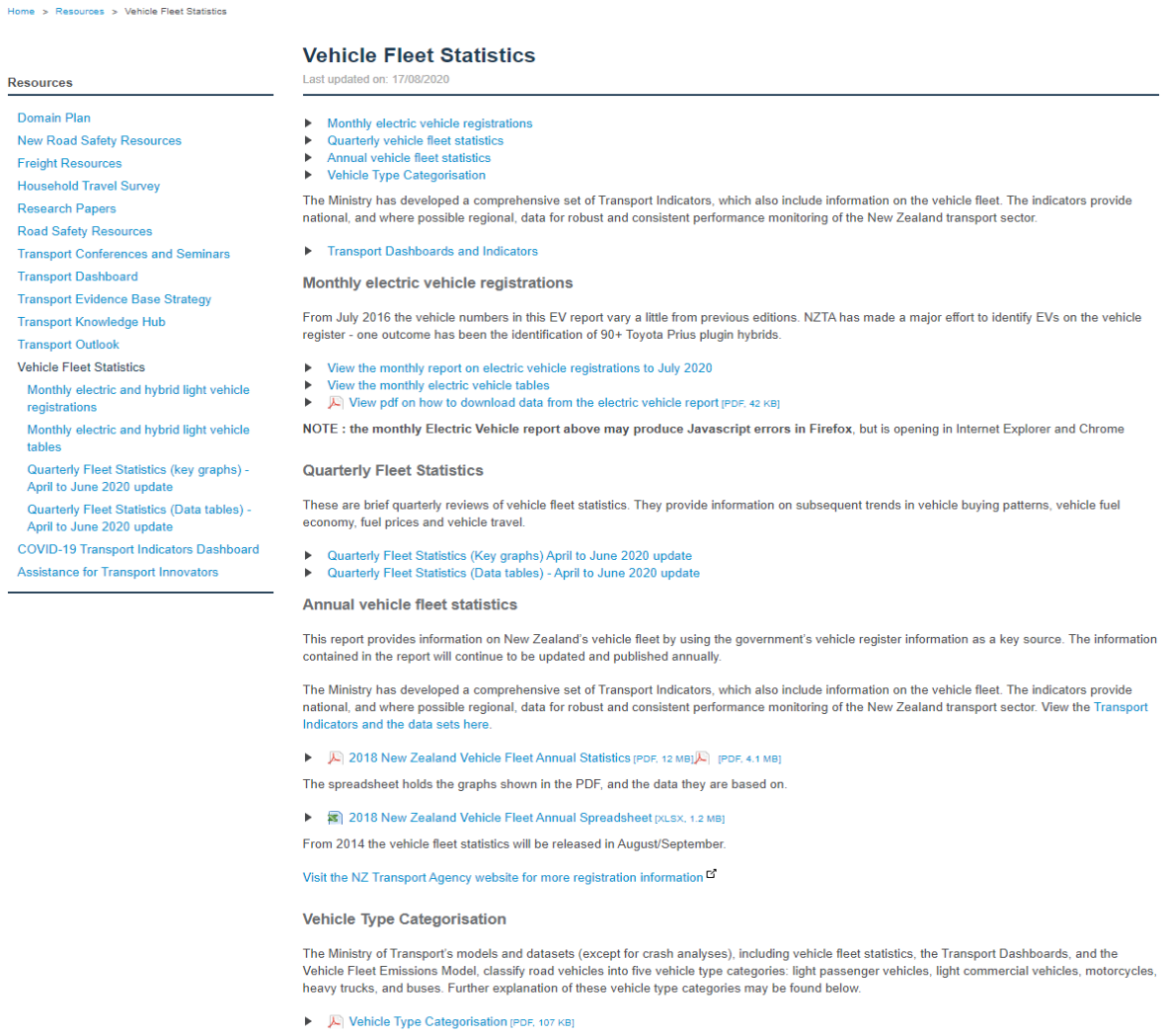


Interpretation									

2.1 Data Collection

The initial data collect from Ministry of Transport at page (https://www.transport.govt.nz/mot-resources/vehicle-fleet-statistics/). After initial data inspection, the data of NZ-Vehicle-Fleet-Statistics-2018_web.xlsx is a combination of different data tables in an excel file. it provides all the data the project needed, they are number of different fleets, age distribution between different fleets and Co2 emission for different fleets. There are some other data tables might use for mining in further insights, for example, average travel milage for different fleets.

A snapshot from this webpage: Download "2018 New Zealand Vehicle Fleet Annual Spreadsheet [XLSX, 1.2 MB] ". (Figure 1)



Home > Resources > Vehicle Fleet Statistics

Resources

- Domain Plan
- New Road Safety Resources
- Freight Resources
- Household Travel Survey
- Research Papers
- Road Safety Resources
- Transport Conferences and Seminars
- Transport Dashboard
- Transport Evidence Base Strategy
- Transport Knowledge Hub
- Transport Outlook
- Vehicle Fleet Statistics
  - Monthly electric and hybrid light vehicle registrations
  - Monthly electric and hybrid light vehicle tables
  - Quarterly Fleet Statistics (key graphs) - April to June 2020 update
  - Quarterly Fleet Statistics (Data tables) - April to June 2020 update
- COVID-19 Transport Indicators Dashboard
- Assistance for Transport Innovators

Vehicle Fleet Statistics

Last updated on: 17/08/2020

- Monthly electric vehicle registrations
- Quarterly vehicle fleet statistics
- Annual vehicle fleet statistics
- Vehicle Type Categorisation

The Ministry has developed a comprehensive set of Transport Indicators, which also include information on the vehicle fleet. The indicators provide national, and where possible regional, data for robust and consistent performance monitoring of the New Zealand transport sector.

- Transport Dashboards and Indicators

Monthly electric vehicle registrations

From July 2016 the vehicle numbers in this EV report vary a little from previous editions. NZTA has made a major effort to identify EVs on the vehicle register - one outcome has been the identification of 90+ Toyota Prius plugin hybrids.

- View the monthly report on electric vehicle registrations to July 2020
- View the monthly electric vehicle tables
- View pdf on how to download data from the electric vehicle report [PDF, 42 KB]

NOTE : the monthly Electric Vehicle report above may produce Javascript errors in Firefox, but is opening in Internet Explorer and Chrome

Quarterly Fleet Statistics

These are brief quarterly reviews of vehicle fleet statistics. They provide information on subsequent trends in vehicle buying patterns, vehicle fuel economy, fuel prices and vehicle travel.

- Quarterly Fleet Statistics (Key graphs) April to June 2020 update
- Quarterly Fleet Statistics (Data tables) - April to June 2020 update

Annual vehicle fleet statistics

This report provides information on New Zealand's vehicle fleet by using the government's vehicle register information as a key source. The information contained in the report will continue to be updated and published annually.

The Ministry has developed a comprehensive set of Transport Indicators, which also include information on the vehicle fleet. The indicators provide national, and where possible regional, data for robust and consistent performance monitoring of the New Zealand transport sector. View the Transport Indicators and the data sets here.

- 2018 New Zealand Vehicle Fleet Annual Statistics [PDF, 12 MB] [PDF, 4.1 MB]

The spreadsheet holds the graphs shown in the PDF, and the data they are based on.

- 2018 New Zealand Vehicle Fleet Annual Spreadsheet [XLSX, 1.2 MB]

From 2014 the vehicle fleet statistics will be released in August/September.

Visit the NZ Transport Agency website for more registration information

Vehicle Type Categorisation

The Ministry of Transport's models and datasets (except for crash analyses), including vehicle fleet statistics, the Transport Dashboards, and the Vehicle Fleet Emissions Model, classify road vehicles into five vehicle type categories: light passenger vehicles, light commercial vehicles, motorcycles, heavy trucks, and buses. Further explanation of these vehicle type categories may be found below.

- Vehicle Type Categorisation [PDF, 107 KB]

Figure 1: Screenshot for NZTA page.

2.2 Data Description

The NZ-Vehicle-Fleet-Statistics-2018_web.xlsx is a combination of different data tables in a excel file.

For the data that project need:

- Number of Fleets:
  "Composition of Fleet data at figure 2.10 2.20 2.30 2.40
- Age Distribution of Fleets:
  Light Vehicles age distribution data at figure 2.10
- Co2 Emission Data:
  CO2 emissions data at figure 1.10

The New Zealand 2018 Vehicle Fleet : Data Spreadsheet
Version 5.0, October 2019

Analysis produced by Ministry of Transport
info@transport.govt.nz

Overview
Figure 1.1 Composition of the NZ Fleet
Figure 1.1 extra Vehicle average ages
Figure 1.2 Composition of the NZ fleet relative to Jan 2000
Figure 1.3a,c Travel by light passenger/light commercial/other

Figure 1.4 Light fleet travel by year
Figure 1.5 Light fleet ownership per capita by year
Figure 1.5b,c Regional light fleet ownership per capita
Figure 1.6 Light fleet travel per capita by year
Figure 1.7 Light fleet average vehicle travel by year
Table Fleet travel by vehicle type
Figure 1.8 International comparisons of fleet ages

Figure 1.10 Fleet share of CO2 emissions
Figure 1.11 Population and fleet growth

Composition of the Fleet
Figure 2.1 Number of new/used light vehicles by year
Figure 2.2 Percentage of used imports in the light/truck/bus fleets, by year
Figure 2.3 Average age of Light, Trucks and Buses by year
Figure 2.4 Light fleet average age in detail, by year
Figure 2.5a Light fleet year of manufacture
Figure 2.5b Light passenger fleet year of manufacture in 5 year blocks
Figure 2.5c Light commercial fleet year of manufacture in 5 year blocks
Figure 2.6a Motorcycle year of manufacture
Figure 2.6b Motorcycle year of manufacture in 5 year blocks
Figure 2.7a Truck year of manufacture
Figure 2.7b Truck year of manufacture in 5 year blocks
Figure 2.8a Bus year of manufacture
Figure 2.8b Bus year of manufacture in 5 year blocks
Figure 2.9 Heavy vehicle mass
Figure2.10 Light fleet age distribution
Figure 2.11 Light fleet age structure

Figure 2.13 Light diesel fleet age breakdown

Vehicle age and travel
Figure 3.1 Total LPV, LCV, Truck and Bus travel by year of manufacture in 5 year blocks
Figure 3.2 Light, truck, bus travel by new/used, by year of manufacture in 5 year blocks
Figure 3.2b Travel by new/used passenger light fleet
Figure 3.4 LPV, LCV, Truck and Bus travel per vehicle by year of manufacture in 5 year blocks
Figure 3.5 Average light travel by year of manufacture
Table 3 Light fleet travel by engine size

Engine capacity trends
Figure 4.1a Light fleet average engine capacity by month
Figure 4.1b Light fleet average engine capacity by petrol/diesel by year
Figure 4.2a Light fleet numbers within cc bands, by month
Figure 4.2b Light fleet numbers within cc bands, relative to Jan 2000
Figure 4.3a Light passenger average travel by cc band, by year of manufacture
Figure 4.3b Light commercial average travel by cc band, by year of manufacture
Figure 4.4 Motorcycle fleet engine composition by year
Figure 4.5 Average motorcycle fleet engine capacity by year

Vehicles entering and leaving the fleet
Entering and leaving
Figure 5.1 Entry and exit from the fleet
Figure 5.2abcd Vehicles entering/leaving the fleet by Year of Manufacture
Figure 5.3 Detailed engine size breakdown of vehicles entering and leaving the light fleet

Entering
Figure 6.1 Number of NZ new/used imports entering the light fleet by year
Figure 6.2a Average age of used imports entering the light fleet by year
Figure 6.2b Used light imports : Year of manufacture and fuel
Figure 6.2c Average age of used imports entering the truck and bus fleets by year
Figure 6.3 Average engine size of vehicles entering the light fleet, by petrol/diesel and year
Figure 6.4a Numbers of new and used imports entering the light fleet, by engize size band and year
Figure 6.5a Numbers of motorcycles entering the fleet, by engize size band and year
Figure 6.5b Average engine capacity of motorcycles entering the fleet, by year
Figure 6.7a Country of manufacture of vehicles entering the fleet
Figure 6.7b Country of origin of vehicles entering the fleet
Figure 6.8 NZ new and used imports entering the light fleet by fuel type

Leaving
Figure 7.1a Number of light fleet used imports/NZ new scrapped, by year
Figure 7.1b Number of heavy fleet used imports/NZ new scrapped, by year
Figure 7.2a Average age of light fleet used imports/NZ new when scrapped, by year
Figure 7.2b Average age of vehicles when scrapped, by year and fuel
Figure 7.3a,b,c Last odometer reading of scrapped vehicles
Figure 7.3d,e Last odometer reading of scrapped vehicles by cc band

Petrol/Diesel vehicles and travel, CNG/LPG/electric/Hybrid
Figure 8.1 Diesel vehicles in the light, truck and bus fleets
Figure 8.2 Petrol and diesel travel
Figure 8.2a Percentage of light passenger/commercial vehicles by petrol/diesel
Figure 8.2b Percentage of light passenger/commercial travel by petrol/diesel
Figure 8.3 Light fleet petrol and diesel travel by year of manufacture in 5 year blocks
Table 6 Primary fuel types by vehicle type
Figure 8.4ab LPG, CNG and electric light vehicles

Fuel consumption
Figure 9.0a Real world emissions vs laboratory test results
Figure 9.0b Divergence between real world and test petrol economy
Figure 9.0extra Vehicles registered by month
Figure 9.1 New lights : quarterly CO2 emissions bands
Figure 9.2 Used lights : quarterly CO2 emissions bands
Figure 9.3 Used+new lights : quarterly CO2 emissions bands
Figure 9.4 Average quarterly CO2 emissions of light fleet registrations
Figure 9.5 Aerage fuel economy of NZ new light registrations by cc band and fuel
Figure 9.11 Emissions standards of vehicles in the light fleet

The average light vehicle in use
Figure 10.1 Travel weighted vehicle age by year
Figure 10.2 Travel weighted engine size by year

Road freight
Figure 11.1 Truck and trailer travel
Figure 11.2 Truck+trailer tonne-km

Figure 2: Screenshot for NZ-Vehicle-Fleet-Statistics-2018_web.xlsx contents page.

no data for the motorcycle age distribution and heavy age distribution, mentioned in risk we will further discuss details of data tables in section 2.3

2.3 Date Exploration

- Number of Fleets:
  using Pandas load “Composition of Fleet data” into a data structure (dataframe), Figure 3,4 on terminal while using print () in python

This dataframe contains 34 columns and 19 rows. The first column is named period for indicate the year of the data of following columns, which show as red rectangle in Figure 5.

The Remain 33 columns are number of fleets, average age and percentage of number fleets which generate from number of fleets. There are mainly grouped by light, motorcycle, truck and bus 4 groups, for each group have 2 categories to imported and used imported. The light fleets group has to two subgroups: passenger and commercial.

19 rows are representing the data of different fleets with in different year from 2000 to 2018.

There are four different range or type of data in this data table:
The period data column is in range [2000, 2018] as integers in a red rectangle below.
The number of fleets columns are in large range integers, an example column is in a yellow rectangle.
The average ages of fleets columns are in a small range of floats, an example column is in a blue rectangle.
The percentage of fleets columns are in range [0,1] as floats, an example column is in a green rectangle.

	Period	Total light new	...	Light used average age	NZ new light average age
0	2000	1527641	...	11.052865	12.288986
1	2001	1510448	...	11.386907	12.363645
2	2002	1504464	...	11.639551	12.367758
3	2003	1510494	...	11.837801	12.316135
4	2004	1523241	...	12.104681	12.217946
5	2005	1541884	...	12.422956	12.107238
6	2006	1561044	...	12.833032	12.041413
7	2007	1584733	...	13.221731	11.983797
8	2008	1605137	...	13.689315	11.993982
9	2009	1610696	...	14.225691	12.242148
10	2010	1628515	...	14.657012	12.406411
11	2011	1643564	...	15.095085	12.530054
12	2012	1690422	...	15.514546	12.602192
13	2013	1748586	...	15.780167	12.619205
14	2014	1817743	...	15.893753	12.566808
15	2015	1888590	...	15.974477	12.485732
16	2016	1975136	...	16.106811	12.376743
17	2017	2066267	...	16.192062	12.235900
18	2018	2151898	...	16.423379	12.166780

[19 rows x 34 columns]

Figure 3: Screenshot for number of fleets overview in pandas data frame.

	Bus Used Import	Light used %	Truck used %	Bus used %
0	14.412500	0.387554	0.257613	0.244488
1	14.732752	0.410661	0.274291	0.269294
2	14.856246	0.431699	0.295766	0.313531
3	15.273053	0.452473	0.319948	0.339596
4	15.641497	0.468574	0.346510	0.354958
5	16.060986	0.480230	0.364654	0.369892
6	16.593093	0.484656	0.380547	0.387041
7	16.774632	0.486825	0.393303	0.413788
8	17.179655	0.483566	0.395025	0.426217
9	17.862209	0.480322	0.394553	0.416062
10	18.599363	0.478369	0.393464	0.409976
11	19.412533	0.472736	0.388384	0.399227
12	20.029412	0.465967	0.382446	0.394209
13	20.384393	0.460824	0.375790	0.387371
14	20.916092	0.458805	0.368115	0.379788
15	21.443662	0.457666	0.362077	0.370067
16	21.978696	0.456122	0.356496	0.348863
17	22.411086	0.454848	0.350588	0.332229
18	22.890590	0.451085	0.343582	0.310045

Figure 4: Screenshot for number of fleets overview in pandas data frame.

```
Index(['Period', 'Total light new', 'Total light used import', 'Total LPV new',
       ' Total LPV used', 'Total LCV new', ' Total LCV used', 'Total MC new',
       'Total MC used', 'Total truck new', 'Total truck used', 'Total bus new',
       'Total bus used', 'Light passenger NZ new',
       'Light passenger used import', 'Light commercial NZ New',
       'Light commercial used import', 'Motorcycle NZ New',
       'Motorcycle Used Import', 'Truck NZ New', 'Truck Used Import',
       'Bus NZ New', 'Bus Used Import', 'Light used %', 'Truck used %',
       'Bus used %', 'Light fleet average age', 'Light passenger average age',
       'Light commercial average age', 'Motorcycle average age',
       'Truck fleet average age', 'Bus fleet average age',
       'Light used average age', 'NZ new light average age'],
      dtype='object')
```

Figure 5: Screenshot for number of fleets overview in pandas data frame.

