Car Fuel Efficiency Analysis

Harrison Bailye