- Age Distribution of Fleets:
  using Pandas load “Light Vehicles age distribution data” into a data structure (dataframe), Figure 6 on terminal while using print () in python

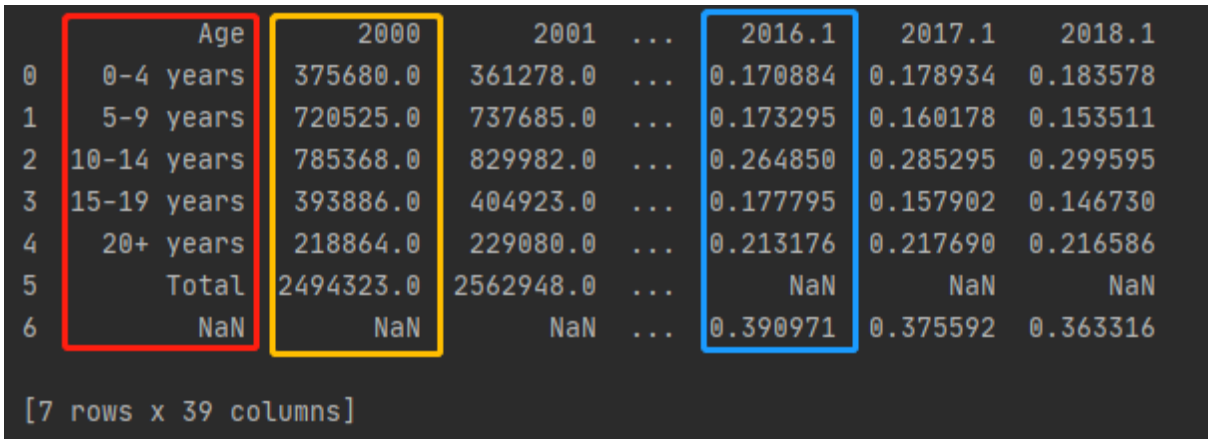
The age distribution of light fleets data table has 7 rows and 39 columns, it is order by different age groups in rows and the columns are the years. This data table storge the

number of fleets in different age groups and the corresponding percentage of them from 2000 to 2018, thus there are 38 columns for then and first columns for the age groups information.

- The Age groups information are in string type as a red rectangle below for show the information for different groups.
- The Yellow rectangle is an example for the fleets of age groups in float type but they are integers.
- The Blue rectangle is an example for the percentages of fleets of corresponding age groups in float type between [0,1]

The percentage of fleets columns have a different start entry, 2000, …, 2018 comparing 2000.1, …, 2018.1. This is cause by when pandas load columns with same entry will rename it. There are some missing values in percentage of fleets for the total groups in 5th row, this is because the total percentage is 1 which is a trivial group in percentage columns. Instead, it using 15+ years group for the total recorded in 6th row, this is also explained why there also have some missing values in 6th row.

For dealing with these problems will be mentioned in step 3.



	Age	2000	2001	...	2016.1	2017.1	2018.1
0	0-4 years	375680.0	361278.0	...	0.170884	0.178934	0.183578
1	5-9 years	720525.0	737685.0	...	0.173295	0.160178	0.153511
2	10-14 years	785368.0	829982.0	...	0.264850	0.285295	0.299595
3	15-19 years	393886.0	404923.0	...	0.177795	0.157902	0.146730
4	20+ years	218864.0	229080.0	...	0.213176	0.217690	0.216586
5	Total	2494323.0	2562948.0	...	NaN	NaN	NaN
6	NaN	NaN	NaN	...	0.390971	0.375592	0.363316

[7 rows x 39 columns]

Figure 6: Screenshot for number of fleets overview in pandas data frame.

- Co2 Emission Data:
  Using Pandas load “CO2 emissions data” into a data structure (dataframe), Figure 6 on terminal while using print () in python.

	Year	Light passenger	Light commercial	Motorcycle	Heavy fleet
0	2001	6.765291	1.504675	0.028235	2.388743
1	2002	7.066370	1.559698	0.027965	2.572510
2	2003	7.375728	1.585027	0.028364	2.639134
3	2004	7.671046	1.582260	0.029396	2.653599
4	2005	7.549507	1.628644	0.032007	2.860993
5	2006	7.644814	1.663934	0.037437	2.919942
6	2007	7.796914	1.718954	0.040616	3.009903
7	2008	7.692528	1.764464	0.045334	3.077349
8	2009	7.626856	1.774602	0.046301	2.990650
9	2010	7.697162	1.830283	0.046434	3.109751
10	2011	7.575258	1.861729	0.045124	3.200497
11	2012	7.444775	1.879310	0.044464	3.213818
12	2013	7.415187	1.937471	0.045089	3.288810
13	2014	7.430754	1.993417	0.045306	3.345710
14	2015	7.662124	2.121840	0.046825	3.454175
15	2016	7.800260	2.237802	0.047416	3.509366
16	2017	8.077407	2.472227	0.047200	3.861072

Figure 7: Screenshot for number of fleets overview in pandas data frame.

This file contains 17 rows and 5 columns, 4 columns for Light passenger, Light commercial, Motorcycle and Heavy fleet from 2001 to 2017.

- Red rectangle shows the Year data which is similar with Period in the number of fleet data table with different column entry.
- Yellow rectangle is an example of the Co2 emission of 4 groups in float numbers, because the Million Tonnes CO2.

There 2 rows of missing data founded in this file, it does not contain CO2 emission data in the year 2000 and 2018.

However, the differences with two files above are this file has no totally light fleets group and no bus group, these rows need to imputation in step 3.

2.3.1 Exploration of most important predictors (features)

Hypothesis among the used vehicles will contribute more Co2 emission compre with same age range of new vehicles.

There is a relationship between number, age and Co2 emssion in fleets.

Assumpution: The age of fleets will be a significate feature, base on new vehicles will have apply harder emission standard.

The heavy vehicle and motorcycle data cannot split by age because The NZ Transport have not statistics the age distribution of them. However, they have taken 3.65% and 3.69% of

totally vehicle, count by mean vehicle number/meanTotal numbe, over 18 years, so that dismiss these two types of fleets.

2.4 Data quality

2.4.1 Missing values imputation

There are two missing rows can find in CO2 emission data, which will impute by mean imputation. Scikit-learn provide mean imputation.

- create two missing rows for year 2000 and year 2018 with np.nan.

```
[[2000, nan, nan, nan, nan], [2018, nan, nan, nan, nan]]
```

- create scikit-learn mean imputation object fit for Co2 emission data and impute the missing rows.

```
imp = SimpleImputer(missing_values=np.nan, strategy='mean')
imp.fit(co2_emission_df)
X = imp.transform(missing_values)
```

```
[[2.00000000e+03 7.54658706e+00 1.83037287e+00 4.02067179e-02
  3.06447186e+00]
 [2.01800000e+03 7.54658706e+00 1.83037287e+00 4.02067179e-02
  3.06447186e+00]]
```

- Adding these missing rows to the C
- Co2 emission data frame.

2.4.2 Check features normality

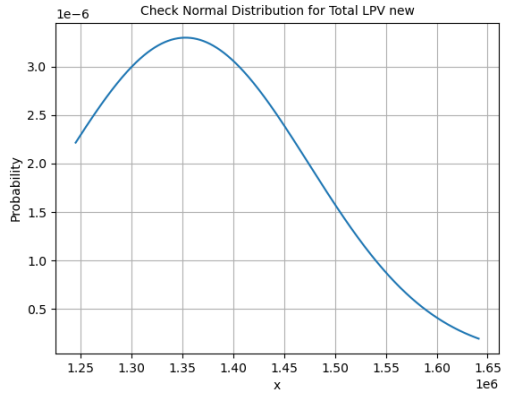
	Year	Light passenger	Light commercial	Motorcycle	Heavy fleet
0	2000.0	7.546587	1.830373	0.040207	3.064472
1	2001.0	6.765291	1.504675	0.028235	2.388743
2	2002.0	7.066370	1.559698	0.027965	2.572510
3	2003.0	7.375728	1.585027	0.028364	2.639134
4	2004.0	7.671046	1.582260	0.029396	2.653599
5	2005.0	7.549507	1.628644	0.032007	2.860993
6	2006.0	7.644814	1.663934	0.037437	2.919942
7	2007.0	7.796914	1.718954	0.040616	3.009903
8	2008.0	7.692528	1.764464	0.045334	3.077349
9	2009.0	7.626856	1.774602	0.046301	2.990650
10	2010.0	7.697162	1.830283	0.046434	3.109751
11	2011.0	7.575258	1.861729	0.045124	3.200497
12	2012.0	7.444775	1.879310	0.044464	3.213818
13	2013.0	7.415187	1.937471	0.045089	3.288810
14	2014.0	7.430754	1.993417	0.045306	3.345710
15	2015.0	7.662124	2.121840	0.046825	3.454175
16	2016.0	7.800260	2.237802	0.047416	3.509366
17	2017.0	8.077407	2.472227	0.047200	3.861072
18	2018.0	7.546587	1.830373	0.040207	3.064472

Figure 8: added information.

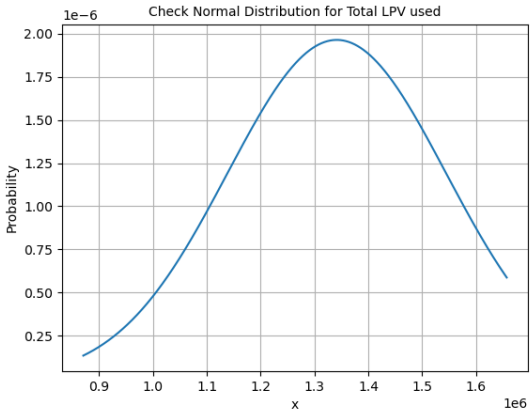
According to the CO2 emission data, check data normality within these files. If the fleet's groups follow the normal distribution or skew normal distribution then the average age of these groups must follow the same distribution, because they are from the same original fleets data.

Mean and standard deviation able to get from data, then build pdf for the mean and std dev to plot the figures for check the normality.

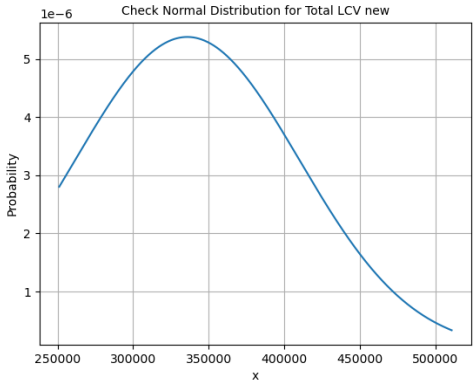
Total LPV new:



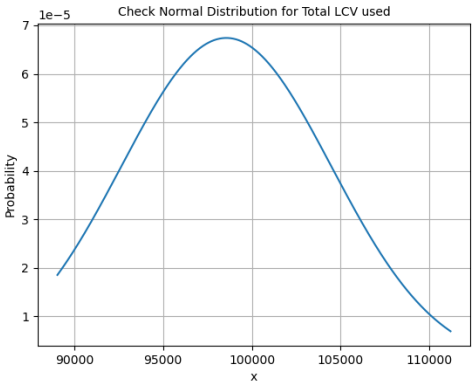
Total LPV used:



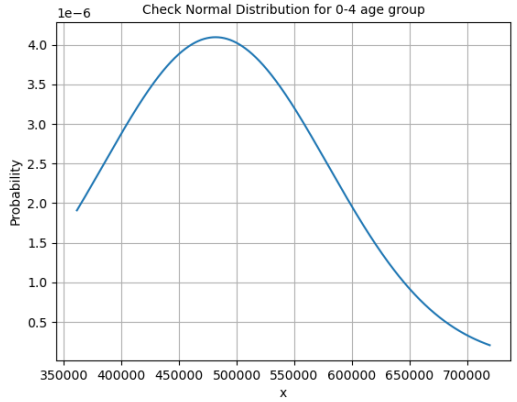
Total LCV new:



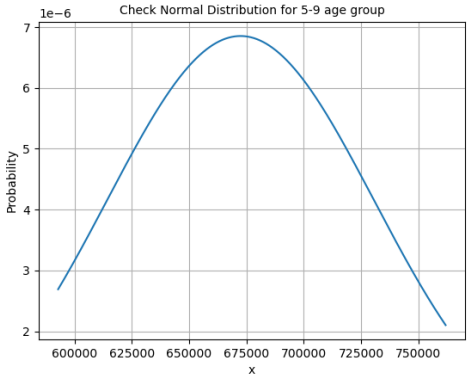
Total LCV used:



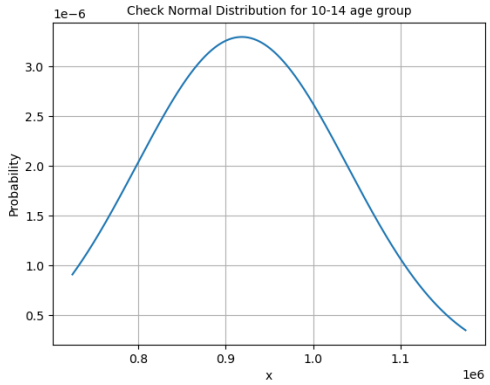
0-4 age group:



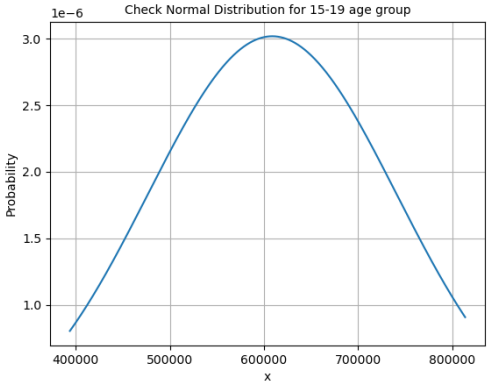
5-9 age group



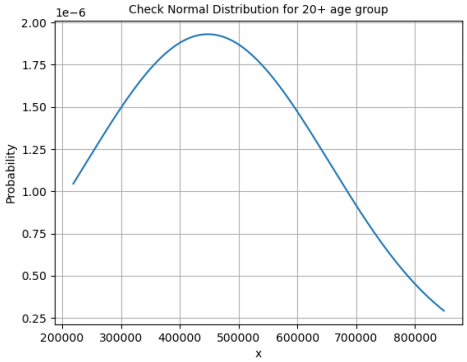
10-14 age group:



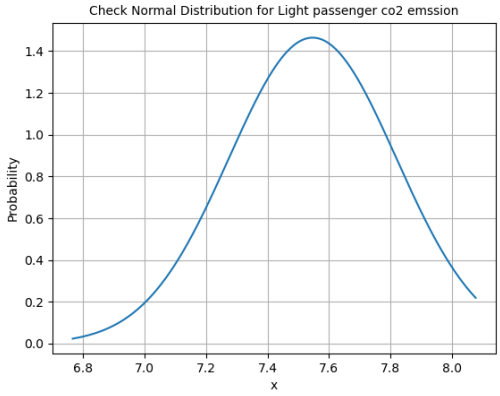
15-19 age group:



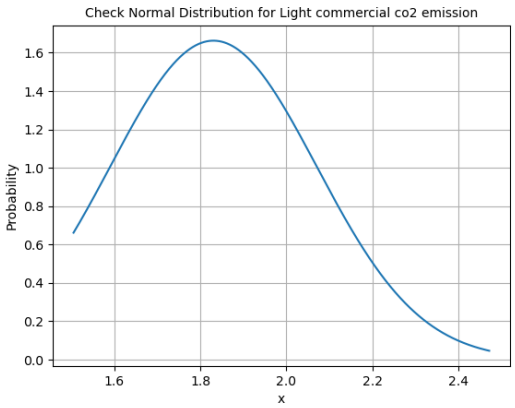
20+ age group:



Light passenger group:



Light commercial group:



All the groups are following the normal and skew normal distribution.

3. Data preparation

3.1 Data Selection

Considering the business and datamining objectives is around real emission values and model predict emission values. In the Data Selection part will start from CO2 emission data, and there are no age distributions data for the motor cycle fleets and heavy fleets, and we mentioned above in step 2.3.1 that we discard these two groups data. Hence, select all the columns related to Light Passenger group and Light commercial group.

Number of Fleets:

There are 15 columns related to LPV and LCV in number of fleets data tables, marked with Figure 9 red rectangle below:

```
Index(['Period', 'Total light new', 'Total light used import', 'Total LPV new',
       ' Total LPV used', 'Total LCV new', ' Total LCV used', 'Total MC new',
       'Total MC used', 'Total truck new', 'Total truck used', 'Total bus new',
       'Total bus used', 'Light passenger NZ new',
       'Light passenger used import', 'Light commercial NZ New',
       'Light commercial used import', 'Motorcycle NZ New',
       'Motorcycle Used Import', 'Truck NZ New', 'Truck Used Import',
       'Bus NZ New', 'Bus Used Import', 'Light used %', 'Truck used %',
       'Bus used %', 'Light fleet average age', 'Light passenger average age',
       'Light commercial average age', 'Motorcycle average age',
       'Truck fleet average age', 'Bus fleet average age',
       'Light used average age', 'NZ new light average age',
      dtype='object')
```

Figure 9:

Light age distribution:

All the light age distribution columns are needed; however, the age distribution data table need to transform for coincide with others.

- Construct data frames for numbers and percentages within different age groups
- Drop unnecessary columns which is using presentation better in Excel
- Reset index and columns appropriately.

New data frame has 13 columns 19 rows, 6 columns for numbers and 6 for columns from 2000 to 2018.

```
Index(['Period', '0-4 years', '5-9 years', '10-14 years', '15-19 years',
       '20+ years', 'Total', '0-4 years percentage', '5-9 years  percentage',
       '10-14 years percentage', '15-19 years percentage',
       '20+ years percentage', '15+ years percentage'],
      dtype='object')
```

Figure 10:

	Period	0-4 years	5-9 years	10-14 years	15-19 years	20+ years
0	2000	375680	720525	785368	393886	218864
1	2001	361278	737685	829982	404923	229080
2	2002	371546	730168	887866	419316	238400
3	2003	397164	760055	907729	446613	247192
4	2004	422522	761918	926784	488853	266249
5	2005	451885	756408	938797	536089	283289
6	2006	475895	661033	1.02321e+06	572331	296656
7	2007	498470	609594	1.0527e+06	616195	311134
8	2008	503736	592814	1.05052e+06	627768	333272
9	2009	471609	594269	995752	661093	376686
10	2010	448838	630819	926398	687235	428672
11	2011	425980	649268	809535	768191	464176
12	2012	423475	663850	752875	808744	516439
13	2013	443043	704420	724437	813648	557519
14	2014	502841	698847	765817	773998	617248
15	2015	559050	665914	854452	728910	674011
16	2016	620579	629335	961822	645677	774166
17	2017	678208	607116	1.08134e+06	598489	825103
18	2018	719677	601804	1174496	575220	849077

Figure 11:

	0-4 years percentage	5-9 years percentage	10-14 years percentage
0	0.150614	0.288866	0.314862
1	0.140962	0.287827	0.323839
2	0.140349	0.275817	0.335386
3	0.143965	0.275507	0.329036
4	0.147409	0.265817	0.323335
5	0.152331	0.254986	0.31647
6	0.157106	0.218226	0.337791
7	0.161417	0.197402	0.340889
8	0.162071	0.190731	0.337994
9	0.152161	0.191736	0.321272
10	0.143768	0.202059	0.296736
11	0.136657	0.208289	0.259704
12	0.133783	0.209722	0.237846
13	0.136612	0.217208	0.22338
14	0.149711	0.208068	0.228006
15	0.160539	0.191226	0.245367
16	0.170884	0.173295	0.26485
17	0.178934	0.160178	0.285295
18	0.183578	0.153511	0.299595

Figure 12:

Needs to transpose first then divided into two groups of number of age groups and percentages of age groups.

Co2 Emission:

Co2 emission data are using for target, mentioned in step2.2, thus only Light passenger and Light commercial columns are select for corresponding output:

```
Index(['Year', 'Light passenger', 'Light commercial', 'Motorcycle',
       'Heavy fleet'],
      dtype='object')
```

Figure 13:

Hence, one risk meet here that is we do not have enough age group data for heavy and motorcycle fleets. In step 1.2, the contingency method is using the majority groups data which are the LPV and LCV to measure the whole population.

3.2 Data Cleaning

The cleaning includes two aspects:

1. Cleaning for the null values, in step 3.1, we have cleaned the nan (not a number) values in light fleets age distribution table which we mentioned in step 2.3, because these nan values are just null values for make the excel looks more convenient.
2. Excluding the columns which are redundant columns. There are some fields are redundant for Light fleets groups. For example, the “Total Light New” field are including the number of new fleets both for Light passenger and Light commercial, thus we can clean some cross-hold information groups between Light Passenger and Light Commercial. The project needs columns direct have relationship with LPV and LCV, because they are the target output values columns.

- For number of fleets data frame, we select 5 columns and marked with yellow rectangle which distinct with red rectangle above.

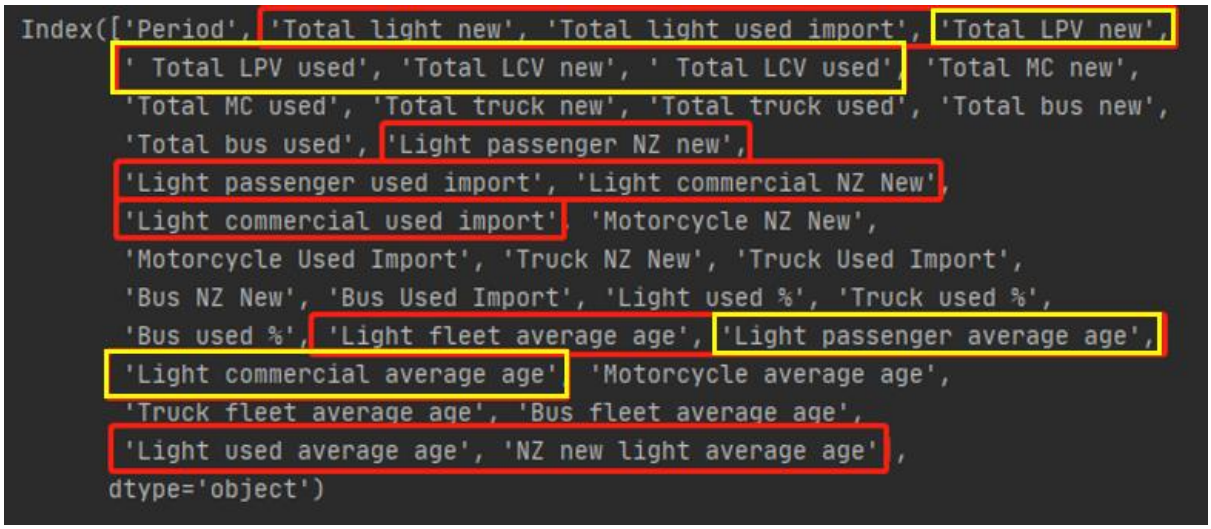


Figure 14:

- For light fleets age distribution data frame, we clean the null values and next step is remaining all the percentage columns for construct new features that we will explained in 3.3.

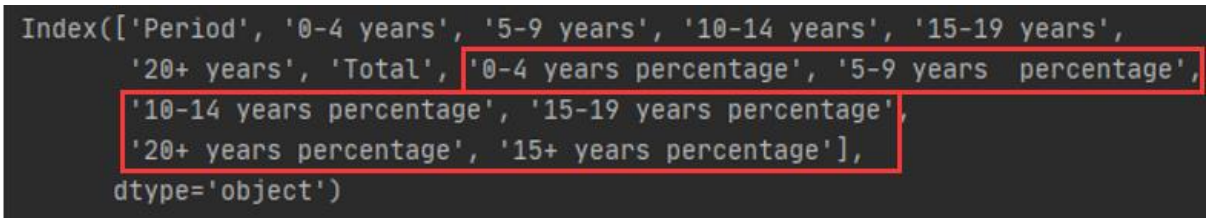


Figure 15:

- **For Co2 emission data frame will same as 3.1, because they are target values.**

Hence, after step 3.2, there are 12 columns for inputs and 2 columns for the target output. Storage these columns in lists then in the following steps could select these columns using these lists, by using df[columns].

```
# ---- step 3.2 ----
num_fleets_select_cols = ['Period', 'Total LPV new', ' Total LPV used', 'Total LCV new', ' Total LCV used',
                          'Light passenger average age', 'Light commercial average age']

percentage_columns = ['0-4 years percentage', '5-9 years  percentage','10-14 years percentage',
                      '15-19 years percentage', '20+ years percentage', '15+ years percentage']
```

Figure 16:

3.3 Data Construct

After we have cleaned the redundant fields, the next step for Light Passenger and Light Commercial is construct new features with “Light Fleet age distribution data”.

At first, the reason why remains the percentage of age groups in step 3.2 is, instead using percentage and the number of fleets for age groups are combine with Light Passenger and Light Commercial groups that is we cannot know how many vehicles exactly in an age range belongs to which groups (LPV or LCV). However, we can estimate the number of vehicles in different age ranges for different groups by construct new features with fields g to l in “Light Fleet age distribution data” which are the percentage for age ranges, if the Light Passenger and Light Commercial groups are following normal distribution.

Next step is applying the percentage of age groups to the number of Light Passenger and Light Commercial to get the number of age groups in LPV new, LPV used, LCV new or LCV used. There are 24 columns which generated by 6 age groups * 4 number fleets group and 1 column for the Period. i.e. Elements in yellow rectangle shown below times blue rectangle above.

```
# ---- step 3.3 ----
nums_columns = ['Total LPV new', ' Total LPV used', 'Total LCV new', ' Total LCV used']
new_age_distribution_df = step_3_3_construct_new_distribution_df(nums_columns, percentage_columns,
                                                                 number_fleets_df, new_age_distribution_df)
```

Figure 17:

```
new_table = {}
for num_column in nums_columns:
    for percenate_column in percentage_columns:
        new_column = new_age_distribution_df[percenate_column] * number_fleets_df[num_column]
        new_column_name = '%s of %s' % (percenate_column[:-11].strip(), num_column[6:].strip())
        new_table[new_column_name] = new_column

new_age_distribution_df = pd.DataFrame(new_table)

return new_age_distribution_df
```

Figure 18:

	Period	0-4 years of LPV new	...	20+ years of LCV used	15+ years of LCV used
0	2000	192258	...	8424	23584
1	2001	177089	...	8543	23646
2	2002	174847	...	8707	24023
3	2003	179270	...	8850	24840
4	2004	184137	...	9449	26798
5	2005	191531	...	9903	28645
6	2006	199128	...	10201	29883
7	2007	206736	...	10583	31543
8	2008	209486	...	11027	31798
9	2009	197036	...	12037	33164
10	2010	187903	...	13087	34068
11	2011	179816	...	13589	36079
12	2012	180668	...	14529	37281
13	2013	189837	...	15394	37861
14	2014	214794	...	16872	38030
15	2015	237588	...	18321	38135
16	2016	262336	...	21297	39060
17	2017	284522	...	23155	39950
18	2018	301294	...	24081	40395

[19 rows x 25 columns]

Figure 19:

3.4 Data Integration

There are current 3 data frames:

1. Number of fleets which has 6 columns after data cleaning
2. New constructed age distribution has 24 columns which we constructed in step 3.3
3. Co2 emission data has 2 columns for LPV and LCV.

All data frames are using for the models then we merge these two data frames into one. Hence, we got one data frame for input values with 30 columns and 1 period column, last 2columns for co2 emission target values, explained in red rectangle below.

Examples:

- Period: Yellow Rectangle
- 30 Columns for features: Blue Rectangle
- 2 Columns for targets: Green Rectangle

	Period	Total LPV new	...	Light passenger	Light commercial
0	2000	1276498	...	7.546587	1.830373
1	2001	1256293	...	6.765291	1.504675
2	2002	1245805	...	7.066370	1.559698
3	2003	1245233	...	7.375728	1.585027
4	2004	1249158	...	7.671046	1.582260
5	2005	1257339	...	7.549507	1.628644
6	2006	1267476	...	7.644814	1.663934
7	2007	1280762	...	7.796914	1.718954
8	2008	1292558	...	7.692528	1.764464
9	2009	1294924	...	7.626856	1.774602
10	2010	1306995	...	7.697162	1.830283
11	2011	1315826	...	7.575258	1.861729
12	2012	1350457	...	7.444775	1.879310
13	2013	1389608	...	7.415187	1.937471
14	2014	1434731	...	7.430754	1.993417
15	2015	1479943	...	7.662124	2.121840
16	2016	1535175	...	7.800260	2.237802
17	2017	1590094	...	8.077407	2.472227
18	2018	1641232	...	7.546587	1.830373

[19 rows x 33 columns]

Figure 20:

3.5 Data Formatting

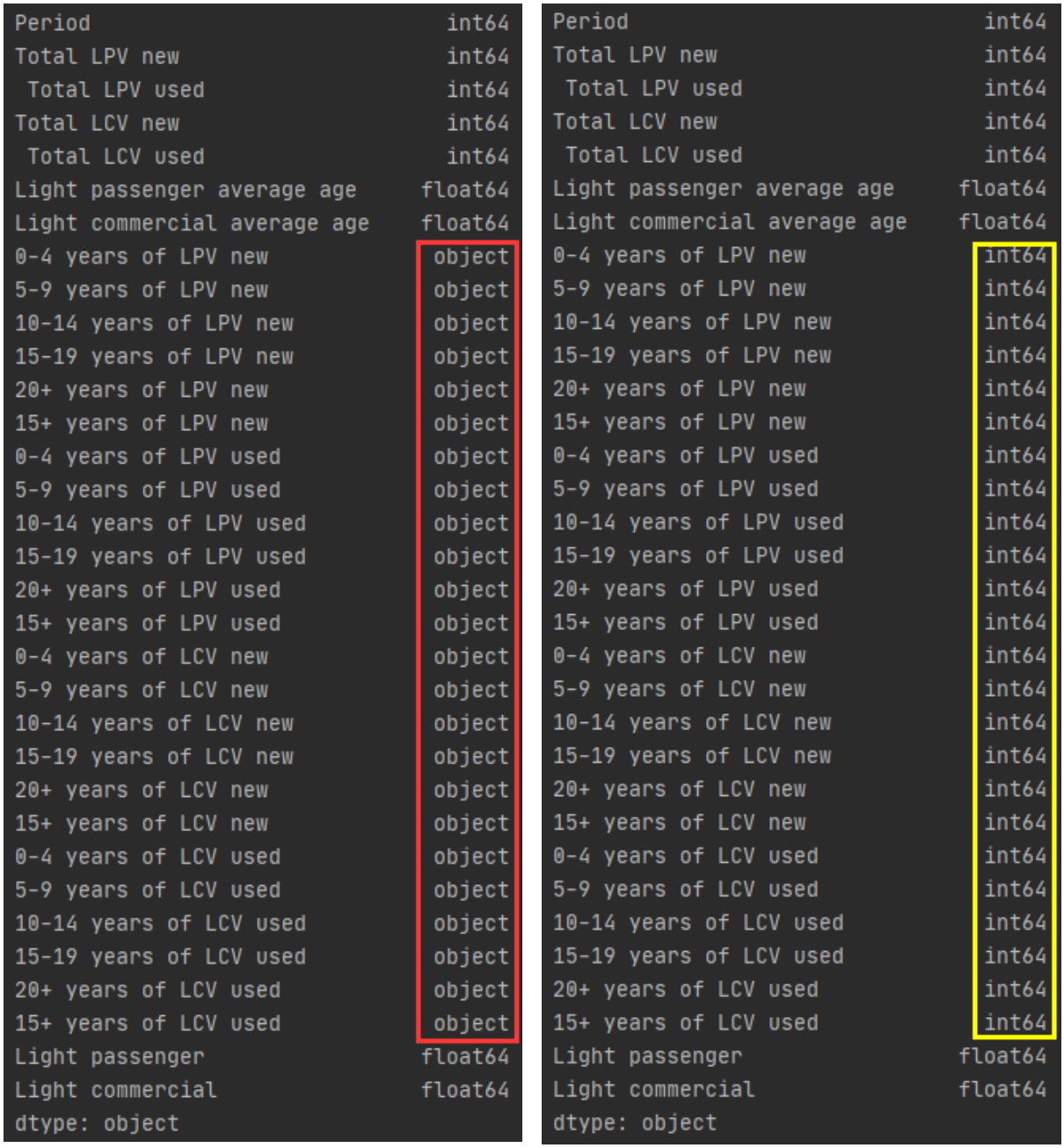
Since the new construct data is object type shown as figure below which actually is float, in this step we need to convert to integer, because there are not exist that parts of vehicles could run on the road.

Convert object type shown in rectangle below to int64 type in yellow below:
age and Co2 emission columns already in float64 type, others should be in int64 type, only new constructed columns are in object type thus it will only change type for these columns.

```
def step_3_5_convert_object_to_int(data_df):
    for column in data_df.columns:
        if data_df.dtypes[column] != np.float64:
            data_df[column] = data_df[column].astype(np.int64)

    return data_df
```

Figure 21:



Period	int64
Total LPV new	int64
Total LPV used	int64
Total LCV new	int64
Total LCV used	int64
Light passenger average age	float64
Light commercial average age	float64
0-4 years of LPV new	object
5-9 years of LPV new	object
10-14 years of LPV new	object
15-19 years of LPV new	object
20+ years of LPV new	object
15+ years of LPV new	object
0-4 years of LPV used	object
5-9 years of LPV used	object
10-14 years of LPV used	object
15-19 years of LPV used	object
20+ years of LPV used	object
15+ years of LPV used	object
0-4 years of LCV new	object
5-9 years of LCV new	object
10-14 years of LCV new	object
15-19 years of LCV new	object
20+ years of LCV new	object
15+ years of LCV new	object
0-4 years of LCV used	object
5-9 years of LCV used	object
10-14 years of LCV used	object
15-19 years of LCV used	object
20+ years of LCV used	object
15+ years of LCV used	object
Light passenger	float64
Light commercial	float64
dtype: object	

Period	int64
Total LPV new	int64
Total LPV used	int64
Total LCV new	int64
Total LCV used	int64
Light passenger average age	float64
Light commercial average age	float64
0-4 years of LPV new	int64
5-9 years of LPV new	int64
10-14 years of LPV new	int64
15-19 years of LPV new	int64
20+ years of LPV new	int64
15+ years of LPV new	int64
0-4 years of LPV used	int64
5-9 years of LPV used	int64
10-14 years of LPV used	int64
15-19 years of LPV used	int64
20+ years of LPV used	int64
15+ years of LPV used	int64
0-4 years of LCV new	int64
5-9 years of LCV new	int64
10-14 years of LCV new	int64
15-19 years of LCV new	int64
20+ years of LCV new	int64
15+ years of LCV new	int64
0-4 years of LCV used	int64
5-9 years of LCV used	int64
10-14 years of LCV used	int64
15-19 years of LCV used	int64
20+ years of LCV used	int64
15+ years of LCV used	int64
Light passenger	float64
Light commercial	float64
dtype: object	

Figure 22:

4.1 Data Reduction

Although, most of features are needed for our project success criteria, we go through on Principle Component Analysis (PCA) for find are there any features could be reduced.

we set all features (n) go through PCA, thus there will be return n floats values (ratios) which sum equal to 1, it will discuss how many variances that the n features under this PCA component can explained. In this case, we are aim for the components can explained over 90% variances.

The PCA components have n floats for n features, and how the bigger of float is then how the feature important is. We run PCA separately on LPV_input and LCV_input data frames,

and rank importance of all the features among the components which explained over 90% variances.

For LPV_input:
Component explained variance ration:

```
[8.15390324e-01 9.41978448e-02 8.17119244e-02 7.50059559e-03
 8.88775169e-04 2.98629476e-04 7.93352645e-06 2.14514515e-06
 1.68783289e-06 1.40183256e-07 6.51477294e-13 4.36104833e-13
 1.36339300e-13 1.06180843e-13 2.28458149e-14 2.16830936e-16]
```

The first 2 components explained 0.9095881686967918 variance; thus, we will rank the features importance for first 2 components.

First 2 components:

```
[[-1.72400829e-05 -3.32328525e-01 -6.17384026e-01 -2.98352232e-06
  -8.72487626e-02  9.69261430e-02  6.11808248e-03 -1.02857621e-01
  -2.45265666e-01 -3.48123730e-01 -1.29053217e-01  1.79344876e-02
  -8.38773119e-02 -1.48761857e-01 -2.73625879e-01 -4.22388682e-01]
 [ 5.14555414e-06  5.77236642e-02 -4.48752542e-01  2.65330341e-06
  -9.84292324e-02  1.39055422e-01 -3.62034556e-01  2.43986527e-01
   1.35142869e-01  3.79131164e-01 -1.77394300e-01  9.80370954e-03
  -5.33210681e-01  1.61312178e-01  9.07354028e-02  2.52049435e-01]]
```

Rank the importance for components in ascending:

```
[2, 15, 9, 1, 14, 8, 13, 10, 7, 4, 12, 0, 3, 6, 11, 5]
[12, 2, 6, 10, 4, 3, 0, 11, 1, 14, 8, 5, 13, 7, 15, 9]
```

Mark the top 4 unnecessary columns with rectangels:

```
LPV_cols = ['Period', 'Total LPV new', 'Total LPV used', 'Light passenger average age',
            '0-4 years of LPV new', '5-9 years of LPV new', '10-14 years of LPV new',
            '15-19 years of LPV new', '20+ years of LPV new', '15+ years of LPV new',
            '0-4 years of LPV used', '5-9 years of LPV used', '10-14 years of LPV used',
            '15-19 years of LPV used', '20+ years of LPV used', '15+ years of LPV used', ]
```

```
LPV_cols = ['Period', 'Total LPV new', 'Total LPV used', 'Light passenger average age',
            '0-4 years of LPV new', '5-9 years of LPV new', '10-14 years of LPV new',
            '15-19 years of LPV new', '20+ years of LPV new', '15+ years of LPV new',
            '0-4 years of LPV used', '5-9 years of LPV used', '10-14 years of LPV used',
            '15-19 years of LPV used', '20+ years of LPV used', '15+ years of LPV used', ]
```

Explanation of ranks, the first rank takes the most important ratio around 0.815, thus the reduction majority base on the first component rank, it says the most unnecessary features are the 15+ years of LPV groups, and these two features are combining from 15-19 years and 20+ years groups, thus we can reduce these two features.

The second rank is more focus on the 10-14-year group, it causes these groups do not have large variance than others in this component analysis, we cannot satisfy project requirement if we ignore these.

For LCV_input:

We ignore 15+ year group first for coincide with LPV result:

Component explained variance ration:

```
[9.44778670e-01 4.33235385e-02 1.04526159e-02 1.02785875e-03
 3.02791805e-04 1.00328575e-04 1.22603900e-05 1.63474506e-06
 1.63803424e-07 1.37771791e-07 7.17203231e-12 4.10794208e-12
 6.39920456e-13 1.36431569e-14]
```

The first component explained 0.94477867 variance; thus, we will rank the features importance for first components.

First component:

```
[[ 6.19398749e-05  8.88799538e-01  1.94828197e-02  2.29525335e-07
   1.87205271e-01  3.56333289e-02  1.80884277e-01  1.49093445e-01
   3.35983454e-01  1.51997423e-02 -3.88344527e-02 -1.84863626e-02
   3.15306880e-03  5.84513186e-02]
```

Rank the importance for components in ascending:

```
[10, 11, 3, 0, 12, 9, 2, 5, 13, 7, 6, 4, 8, 1]
```

Mark the top 4 unnecessary columns with red rectangles:

```
LCV_cols = ['Period',  'Total LCV new', ' Total LCV used', 'Light commercial average age'
            '0-4 years of LCV new', '5-9 years of LCV new', '10-14 years of LCV new',
            '15-19 years of LCV new', '20+ years of LCV new',
            '0-4 years of LCV used', '5-9 years of LCV used' '10-14 years of LCV used',
            '15-19 years of LCV used', '20+ years of LCV used', ]
```

As shown on the figure we know that Period is not a relevant feature and it will not attend model predict, thus we can ignore it. Light commercial average age has a lower rank because it has a low variance, that means the average of light passenger not have a large variation.

5-9 years and 10-14 of LCV used we cannot ignore it; it is important component for this project success.

To sum up, we reduce features related to 15+ year groups both for LPV and LCV inputs and they have explained in data because it is combining with 15-19 years groups and 20+ years groups. we add the target column to the return list for the next steps generate LPV and LCV data frame.

```
reduced_LPV_cols = ['Total LPV new', ' Total LPV used', 'Light passenger average age',
                    '0-4 years of LPV new', '5-9 years of LPV new', '10-14 years of LPV new',
                    '15-19 years of LPV new', '20+ years of LPV new',
                    '0-4 years of LPV used', '5-9 years of LPV used', '10-14 years of LPV used',
                    '15-19 years of LPV used', '20+ years of LPV used', 'Light passenger']
```

```
reduced_LCV_cols = ['Total LCV new', ' Total LCV used', 'Light commercial average age',
                    '0-4 years of LCV new', '5-9 years of LCV new', '10-14 years of LCV new',
                    '15-19 years of LCV new', '20+ years of LCV new',
                    '0-4 years of LCV used', '5-9 years of LCV used', '10-14 years of LCV used',
                    '15-19 years of LCV used', '20+ years of LCV used', 'Light commercial']
```

Figure 23:

4.2 Data Transformation/Normalization

In Data Normalization, because our data set columns have different ranges. for example, for the Total LPV used column range [870690,1657190] and Co2 emission range [6.76, 7.54]. As shown in the different colour rectangles below. If not apply log transformation, this might influence parameters like (β_i) in linear regression model $y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i \;\; i = 1, \ldots, n.$

For avoid overfitting in next steps of machine learning models this step we will do normalization under L2 normalization.

$$L' = L + \frac{\lambda}{2n} \sum_{w} w^2 \quad w \leftarrow \left(1 - \frac{\eta\lambda}{n}\right) w - \eta \frac{\partial L}{\partial w}$$

The above equations are explained the L2 normalization will consider all the feature ranges, it will prevent weights from going to large, and transform the values for all data in range (0,1), but not equal to 0 or 1.

LPV before l2 normalization:

	Total LPV new	Total LPV used	...	20+ years of LPV used	Light passenger
0	1276498	870680	...	76397	7.546587059314
1	1256293	956915	...	85530	6.765290559200
2	1245805	1046144	...	94209	7.066370322830
3	1245233	1149489	...	102997	7.375727557740
4	1249158	1241364	...	115308	7.671045880500
5	1257339	1320880	...	126140	7.549506885850
6	1267476	1363917	...	133574	7.644814489260
7	1280762	1398318	...	140884	7.796914139070
8	1292558	1400142	...	150132	7.692528220200
9	1294924	1389668	...	168892	7.626856430540
10	1306995	1398139	...	191976	7.697161857650
11	1315826	1382331	...	205843	7.575257564360
12	1350457	1385912	...	226114	7.444775230660
13	1389608	1404935	...	241523	7.415187007980
14	1434731	1449199	...	266323	7.430753561430
15	1479943	1499088	...	290150	7.662123754430
16	1535175	1556538	...	331816	7.800259903390
17	1590094	1617627	...	352141	8.077406643250
18	1641232	1657190	...	358924	7.546587059314

[19 rows x 14 columns]

LPV after l2 normalization:

	Total LPV new	Total LPV used	...	20+ years of LPV used	Light passenger
0	0.742510709923	0.506455337114	...	0.044438448557	0.000004389683
1	0.714235993429	0.544031635655	...	0.048626080475	0.000003846248
2	0.687310716569	0.577157727152	...	0.051975112716	0.000003898517
3	0.660092946235	0.609339441433	...	0.054598290588	0.000003909843
4	0.638363673527	0.634380665396	...	0.058926443626	0.000003920174
5	0.621636080669	0.653051139139	...	0.062364386387	0.000003732522
6	0.613374294298	0.660045339996	...	0.064640954138	0.000003699583
7	0.608718792218	0.664590645332	...	0.066959152694	0.000003705707
8	0.611819784687	0.662743626956	...	0.071063525130	0.000003641184
9	0.616349577400	0.661445215724	...	0.080388125346	0.000003630182
10	0.619061376415	0.662231954797	...	0.090929901644	0.000003645780
11	0.626165715044	0.657813631166	...	0.097955071021	0.000003604859
12	0.634350720120	0.651005011802	...	0.106212621897	0.000003497037
13	0.639754966500	0.646811290565	...	0.111193616310	0.000003413842
14	0.641032021600	0.647496265621	...	0.118992041775	0.000003320031
15	0.640317289155	0.648600631487	...	0.125537308834	0.000003315121
16	0.639310558085	0.648206997547	...	0.138181948078	0.000003248352
17	0.636917702684	0.647946141951	...	0.141051306867	0.000003235433
18	0.638267727303	0.644473721576	...	0.139583925828	0.000002934834

[19 rows x 14 columns]

LCV before l2 normalization:

	Total LCV new	Total LCV used	...	20+ years of LCV used	Light commercial
0	251143	96007	...	8424	1.830372872481
1	254155	95590	...	8543	1.504674907110
2	258659	96693	...	8707	1.559697576910
3	265261	98774	...	8850	1.585027089610
4	274083	101725	...	9449	1.582260375660
5	284545	103708	...	9903	1.628644350860
6	293568	104168	...	10201	1.663934474560
7	303971	105041	...	10583	1.718953674540
8	312579	102839	...	11027	1.764464217150
9	315772	99049	...	12037	1.774602036010
10	321520	95312	...	13087	1.830283394650
11	327738	91259	...	13589	1.861729449030
12	339965	89053	...	14529	1.879310423130
13	358978	89550	...	15394	1.937471009460
14	383012	91813	...	16872	1.993416827040
15	408647	94661	...	18321	2.121840162330
16	439961	99907	...	21297	2.237801832490
17	476173	106367	...	23155	2.472227031640
18	510666	111186	...	24081	1.830372872481

[19 rows x 14 columns]

LCV after l2 normalization:

	Total LCV new	Total LCV used	...	20+ years of LCV used	Light commercial
0	0.839536134866	0.320938054017	...	0.028160260888	0.000006118682
1	0.840361229869	0.316067478362	...	0.028247352941	0.000004975194
2	0.840681605376	0.314267148905	...	0.028299091615	0.000005069257
3	0.841871302266	0.313483685917	...	0.028087660927	0.000005030475
4	0.843734173313	0.313149151097	...	0.029087700454	0.000004870813
5	0.847104711779	0.308743908518	...	0.029481726830	0.000004848556
6	0.849175069444	0.301316453543	...	0.029507422074	0.000004813098
7	0.851808104817	0.294352997944	...	0.029656398713	0.000004816968
8	0.856793521130	0.281886463644	...	0.030225517893	0.000004836478
9	0.862657711673	0.270592021090	...	0.032883887347	0.000004848036
10	0.869141620419	0.257649994169	...	0.035377134817	0.000004947672
11	0.874902908229	0.243617659539	...	0.036276097431	0.000004969923
12	0.879294390315	0.230329014283	...	0.037578186569	0.000004860698
13	0.882703900402	0.220197712063	...	0.037852859626	0.000004764117
14	0.886037771130	0.212394875045	...	0.039030707326	0.000004611455
15	0.887910979958	0.205680064392	...	0.039807993363	0.000004610349
16	0.887831477644	0.201610095979	...	0.042976870630	0.000004515834
17	0.886716767748	0.198073814423	...	0.043118628644	0.000004603716
18	0.886282249827	0.192967963854	...	0.041793584962	0.000003176689

[19 rows x 14 columns]

5 Data Ming Objectives

5.1 Match Objective of datamining

This project is a supervised learning problem, because its data have labels and if data does not have target value it will be an un-supervised learning problem. Since it is a supervised learning problem, it will be either a classification problem or regression problem.

There are two datamining objectives mentioned in step 1.3, the first objective is for predict Co2 emission values for fleets and it is not is not predicting labels for flees, thus it is a regression problem. The second objective is basing on the models created for the first object to divide the Co2 emission for age groups among fleets, for second objective, needs to running the models several times and base on the benchmarks mentioned in step 1.3.1.2.

5.1.1 Data type accessible for data mining

Since it is a regression model which is means the target predict value have relation with all feature values, it needs all the data should be numeric and better not equal to 0 or 1 which is same as Boolean values. If some values are equal to 0 it might make some features useless, because whatever 0 relation to other features will be useless. If some values are equal to 2 then it might cause this feature become useless, because every thing relation to1 will be themselves. Hence, in step 3.5 we know that there are all numeric features.

For the model training aspect, good data will cause a more reliable result. Data transformation able to reduce the chance of overfitting, our data have a large range for different columns and we have done data transformation at step 4.2.

5.1.2 Datamining goals/objectives

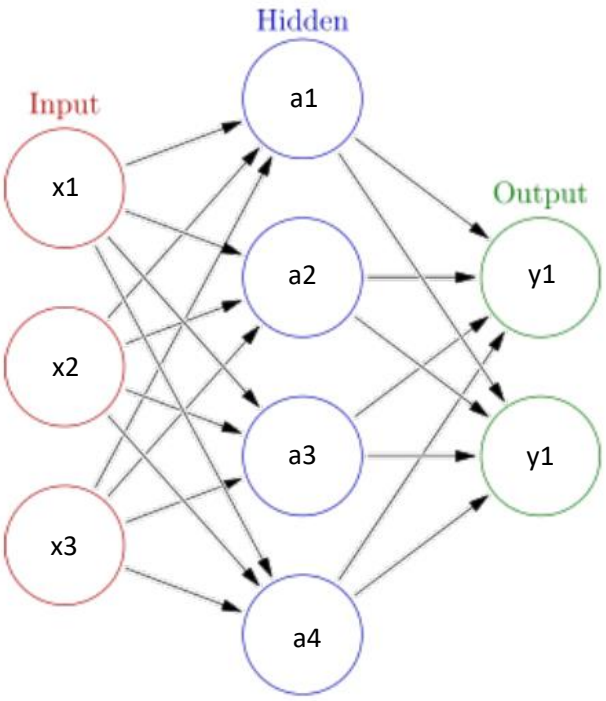
In this iteration, all the models will construct by using scikit learn libraries in python.

For the objectives mentioned in 5.1 and 1.3 about a regression problem. The most traditional way is model working on Linear Model, here is its equation denote equation 1:

$$equation\ 1: \ y_i = \beta_0 + \ \beta_1 x_{i1} + \cdots + \ \beta_p x_{ip} + \varepsilon_i \ \ i = 1, \ldots, n.$$

where y_i is our target values, $(\beta_0\, \beta_1 \ldots \beta_p)$ are the model coefficients, $(x_{i1} \ldots x_{ip})$ are the features. Objective for the linear model is for find best coefficients based on our data that could predict accuracy target values. We will discuss how to choose the best method of optimization in step 5.2.

Second model is Neural Network, equation 2:



Similar with equation 1, x indicates input features, then follow some arrows to the next node in hidden layer, the arrows indicate some activation function with coefficients like equation 1. After feed-forward propagation from input nodes through hidden nodes to the output nodes. The objective of Neural Network is similar with method above which is for find the best coefficients to have an accuracy predict output. We will discuss how to choose the best method of optimization in step 5.2.

Some other models will not be introduced in this iteration, because there are more rely on the training data their objective is for finding similar one to the training data that will not fulfil our datamining objective 2, because we only have aggregate data of whole age groups of fleets and we want to divide separate groups from them, thus no similar data.

5.1.3 modelling requirement assumptions and criteria

The general modelling requirement for both Linear Regression and Neural Network.

- Splitting the dataset into training and test set, if you do not have unseen data for the model then it will easily cause overfitting.
- A loss function, using for training and tell the model how close to the real predict value.
- For the Scikit-Learn requirement, needs to normalized the data, which we already done in step 4.2
- Learning rate, for control the learning speed and able to avoid over fitting.
- Stop criteria, a method for avoid over fitting, might set maximum iteration times or wait until model convergence.

- Noisy data, for avoid over fitting and it will be vectors with all 0 for this problem to indicate no vehicles on road means no emission.

5.2 Datamining method selection

The following algorithms are we select for this iteration:

- Select multiple optimization techniques for Linear model which refer to Ordinary least squares, Stochastic Gradient Descent, and least square with l2 regularization.
- Select Neural Network with 13-X-1 architecture that 13 is the input nodes and 1 is output node, because we have 13 features and 1 output target.

There are many loss functions provided in Scikit-learn for these models that will select based on performance of training data in step 6.

Some other algorithms such as Decision Trees (C5.0), SVM, KNN will not use for this iteration. The reasons are mentioned in step 5.1, they are aim for find most close values in the training data for the test data, they are not suit for this problem's dataset. For example, Decision trees are using tree structure to find nearest instance. SVM is trying to find a hyper-plane could divide N-dimensional data instances into two groups then find the nearest instance, and KNN is using for base on one instance find nearest K vectors.

Splitting dataset with 70% for training and 30% for testing, add noisy data for both test set and training set (example in figure of red rectangles below), other parameters mentioned in 5.1 will set at 6.1 during algorithm selection.

```
# ---- step 5 ----
LPV_X_train, LPV_X_test, LPV_y_train, LPV_y_test = step_5_split_data_add_noisy_data(LPV_df,test_size=0.3)
LCV_X_train, LCV_X_test, LCV_y_train, LCV_y_test = step_5_split_data_add_noisy_data(LCV_df, test_size=0.3)
```

```
     Total LPV new  ...  20+ years of LPV used
7    0.608718792218 ...          0.066959152694
17   0.636917702684 ...          0.141051306867
5    0.621636080669 ...          0.062364386387
1    0.714235993429 ...          0.048626080475
2    0.687310716569 ...          0.051975112716
10   0.619061376415 ...          0.090929901644
6    0.613374294298 ...          0.064640954138
18   0.638267727303 ...          0.139583925828
0    0.742510709923 ...          0.044438448557
14   0.641032021600 ...          0.118992041775
12   0.634350720120 ...          0.106212621897
16   0.639310558085 ...          0.138181948078
8    0.611819784687 ...          0.071063525130
0    0.000000000000 ...          0.000000000000

[14 rows x 13 columns]
```

```
7     0.000003705707
17    0.000003235433
5     0.000003732522
1     0.000003846248
2     0.000003898517
10    0.000003645780
6     0.000003699583
18    0.000002934834
0     0.000004389683
14    0.000003320031
12    0.000003497037
16    0.000003248352
8     0.000003641184
0     0.000000000000
dtype: float64
```

Figure 24: adding noisy data

The objective for select one right algorithm for conduct this iteration is which one able to reflect the most accuracy predict. The procedures:

1. Run multiple algorithms, for linear model, Neural Network, SVM, Decision Tree and KNN. For prove assumption above that SVM, Decision Tree and KNN might cause over fitting or under fitting problem, refer to step 6.1.
2. Comparing models to select one with accuracy more than 80%, refer to step 6.2.
3. Each selected algorithm run serial times with different setting for find the best setting to get the most accuracy models, refer to step 6.3.

6. Conduct Datamining

6.1 Conduct exploratory analysis and discuss

In this iteration using R2 measurement for comparing models in a Regression problem, $\mathrm{R2}(y,\hat{y}) = 1 - \sum ni = 1(yi - \hat{y})2\sum ni = 1\left(yi - \overline{y}\right)2$ where y *is real values,* $\hat{y}$ *is the predict values*. R2 will return a value from 0 – 1, good models will more and more close to 1. There are 4 main measurement criteria, MSE, RMSE, R2 and MAE. MSE is for sum of the square residuals with predict values and real values, RMSE is root of MSE, MAE sum of absolute value of residuals.

Under Random Seed 722, splitting LPV and LCV dataset into 70% of training and 30% of testing set respectively which we done in step 5, the success models will have R2 with greater than 0.8.

6.1.1 conduct on **Decision Tree, SVM, and KNN regressor**

For Decision Tree, SVM, and KNN regressor in sci-kit learn need to manually normalize data which we done in step 4.2 under l2 normalization, and the regressor.score() will return accuracy under R2 measurement.

```
regressors = [ DecisionTreeRegressor(), svm.SVR(), KNeighborsRegressor(), ]
```

```
print('LPV')
for regressor in regressors:
    regressor.fit(LPV_X_train, LPV_y_train)
    print(regressor.score(LPV_X_test, LPV_y_test))

print('LCV')
for regressor in regressors:
    regressor.fit(LCV_X_train, LCV_y_train)
    print(regressor.score(LCV_X_test, LCV_y_test))
```

```
LPV
0.9962056700974871
-0.5084046222277243
0.126402142288204
```

```
LCV
0.997907444247352
-0.43329722091847733
0.1358638840813885
```

Figure 25: Conduct on decision tree, svm and knn regressors

6.1.2 conduct on **Ordinary least squares, Stochastic Gradient Descent, least square with l2 regularization and Neural Network.**

For Stochastic Gradient Descent and Neural Network regressor in sci-kit learn will force data under l2 normalization, LinearRegression() and Ridge() in scikit-learn for compare with Ordinary least squares and least square with l2 regularization.

Thus, we need to input original to these regressors that cancel what I have done in step 4.2, because it will force doing this on these regressors, if we pass data already under l2 normalization these regressors will do it again that will cause models cannot convergence.

```
def step_4_2_normalization(data_df,LPV_cols,LCV_cols):
    LPV_df, LCV_df = data_df[LPV_cols],data_df[LCV_cols]
    # LPV_data,LCV_data = normalize( LPV_df, axis=1, norm='l2'),normalize(LCV_df, axis=1, norm='l2')
    # LPV_df, LCV_df = pd.DataFrame(LPV_data, columns= LPV_cols),pd.DataFrame(LCV_data, columns= LCV_cols)
    return LPV_df, LCV_df
```

```
regressors = [LinearRegression(), SGDRegressor(), Ridge(),
              MLPRegressor(activation='logistic', hidden_layer_sizes=(7, ), max_iter=100000)]
```

```
LPV
0.38095000172085436
-2.411697917337848e+42
0.997968416540807
0.5959563816190072
```

```
LCV
0.8363572429662157
-2.4202889415388593e+41
0.9866950059579116
0.9485934571993228
```

Figure 26: Conduct on linear models and neural network.

Discuss the R2 result with predict results for select best algorithm(s) further in step 6.2.

6.2 Select data-mining algorithm based on discussion

Select algorithms base on 6.1.1 and 6.1.2 which conduct on many algorithms under a same environment, comparing with the predict values into consideration to find the best algorithm(s) for this project.

- Decision Tree:
  Figure 25 illustrated an overfitting case both in LPV and LCV dataset, we keep discovering on the result.

```
LPV
0.9962056700974871
preict values
[3.89851706e-06 3.64118358e-06 3.64577953e-06 3.73252231e-06
 3.32003071e-06 3.49703732e-06 0.00000000e+00]
```

```
LCV
0.9983486957001382
preict values
[4.81309841e-06 4.84855578e-06 4.97519410e-06 4.60371580e-06
 4.84855578e-06 4.86069776e-06 0.00000000e+00]
-0.43329722091847733
```

There are two results in the test set get a same value which means DT is finding the closest vectors in the training data, since our target data do not have a large range then it is reasonable to have a higher R2 score.

Considering these data have not splitting into single group yet for the data mining success criteria 2, mentioned in step 1.3 and 5, Hence we not using Decision Tree model for this iteration.

- SVM models cannot convergence for both LPV and LCV dataset that the results are all same, hence we not using SVM model for this iteration.

```
-0.5084046222277243
preict values
[2.19484156e-06 2.19484156e-06 2.19484156e-06 2.19484156e-06
 2.19484156e-06 2.19484156e-06 2.19484156e-06]
```

```
-0.43329722091847733
preict values
[3.05934103e-06 3.05934103e-06 3.05934103e-06 3.05934103e-06
 3.05934103e-06 3.05934103e-06 3.05934103e-06]
```

- KNN models have a underfitting problem and getting result similar with Decision Tree which not satisfy the datamining second objective that is the reason we not using KNN model.

```
0.1268707770249623
preict values
[3.77651533e-06 3.68495500e-06 3.47047659e-06 3.73550251e-06
 3.30725734e-06 3.38932656e-06 3.17339403e-06]
```

```
0.1358638840813885
preict values
[5.15720106e-06 4.83515968e-06 5.15720106e-06 4.67094270e-06
 4.81827194e-06 4.71749517e-06 4.19700977e-06]
```

- SGD regressor cannot convergence for both LPV and LCV dataset shown as figures below, hence we not using SGD model for this iteration.

LPV:

```
-2.411697917337848e+42
preict values
[3.64413104e+21 4.25443906e+21 4.51797035e+21 3.81551130e+21
 5.29933229e+21 4.87189520e+21 4.71337935e+07]
```

LCV:

```
-2.4202889415388593e+41
preict values
[-2.98386359e+20 -3.47728251e+20 -2.83840792e+20 -3.97423093e+20
 -3.46063765e+20 -3.37241210e+20  2.98568923e+09]
```

- Ridge regressor have better performance than linear regression regressor for both LPV and LCV dataset, that could prove l2 normalization will improve the model accuracy. All the predict results are different under a linear model which is count for a new value rather than fining a similar value.

  Linear regression LPV:

```
0.38095000172085436
preict values
[ 8.84740756e+00  7.54445898e+00  1.13688112e+01  8.67562212e+00
  8.44414114e+00  3.92981533e+00 -5.68906700e-10]
```

  Linear regression LCV:

```
0.8363572429662157
preict values
[1.91639622e+00 1.56659557e+00 1.63111599e+00 1.90837118e+00
 1.48360499e+00 1.44501048e+00 6.82520707e-12]
```

  Ridge LPV (Linear Regression with l2 normalization):

```
0.997968416540807
preict values
[7.24176821e+00 7.75970512e+00 7.42727849e+00 7.46720780e+00
 7.68156274e+00 7.40577249e+00 5.94947553e-03]
```

  Ridge LCV (Linear Regression with l2 normalization):

```
0.9866950059579116
preict values
[ 1.71425207  1.78450488  1.53293001  2.01170251  1.8987601   1.74247125
 -0.00253902]
```

  Ridge sufficient for the datamining objectives (one, two), hence it will be select for the data mining.

- Neural Network have good R2 results on LCV, but not have good results on predict values, NN models are commonly using in real business and it is a quite tricky algorithm we need to tune many parameter to find it fitness for this problem, thus we cannot move it in confidence level and we will continue tune parameters for

them in the next step 6.3.

```
0.5959563816190072
preict values
[5.74169149 5.74169149 5.74169149 5.74169149 5.74169149 5.74169149
 0.0198571 ]
```

```
0.9485934571993228
preict values
[1.78501735 1.78501735 1.78501735 1.78501735 1.78501735 1.78501735
 0.03864563]
```

Select Ridge and Neural Network for the next steps.

6.3 Build appropriate models

Under same environment in step 6.2 for tune parameters in the Neural Network and Ridge Linear Regression.

Ridge Regression:
There main parameters that we can tuning for Ridge Regression in Scikit-learn is the solver. it has 6 different kind of solvers and we run all of them to find the results difference. Other parameters like iteration times, the default setting is 1000 times which is enough for ridge regressor convergence in this problem and it

```
regressors = [Ridge(solver='sag'),
              Ridge(solver='svd'),
              Ridge(solver='cholesky'),
              Ridge(solver='saga'),
              Ridge(solver='lsqr'),
              Ridge(solver='sparse_cg'),]
```

```
solver sag, LPV r2 score = 0.998475, LCV r2 score = 0.986506
solver svd, LPV r2 score = 0.997968, LCV r2 score = 0.986695
solver cholesky, LPV r2 score = 0.997968, LCV r2 score = 0.986695
solver saga, LPV r2 score = 0.998493, LCV r2 score = 0.990240
solver lsqr, LPV r2 score = 0.996378, LCV r2 score = 0.956272
solver sparse_cg, LPV r2 score = 0.996258, LCV r2 score = 0.925246
```

Ridge predict values of saga solver for LPV test set:

```
[7.37862089 7.57125436 7.46228667 7.4911479  7.6871131  7.57488602
 0.01936457]
```

Ridge predict values of saga solver for LCV test set:

```
[1.69377306 1.69553252 1.58653229 2.09499236 1.74810028 1.832064
 0.02575751]
```

The saga and sag solver have the best results (r2 and predict values) both on LPV and LCV and it is an improved version of stochastic average gradient descent. The difference between ridge regressor and sgd regressor which we not choose on step 6.2 is via different means. The saga solver has an unbiased version of means than sag. The benefit of Stochastic Gradient Descent is often faster than other solvers, and it is a local search technique which will easily get stuck at local optimal, however it has a stochastic operation which will help move out from local optimal.

Neural Network:

The main idea is tuning the architecture of NN, for this problem, it will have 13 input nodes and 1 output node. For the hidden layer, more hidden layers will cause overfitting and lack of nodes will cause underfitting. The initial parameter is setting half of input nodes for first hidden layer (6 nodes) and half of first layer node for the second layer (3 nodes), and trying different combination of architectures. For the activation function, logistic function will be best fit for this problem, others always hard to convergence.

```
regressors = [MLPRegressor(activation='logistic', hidden_layer_sizes=(6, 3), max_iter=10000),
              MLPRegressor(activation='logistic', hidden_layer_sizes=(7, 4), max_iter=10000),
              MLPRegressor(activation='logistic', hidden_layer_sizes=(6,), max_iter=10000),
              MLPRegressor(activation='logistic', hidden_layer_sizes=(3,6,), max_iter=10000),
              MLPRegressor(activation='logistic', hidden_layer_sizes=(7,), max_iter=10000),
              MLPRegressor(activation='logistic', hidden_layer_sizes=(8,), max_iter=10000),
              MLPRegressor(activation='logistic', hidden_layer_sizes=(6,1,), max_iter=10000),
              MLPRegressor(activation='logistic', hidden_layer_sizes=(6,2,), max_iter=10000),
              MLPRegressor(activation='logistic', hidden_layer_sizes=(6,4,),max_iter=10000),]
```

LPV	LCV
0.6149785552353468	0.8032965899826979
0.9006714563463101	0.820014566111159
-4.1255137728994145	0.688992265856677
0.7057502954956714	0.3020523289184949
0.6959292178604686	0.9496799305920588
-4.755635278054433	0.7792390522178138
0.7998833827564058	0.6367907026809562
0.852909496077974	0.25935464930566554
0.7883384813543254	0.9143926710502239

The archicture 13-7-4-1 has best r2 result on LPV and third r2 result on LCV. The architecture 13-3-6-1 show the best performance on LCV, and we can see that different architecture have different performance, it might have good performance on one dataset, but cannot convergence on another one. That is for a particular problem might need a particular architecute, after a trade-off between two datasets we select 13-7-4-1 archicture to discover on its predict result.

```
LPV
0.910440703651102
preict values
[6.97769387 6.97769387 6.97769387 6.97769387 6.97769387 6.97769387
 1.52010559]
```

```
LCV
0.8955685112911873
preict values
[1.79445144 1.79445144 1.79445144 1.79445144 1.79445144 1.79445144
 0.38512997]
```

Although we already set the random seed 722, every time training the NN models will cause a different one, because its randomly update parameters inside. However, the results will remain in a range by its architecture. There predict values are all same which not satisfy the datamining objective 2, the reason of this is the output node activation function,

$$logistic\ function: f(x) = \frac{1}{1 + e^{-x}}$$

which is a monotonic function return value greater than 1, and for x in a small range will return values in a smaller range.

To sum up of step 6, choosing ridge regressor for the data mining model, it satisfies all the data mining goals. There are two possible solvers for ridge regressor, we select sag first and if it does not have a good result then we turn into saga solver. Neural Network satisfies objective one but not for two, because Scikit learn does not have a proper activation function for this problem.

7. Datamining

7.1 Create and justify test design

For Datamining Objective One:

Splitting data set to 70% as a training set and 30% as a testing set to avoid overfitting and underfitting the model. The Ridge model based on the saga solver and maximum 1000 times of iterations.

After splitting the training set and testing set, adding noisy data into these files respectively, for avoid overfitting and also make the model learn that no vehicles on road indicate to no Co2 emissions.

For Datamining Objective Two:

Prepare splitting age groups set base on the data in 2017, One reason is the real emission value for 2018 is missing which we imputed in Data Pre-processing. Comparing using mean

values, data in 2017 are more fit current situation. We create files with 10 instances of 2017 and trim it only contains one group data.

```
split_2017_LPV_rows = [[1.59009400e+06, 1.61762700e+06, 1.43115413e+01, 2.84522000e+05, 2.54697000e+05,
                        4.53646000e+05, 2.51078000e+05, 3.46148000e+05, 2.89449000e+05, 2.59108000e+05,
                        4.61501000e+05, 2.55426000e+05, 3.52141000e+05],
                       [284522.0, 0.0, 2.975101564, 284522.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
                       [254697.0, 0.0, 7.975101564, 0.0, 254697.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
                       [453646.0, 0.0, 12.97510156, 0.0, 0.0, 453646.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
                       [251078.0, 0.0, 17.97510156, 0.0, 0.0, 0.0, 251078.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
                       [346148.0, 0.0, 22.97510156, 0.0, 0.0, 0.0, 0.0, 346148.0, 0.0, 0.0, 0.0, 0.0, 0.0],
                       [0.0, 289449.0, 9.130000000, 0.0, 0.0, 0.0, 0.0, 0.0, 289449.0, 0.0, 0.0, 0.0, 0.0],
                       [0.0, 259108.0, 14.13000000, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 259108.0, 0.0, 0.0, 0.0],
                       [0.0, 461501.0, 19.13000000, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 461501.0, 0.0, 0.0],
                       [0.0, 255426.0, 24.13000000, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 255426.0, 0.0],
                       [0.0, 352141.0, 29.13000000, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 352141.0], ]

split_2017_LCV_rows = [[4.76173000e+05, 1.06367000e+05, 1.25145441e+01, 8.52030000e+04, 7.62720000e+04,
                        1.35849000e+05, 7.51880000e+04, 1.03658000e+05, 1.90320000e+04, 1.70370000e+04,
                        3.03460000e+04, 1.67950000e+04, 2.31550000e+04],
                       [085203.0, 0.0, 2.868356783, 085203.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
                       [076272.0, 0.0, 7.868356783, 0.0, 076272.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
                       [135849.0, 0.0, 12.86835678, 0.0, 0.0, 135849.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
                       [075188.0, 0.0, 17.86835678, 0.0, 0.0, 0.0, 075188.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
                       [103658.0, 0.0, 22.86835678, 0.0, 0.0, 0.0, 0.0, 103658.0, 0.0, 0.0, 0.0, 0.0, 0.0],
                       [0.0, 019032.0, 07.98000000, 0.0, 0.0, 0.0, 0.0, 0.0, 019032.0, 0.0, 0.0, 0.0, 0.0],
                       [0.0, 017037.0, 12.98000000, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 017037.0, 0.0, 0.0, 0.0],
                       [0.0, 030346.0, 17.98000000, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 030346.0, 0.0, 0.0],
                       [0.0, 016795.0, 22.98000000, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 016795.0, 0.0],
                       [0.0, 023155.0, 27.98000000, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 023155.0], ]
```

The first rows will be the 2017 data; thus, we can compare with each group predict values against the whole groups, and we can also compare with the whole groups predict values with real values, these are mentions in step 1.3.1.2 benchmark.

Preparing similar 2017 data set form LPV and LCV and convert it to Data Frame for testing the Objective 2.

```
def step_7_1_prepare(rows, columns):

    test_obj_2 = {x:[] for _,x in enumerate(columns)}
    for row in rows:
        for idx, x in enumerate(columns):
            test_obj_2[x].append(row[idx])

    return pd.DataFrame(test_obj_2)
```

	Total LPV new	...	20+ years of LPV used
0	1590094.0	...	352141.0
1	284522.0	...	0.0
2	254697.0	...	0.0
3	453646.0	...	0.0
4	251078.0	...	0.0
5	346148.0	...	0.0
6	0.0	...	0.0
7	0.0	...	0.0
8	0.0	...	0.0
9	0.0	...	0.0
10	0.0	...	352141.0

[11 rows x 13 columns]

	Total LCV new	...	20+ years of LCV used
0	476173.0	...	23155.0
1	85203.0	...	0.0
2	76272.0	...	0.0
3	135849.0	...	0.0
4	75188.0	...	0.0
5	103658.0	...	0.0
6	0.0	...	0.0
7	0.0	...	0.0
8	0.0	...	0.0
9	0.0	...	0.0
10	0.0	...	23155.0

[11 rows x 13 columns]

7.2 Conduct data mining

Run the pipeline for LPV and LCV in figure 27, setting Random Seed with 722, Split the training set and test set with 70% and 30% respectively.

7.2.1 For Data Mining Objective One:

Dataset select and split from the step 1-5 above. Construct ridge regressor with saga solver. Verification of fulfil of Data Mining Objective One with the test set, which is the predict values do not have a large gap comparing real value (Step1.3), and one measurement of this situation introduced in Step 6 is R2 score.

```
# ---- step 5 ----
LPV_X_train, LPV_X_test, LPV_y_train, LPV_y_test = step_5_split_data_add_noisy_data(LPV_df, test_size=0.3)
LCV_X_train, LCV_X_test, LCV_y_train, LCV_y_test = step_5_split_data_add_noisy_data(LCV_df, test_size=0.3)
```

```
# --- build DM models ---
LPV_regressor,LCV_regressor = step_7_2_build_model(_LPV_X_train, LPV_y_train),\
                              step_7_2_build_model(_LCV_X_train, LCV_y_train)

# --- DM objective 1 ---
LPV_score, LCV_score = LPV_regressor.score(LPV_X_test, LPV_y_test),\
                       LCV_regressor.score(LCV_X_test, LCV_y_test)

LPV_predicts, LCV_predicts = LPV_regressor.predict(LPV_X_test), \
                             LCV_regressor.predict(LCV_X_test)

print('Data Mining Objective One')
print('LPV R2 score = %f, LCV R2 score = %f'%(LPV_score, LCV_score))
```

```
Data Mining Objective One
LPV R2 score = 0.998475, LCV R2 score = 0.986506
```

Figure 27: Conduct Data Mining, DM Objective One

7.2.2 For Data Mining Objective Two:

Data sets designed and built from step 7.1, using regression models for predict single age group values is test the fitness of Data Mining Objective Two. The details are the sum of the single group values should not have a large residual with real values.

```
# ---- step 7.1 ----
LPV_2017_obj_2, LCV_2017_obj_2 = step_7_1_prepare(split_2017_LPV_rows,LPV_df.columns[:-1]),\
                                 step_7_1_prepare(split_2017_LCV_rows,LCV_df.columns[:-1])
```

```
# --- DM objective 2 ---
print('Data Mining Objective Two')
LPV_2017_predict,LCV_2017_predict = LPV_regressor.predict(LPV_2017_obj_2), LCV_regressor.predict(LCV_2017_obj_2)
print('2017 LPV real value = 8.07740664325, sum of predict values %f' % np.sum(LPV_2017_predict[1:]))
print('2017 LCV real value = 2.47222703164, sum of predict values %f' % np.sum(LCV_2017_predict[1:]))
# print(LPV_2017_obj_2.columns[2:])
```

```
Data Mining Objective Two
2017 LPV real value = 8.07740664325, sum of predict values 8.015749
2017 LCV real value = 2.47222703164, sum of predict values 2.461277
```

Figure 28: Conduct Data Mining, DM Objective 2

To Sum up for step 7.2, the model we selected, and constructed with parameters tuned in step 6 is fulfil for our datamining objectives. In Objective 1 results shown on figure 27, all the datasets have over 0.99 r2 score in range [0,1] that indicates our predict values have a tiny gap to real values. In Objective 2 results shown on figure 28, the model fulfils Data mining Objective Two, the sum of single groups value has a slight gap between LPV real emission value in 2017 and a tiny gap with LCV 2017 emission value. We will discover the patterns from result in next step 7.3.

7.3 Search for patterns

For discovering patterns, we need to get all the predict values for the Objective One output and Two.

For the Objective One, we output the predict result of test set and label the year of it by using the index as connection. The red rectangles in Figure 29 indicate year in 2000, but the real values and predict values are 0 and 0 close, that is because we have add nosiy data in step 5 for both training and testing data, that the new adding data index will be 0 as the data frame recognized. Indexes will not attend the model training, if we add nosiy data before splitting dataset, it will might cause the training or testing set not contain noisy data, moreover, 2000 year Co2 emssion values are also missing, because these won't influence it.

```
# ---- step 7.3 ----
print('')
print('Data Mining Objective One result discover, LPV')
for idx,y_real_value in enumerate(LPV_y_test):
    year = cleaned_data_df['Period'][LPV_X_test.index[idx]]
    print('%i year, real value = %f, predict value = %f' %(year, y_real_value, LPV_predicts[idx]))

print('Data Mining Objective One result discover, LCV')
for idx,y_real_value in enumerate(LCV_y_test):
    year = cleaned_data_df['Period'][LCV_X_test.index[idx]]
    print('%i year, real value = %f, predict value = %f' %(year, y_real_value, LCV_predicts[idx]))
```

```
Data Mining Objective One result discover, LPV
2003 year, real value = 7.375728, predict value = 7.378621
2009 year, real value = 7.626856, predict value = 7.571254
2011 year, real value = 7.575258, predict value = 7.462287
2004 year, real value = 7.671046, predict value = 7.491148
2015 year, real value = 7.662124, predict value = 7.687113
2013 year, real value = 7.415187, predict value = 7.574886
2000 year, real value = 0.000000, predict value = 0.019365
```

```
Data Mining Objective One result discover, LCV
2004 year, real value = 1.582260, predict value = 1.693773
2009 year, real value = 1.774602, predict value = 1.695533
2002 year, real value = 1.559698, predict value = 1.586532
2014 year, real value = 1.993417, predict value = 2.094992
2010 year, real value = 1.830283, predict value = 1.748100
2011 year, real value = 1.861729, predict value = 1.832064
2000 year, real value = 0.000000, predict value = 0.025758
```

Figure 29: DM Objective One predict values output.

Figure 29 illustrated that all the predict values are have a small gap with real values, the residuals are smaller in one decimal place, that is make our objective 2 results become more reliable.

For the Objective Two, we output the predict result of the 2017 test set we designed in Step 7.1.

```
print('')
print('Data Mining Objective TWO result discover, 2017 LPV')
for idx,single_age_group_value in enumerate(LPV_2017_predict):
    print('%s : %f '%(LPV_2017_obj_2.columns[2:][idx], np.abs(single_age_group_value)))
print('')
print('Data Mining Objective TWO result discover, 2017 LCV')
for idx,single_age_group_value in enumerate(LCV_2017_predict):
    print('%s : %f '%(LCV_2017_obj_2.columns[2:][idx], np.abs(single_age_group_value)))
```

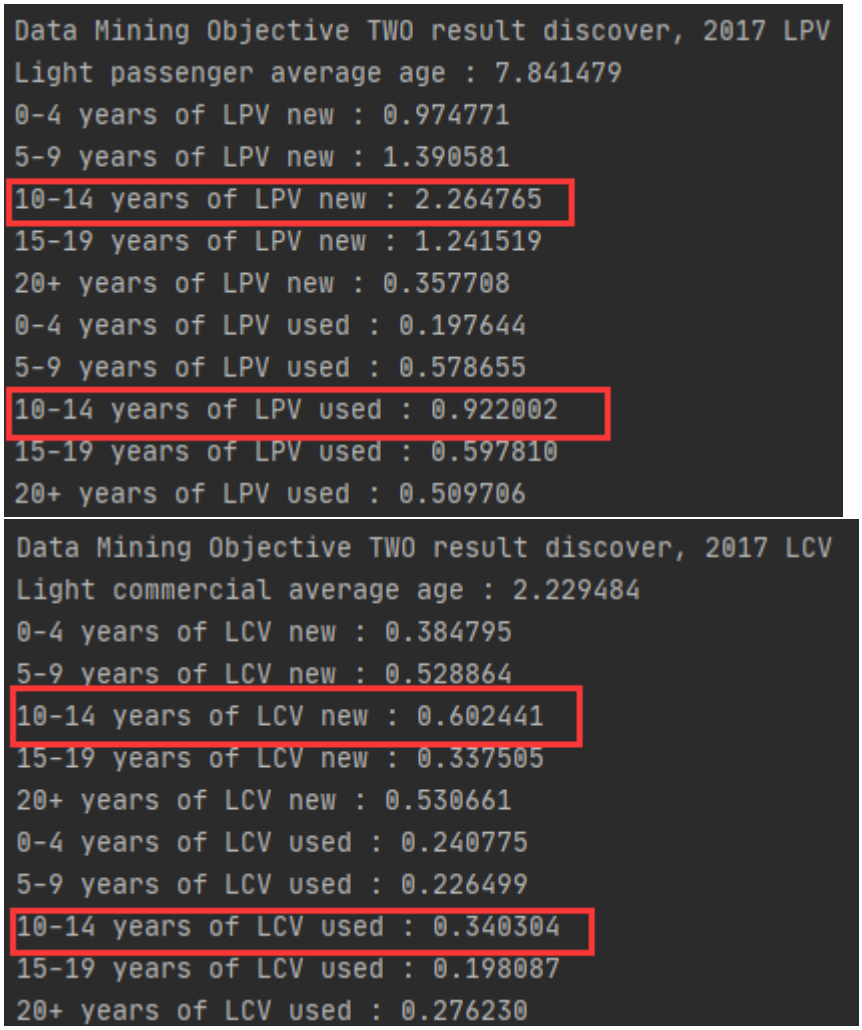


Figure 30: DM Objective Two predict values output.

The relaitionships that we discover for datamining objective 2 are:

1. New import fleets have more Co2 emissions than Used import fleets
2. 10-14 age groups contribute the most Co2 emissions among new import groups or used groups.
3. Some age groups have similar emissions

8 Result Analysis

8.1 Study and discuss the mined patterns

In this part, we will discover the patterns found in step7.3 with its input values, and consider the pattern meaning in the real world. We generate the percentage of single age groups of Co2 emission for LPV and LCV data in 2017.

LPV:

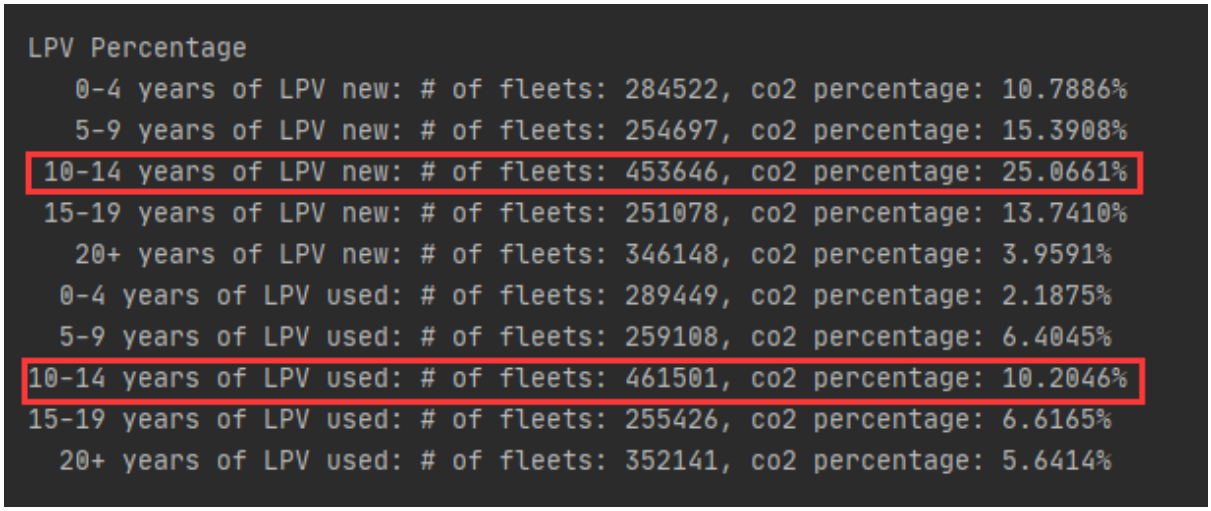


Figure 31:

LCV:

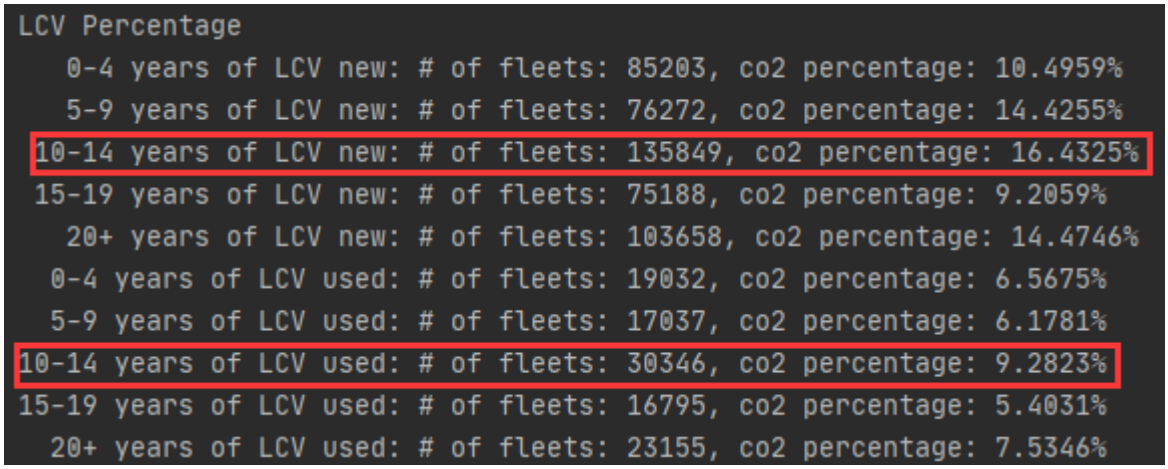


Figure 32:

The figure 31 and 32 shows the models explained age trend in correct way, because the 20+ year groups have samiliar number of fleets, but the used fleets contribute more Co2 emission, it is same as common sense that is 20+ year is counting since they come into New Zealand. They are import as used vehicles that they must have higher used age than new import group. The 20+ year age group of new import excess used import only if they have much number of fleets, like cases in figure 32. The LCV model under sag solver are more accuracy than LPV model under sag solver.

Relationships:

- New import fleets have more Co2 emissions than Used import fleets
  For LCV, the reason of this relationship is the number of new imports fleets greater than used fleets.

  For LPV, the trend of the elder fleets contributes more Co2 emission is right. However, this model more bias on the new import features that explained age trend right and it will not influence LCV result, because they are building from different

models. We will run the models with different random seed and saga solver for discover LPV pattens again in step 8.2.

- 10-14 age groups contribute the most Co2 emissions among new import groups or used groups.
  This is because the models could explain the age trend correctly and 10-14 age groups have largest fleets then they contribute most.

- Some age groups have similar emissions
  Similar reason with relationship2, when they have similar emissions and if one group is elder then they must have less fleets than another group. It has shown this on LCV with group 0-4 years of used and 5-9 years of used.

8.2 Visualize the data, results, models and patterns

For Data Mining Objective One:

LPV predict values against real values:

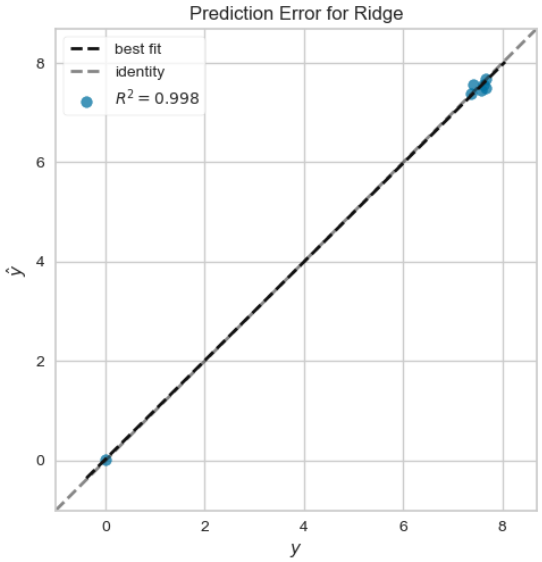


Figure 33:

LPV Fitting Regression Lines:

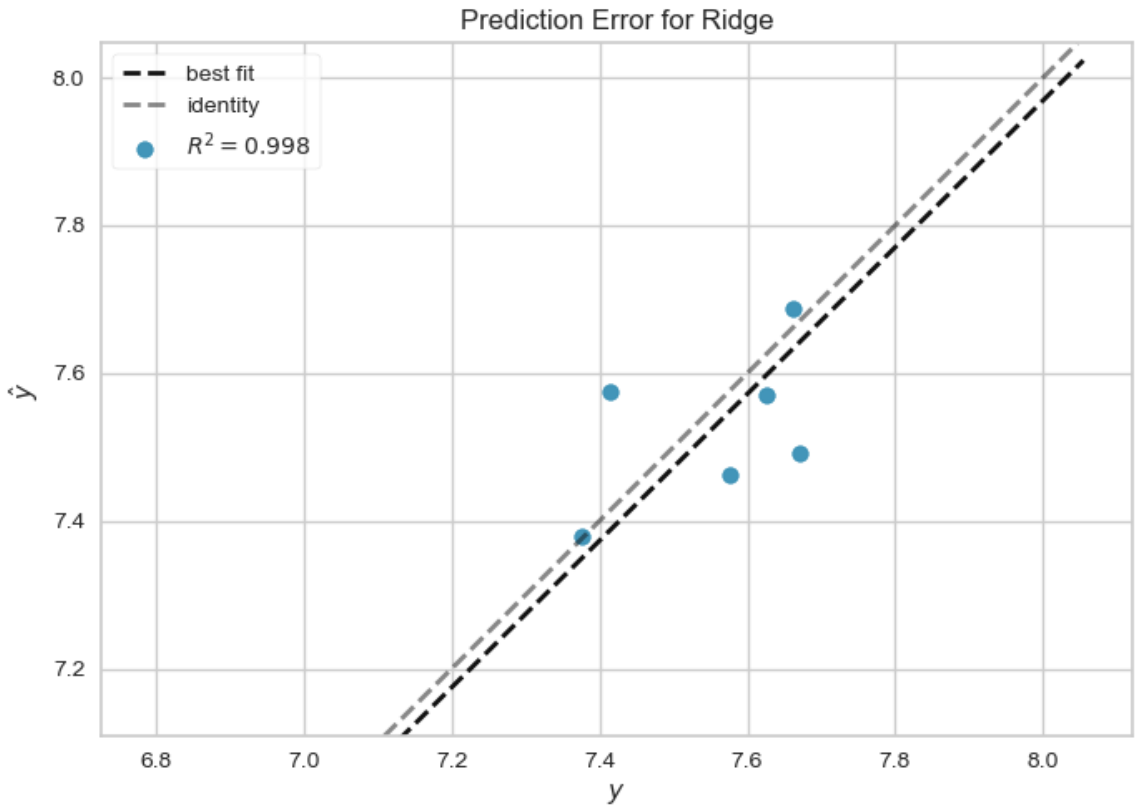


Figure 34:
LPV Residual points for training set and test set:

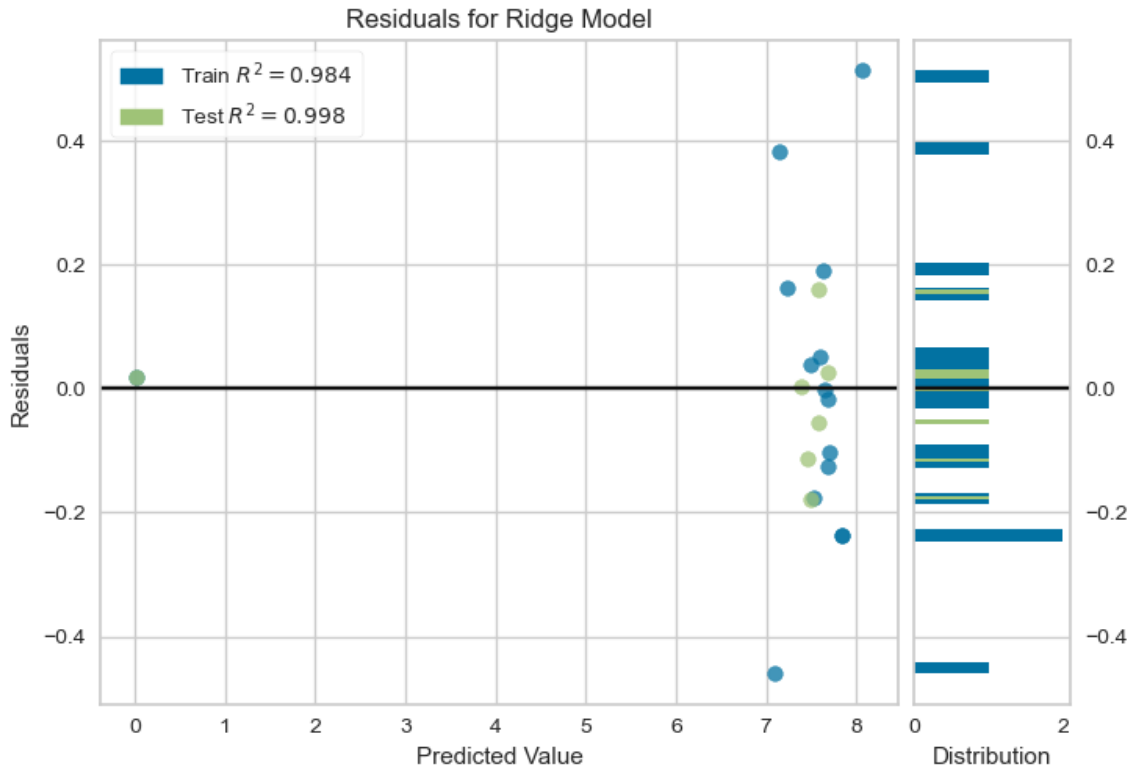


Figure 35:

LCV predict values against real values:

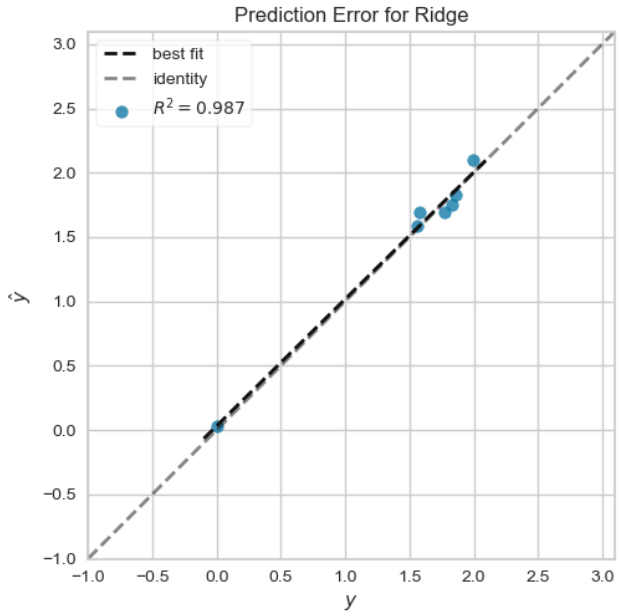


Figure 36:
LCV Fitting Regression Lines:

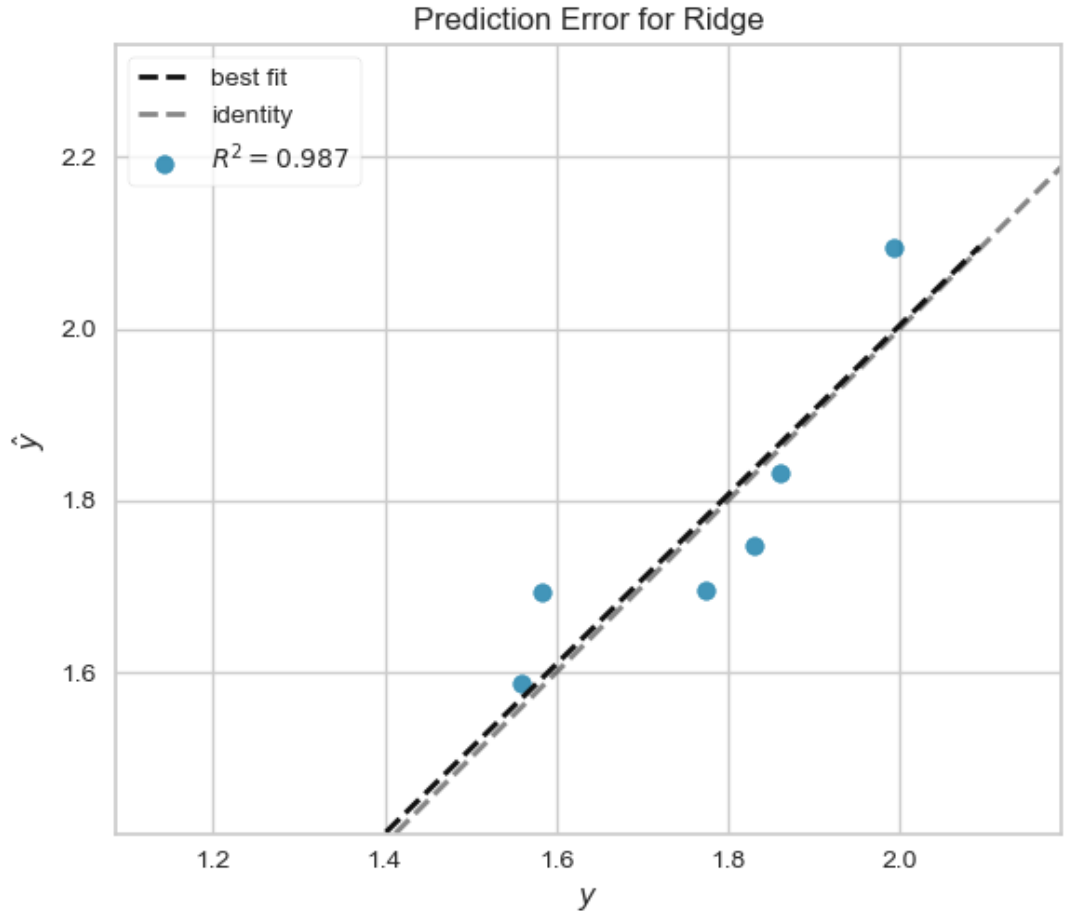


Figure 37:

LCV Residual points for training set and test set:

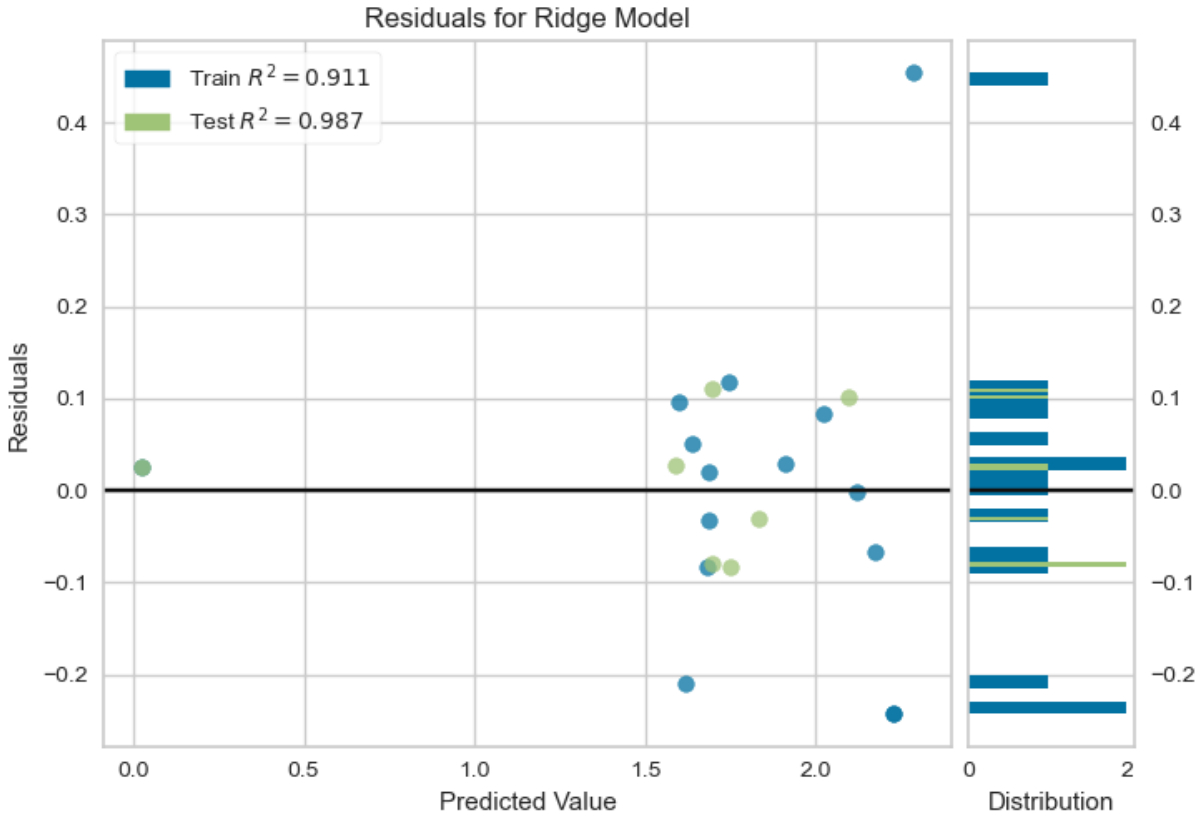


Figure 38:

For Data Mining Objective Two:

LPV Percentage Pie Chart:

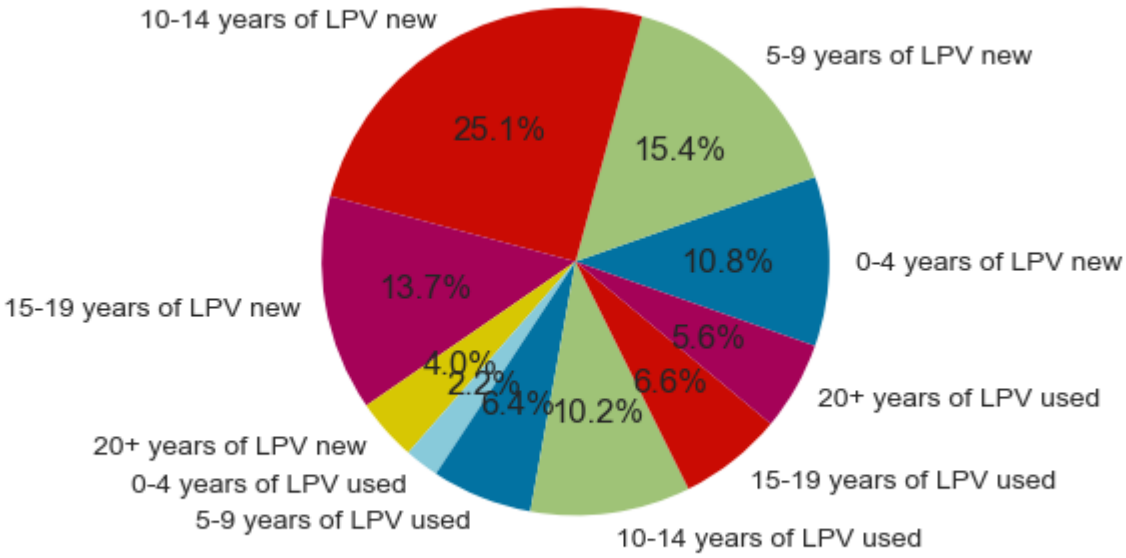


Figure 39:

LCV Percentage Pie Chart:

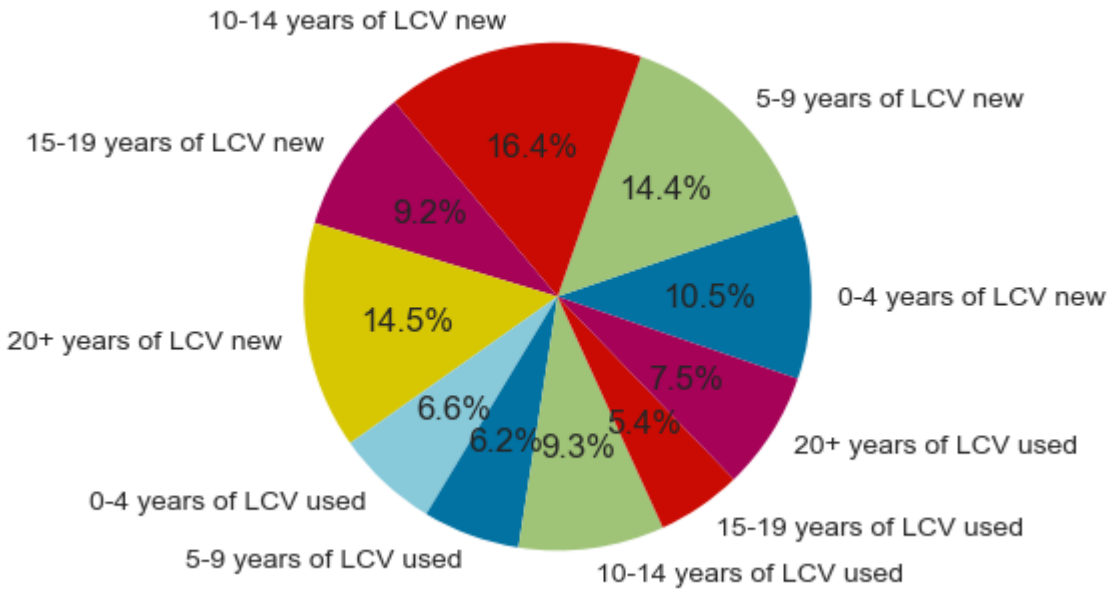


Figure 40:

8.3 Interpret the results, models and patterns

Firstly In 8.1 we evaluate the successful of data mining goals by the criteria, benchmarks and visualization figures, we introduced in Step 1.3.1, 1.3.1.2 and 8.2.

1. Data Mining objective one
Benchmark: For the test data have $small\ residuals$: $Real\ Value - Predict\ Value$
In Figure 29 at Step 7.3 shows all the predict values that predict from test set have a small residuals to the real values.

The regression figures, the diagonal lines with dash lines are the best fitting lines, solid lines are for the fitting regression lines. it shows LCV model have better result than LPV model because LCV lines are closer and LPV seem a small gap of parallel.

The residual figures could find all the data are follow the normal distribution on the right-hand side of distribution graph, and it shows that there are no over fitting and under fitting problem because LPV have r2 score 0.984 and 0.998 for training set and test set. 0.911 and 0.987 for the LCV training and testing set. All the accuracy is exceeded 80%.

Hence, we conclude the model satisfy the Data mining Objective One

2. Data Mining Objective Two

Benchmark: $\sum single\ group\ predict\ value\ -\ total\ group\ predict\ value$
In Figure 28 at Step 7.2.2 illustrated the sum of single age groups values have a small residuals with real value.

```
Data Mining Objective Two
2017 LPV real value = 8.07740664325, sum of predict values 8.015749
2017 LCV real value = 2.47222703164, sum of predict values 2.461277
```

Recall Figure28:

That the accuracy is over 80% to 8.01575/8.0774 = 0.99236 for LPV and 2.461227/2.472227 =0.9955708

It has shown that the percentage of ages could be divided from whole groups, from Figure 30, 39 and 40. we will assess these patterns in step8.4 for decision making for the project business objectives.

Hence, we are able to conclude the data mining objective two is satisfiable.

8.4 Assess and evaluate results, models and patterns

After we conclude that our model satisfies our Data Mining Objective, in this step we discuss about our matching business success criteria on the model discovered patterns.

- According to the models, these models could apply for the subjective where we mentioned in step 1.3 for predict Co2 emissions in the future, because it could predict emission values in a reasonable range. For the Decision maker of New Zealand, they are able to control the number of fleets registrate or work off from road then they can base on these data to estimate trend of Co2 emission.
- The business objective is for find the different age group of fleets take the occupation of the Co2 emission, and notice that the assumption of this project in Step 1.2 mentioned if the fleets with age greater or equal to 10 take 70% of CO2 Emission then the NZ Transport needs to consider apply extra Greenhouse Exhaust Test. The fleets with over 10 years are 5 - 20+ groups in used import vehicles and 10 – 20+ groups in new import vehicles.

Counting the occupation:

```
LPV Percentage
   0-4 years of LPV new: # of fleets: 284522, co2 percentage: 10.7886%
   5-9 years of LPV new: # of fleets: 254697, co2 percentage: 15.3908%
 10-14 years of LPV new: # of fleets: 453646, co2 percentage: 25.0661%
 15-19 years of LPV new: # of fleets: 251078, co2 percentage: 13.7410%
   20+ years of LPV new: # of fleets: 346148, co2 percentage: 3.9591%
  0-4 years of LPV used: # of fleets: 289449, co2 percentage: 2.1875%
  5-9 years of LPV used: # of fleets: 259108, co2 percentage: 6.4045%
10-14 years of LPV used: # of fleets: 461501, co2 percentage: 10.2046%
15-19 years of LPV used: # of fleets: 255426, co2 percentage: 6.6165%
  20+ years of LPV used: # of fleets: 352141, co2 percentage: 5.6414%
```

= 71.6332%

```
LCV Percentage
   0-4 years of LCV new: # of fleets: 85203, co2 percentage: 10.4959%
   5-9 years of LCV new: # of fleets: 76272, co2 percentage: 14.4255%
 10-14 years of LCV new: # of fleets: 135849, co2 percentage: 16.4325%
 15-19 years of LCV new: # of fleets: 75188, co2 percentage: 9.2059%
   20+ years of LCV new: # of fleets: 103658, co2 percentage: 14.4746%
  0-4 years of LCV used: # of fleets: 19032, co2 percentage: 6.5675%
  5-9 years of LCV used: # of fleets: 17037, co2 percentage: 6.1781%
10-14 years of LCV used: # of fleets: 30346, co2 percentage: 9.2823%
15-19 years of LCV used: # of fleets: 16795, co2 percentage: 5.4031%
  20+ years of LCV used: # of fleets: 23155, co2 percentage: 7.5346%
```

=

=68.5111 %

Compute the weight average of occupation among LPV and LCV:

$$\left(\frac{\#\,LPV}{\#\text{LPV}+\#\text{LCV}} * 71.6332\% + \frac{\#\,LCV}{\#\text{LPV}+\#\text{LCV}} * 68.5111\%\right) * 100\% = \left(\frac{3207721}{3207721+582540} * 71.6332\% + \frac{582540}{3207721+582540} * 68.5111\%\right) * 100\% = 71.15335227\%$$

In conclusion, rely on the project assumption and the weight average of the occupation of Co2 emissions reflect that it is necessary to suggest the NZ Transport apply extra Greenhouse Exhaust Test.

8.5 Multiple iteration

There are two aspect of the multiple iteration:

1. Test the pipeline robust
   It is testing for the model stability, and robust that is working for various data. We test the robust through change the random seed, while the random seed has

changed then the training and test data set will be different, because they splitting with different random seed. After many iterations of different seed then the model could reflect a stable result which is the r2 score won't less than 0.9 then we could conclude the system is stable and robust.

2. Get a better result
   In Step 8.1, we find the model with sag solver could explained age trend correct, and using saga solver might be another choice to improve the model for LPV data set, hence this iteration will work for discover this problem.

8.5.1 Test the pipeline robust:

Set random seed to 123 for run the pipeline again, where we just need to change the hyper parameter at project pipeline class front.

```
import random
from help_methods import *

# hyper parameters
# pd.set_option('display.max_columns', None)
pd.set_option('display.precision',12)

# SEED = 722
SEED = 123
random.seed(SEED)
np.random.seed(SEED)
```

For Data Mining Objective one:

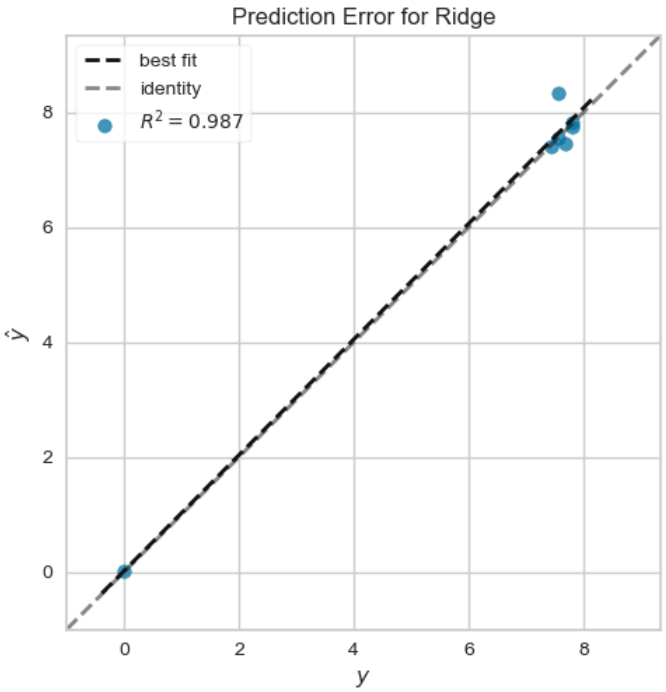
R2 score:

```
Data Mining Objective One
LPV R2 score = 0.986582, LCV R2 score = 0.990117
```

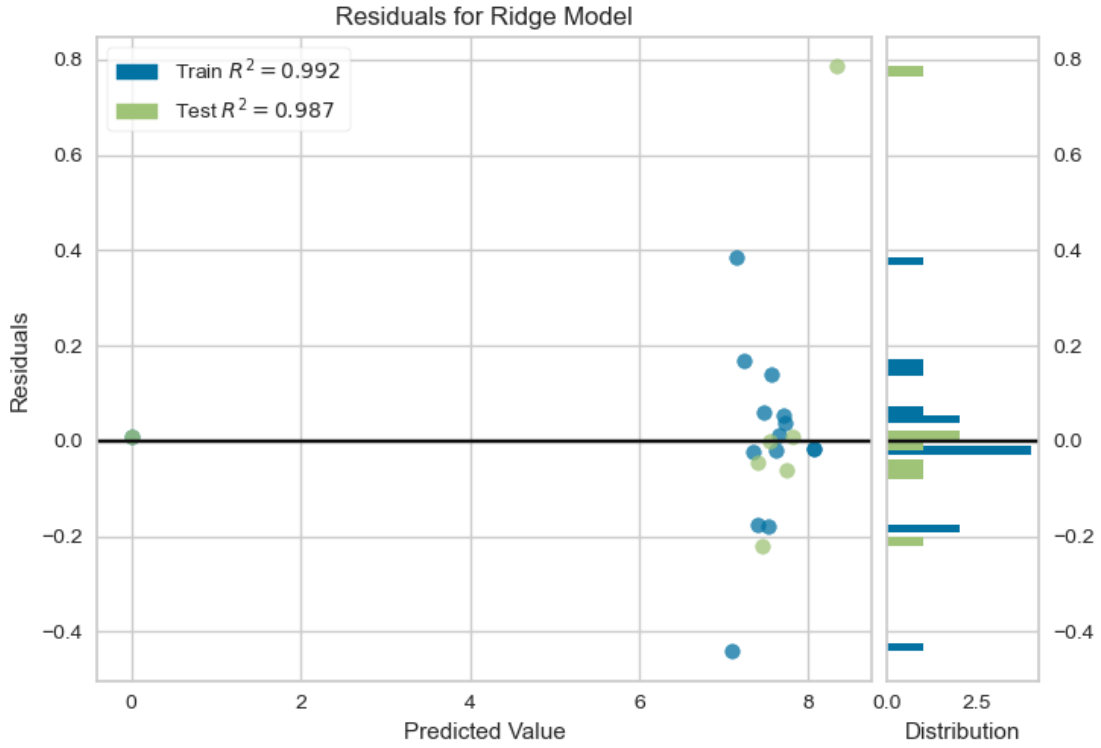
LPV test set predict value and real value compare:

```
Data Mining Objective One result discover, LPV
2018 year, real value = 7.546587, predict value = 8.332351
2005 year, real value = 7.549507, predict value = 7.547572
2004 year, real value = 7.671046, predict value = 7.450356
2016 year, real value = 7.800260, predict value = 7.810558
2007 year, real value = 7.796914, predict value = 7.736865
2012 year, real value = 7.444775, predict value = 7.401041
2000 year, real value = 0.000000, predict value = 0.007863
```

LPV regression figure:



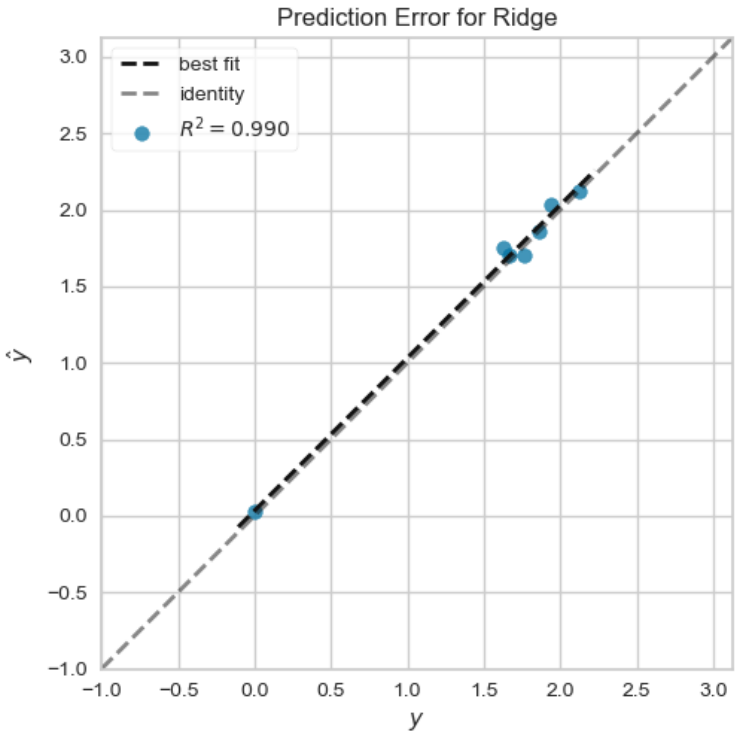
LPV residuals figure:



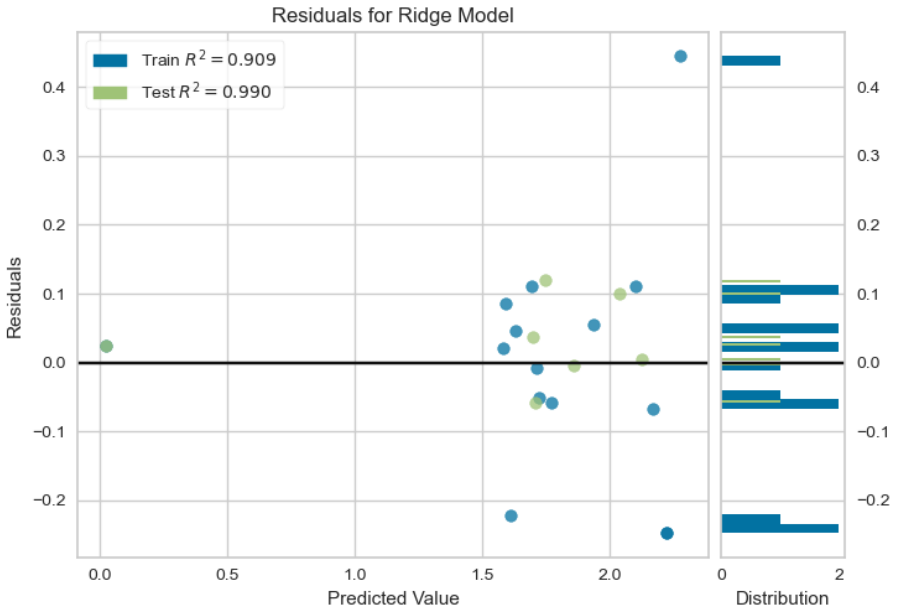
LCV test set predict value and real value compare:

```
Data Mining Objective One result discover, LCV
2006 year, real value = 1.663934, predict value = 1.700581
2008 year, real value = 1.764464, predict value = 1.706391
2013 year, real value = 1.937471, predict value = 2.038372
2011 year, real value = 1.861729, predict value = 1.857957
2005 year, real value = 1.628644, predict value = 1.748923
2015 year, real value = 2.121840, predict value = 2.125894
2000 year, real value = 0.000000, predict value = 0.025446
```

LCV regression figure:



LCV residuals figure:



The all training set and test set of LPV or LCV are different under seed 123 comparing seed 722. The R2 scores for all set of LPV or LCV are over 0.9, and the figures show that the models are similar that training and perform under different sets. One interesting fact is that is the 2018 data are missing values and our testing set contain 2018 data and predict as 8.332 Co2 emission. The fact of 2017 Co2 emission of LPV is 8.077, comparing the fleets growth then the predict value is reasonable.

Hence base on these data we can conclude the pipeline are robust and stable for Data Mining Objective One.

For Data Mining Objective two:

Compare with predict values:

```
Data Mining Objective Two
2017 LPV real value = 8.07740664325, sum of predict values 8.131076
2017 LCV real value = 2.47222703164, sum of predict values 2.453916
```

LPV single groups results discover:

```
Data Mining Objective TWO result discover, 2017 LPV
Light passenger average age : 8.060324
0-4 years of LPV new : 1.056896
5-9 years of LPV new : 1.289108
10-14 years of LPV new : 2.566443
15-19 years of LPV new : 1.205956
20+ years of LPV new : 0.510167
0-4 years of LPV used : 0.147989
5-9 years of LPV used : 0.362701
10-14 years of LPV used : 1.006690
15-19 years of LPV used : 0.468745
20+ years of LPV used : 0.483618
```

LCV single groups results discover:

```
Data Mining Objective TWO result discover, 2017 LCV
Light commercial average age : 2.224930
0-4 years of LCV new : 0.312047
5-9 years of LCV new : 0.431164
10-14 years of LCV new : 0.592100
15-19 years of LCV new : 0.390933
20+ years of LCV new : 0.537158
0-4 years of LCV used : 0.254374
5-9 years of LCV used : 0.236564
10-14 years of LCV used : 0.359984
15-19 years of LCV used : 0.221876
20+ years of LCV used : 0.301918
```

The results show that the sum of single groups against the whole groups real values have small residuals that indicate for satisfy data mining objective two. In addition, the figures illustrated that the all single groups could be predicted for extract Co2 emission from the whole group.

In conclusion, since the pipeline fulfil all the datamining objectives and basing on these data generated from another iteration that the pipeline has a good robustly and stability for Data Mining.

8.5.2 Get a better result:

Set random seed back to 722 and set the solver to saga for run the pipeline again.

```
SEED = 722
# SEED = 123
random.seed(SEED)
np.random.seed(SEED)
```

```
def step_7_2_build_model(X_train, y_train):
    # regressor = Ridge(solver='sag')
    regressor = Ridge(solver='saga')
    regressor.fit(X_train,y_train)
    return regressor
```

First for testify it still fulfil the data mining objectives:

```
Data Mining Objective One
LPV R2 score = 0.998493, LCV R2 score = 0.990210

Data Mining Objective Two
2017 LPV real value = 8.07740664325, sum of predict values 8.246840
2017 LCV real value = 2.47222703164, sum of predict values 2.477284
```

```
Data Mining Objective One result discover, LPV
2003 year, real value = 7.375728, predict value = 7.406143
2009 year, real value = 7.626856, predict value = 7.533452
2011 year, real value = 7.575258, predict value = 7.416712
2004 year, real value = 7.671046, predict value = 7.519350
2015 year, real value = 7.662124, predict value = 7.675690
2013 year, real value = 7.415187, predict value = 7.534058
2000 year, real value = 0.000000, predict value = 0.041292
```

```
Data Mining Objective One result discover, LCV
2004 year, real value = 1.582260, predict value = 1.685134
2009 year, real value = 1.774602, predict value = 1.722853
2002 year, real value = 1.559698, predict value = 1.592112
2014 year, real value = 1.993417, predict value = 2.088093
2010 year, real value = 1.830283, predict value = 1.772734
2011 year, real value = 1.861729, predict value = 1.847744
2000 year, real value = 0.000000, predict value = 0.028322
```

```
Data Mining Objective TWO result discover, 2017 LPV
Light passenger average age : 7.875221
0-4 years of LPV new : 0.922703
5-9 years of LPV new : 1.423192
10-14 years of LPV new : 2.222785
15-19 years of LPV new : 1.136445
20+ years of LPV new : 0.411629
0-4 years of LPV used : 0.235795
5-9 years of LPV used : 0.722593
10-14 years of LPV used : 1.002473
15-19 years of LPV used : 0.550768
20+ years of LPV used : 0.381543
```

```
Data Mining Objective TWO result discover, 2017 LCV
Light commercial average age : 2.222410
0-4 years of LCV new : 0.273303
5-9 years of LCV new : 0.658742
10-14 years of LCV new : 0.052139
15-19 years of LCV new : 0.442853
20+ years of LCV new : 0.348813
0-4 years of LCV used : 0.143391
5-9 years of LCV used : 0.158744
10-14 years of LCV used : 0.205732
15-19 years of LCV used : 0.134159
20+ years of LCV used : 0.163688
```

We can find that using saga solver will still satisfy the data mining objective. The R2 score increase a little, from 0.998475, 0.986506 under sag to 0.998493, 0.990210.

Next, we discovering does the saga solver can explained the model more accuracy than sag solver?

LPV:

```
LPV Percentage
   0-4 years of LPV new: # of fleets: 284522, co2 percentage: 10.2410%
   5-9 years of LPV new: # of fleets: 254697, co2 percentage: 15.7958%
 10-14 years of LPV new: # of fleets: 453646, co2 percentage: 24.6704%
 15-19 years of LPV new: # of fleets: 251078, co2 percentage: 12.6133%
   20+ years of LPV new: # of fleets: 346148, co2 percentage: 4.5686%
  0-4 years of LPV used: # of fleets: 289449, co2 percentage: 2.6171%
  5-9 years of LPV used: # of fleets: 259108, co2 percentage: 8.0200%
10-14 years of LPV used: # of fleets: 461501, co2 percentage: 11.1263%
15-19 years of LPV used: # of fleets: 255426, co2 percentage: 6.1129%
  20+ years of LPV used: # of fleets: 352141, co2 percentage: 4.2347%
```

LCV:

```
LCV Percentage
   0-4 years of LCV new: # of fleets: 85203, co2 percentage: 10.5867%
   5-9 years of LCV new: # of fleets: 76272, co2 percentage: 25.5172%
 10-14 years of LCV new: # of fleets: 135849, co2 percentage: 2.0197%
 15-19 years of LCV new: # of fleets: 75188, co2 percentage: 17.1544%
   20+ years of LCV new: # of fleets: 103658, co2 percentage: 13.5117%
  0-4 years of LCV used: # of fleets: 19032, co2 percentage: 5.5544%
  5-9 years of LCV used: # of fleets: 17037, co2 percentage: 6.1491%
10-14 years of LCV used: # of fleets: 30346, co2 percentage: 7.9693%
15-19 years of LCV used: # of fleets: 16795, co2 percentage: 5.1968%
  20+ years of LCV used: # of fleets: 23155, co2 percentage: 6.3407%
```

For the LPV test set, saga solver has similar results with sag solver and for LCV data set it performance poor than sag solver, because it explained 5-9 and 15-19-years groups much, model have more weight on these two groups and it cannot reflect 10-14 years groups well comparing sag solver.

Finally, we could confirm this iteration results above is the optimal and the pipeline is robust and stable for various data. Since the models are reliable than SPSS, we need to applying other models or consider more on the data aspect at next coming iteration and research paper.

Reference:

1. World Meteorological Organization. (15 January 2020). “WMO confirms 2019 as second hottest year on record”. Retrieved from https://public.wmo.int/en/media/press-release/wmo-confirms-2019-second-hottest-year-record

2. Ministry of Transport. (05 August 2020).” Vehicle Fleet Statistics”. Retrieved from https://www.transport.govt.nz/mot-resources/vehicle-fleet-statistics/

"I acknowledge that the submitted work is my own original work in accordance with the University of Auckland guidelines and policies on academic integrity and copyright. (See: https://www.auckland.ac.nz/en/students/forms-policies-and-guidelines/student-policies-and-guidelines/academic-integrity-copyright.htmlLinks to an external site.).

I also acknowledge that I have appropriate permission to use the data that I have utilised in this project. (For example, if the data belongs to an organisation and the data has not been published in the public domain then the data must be approved by the rights holder.) This includes permission to upload the data file to Canvas. The University of Auckland bears no responsibility for the student's misuse of data."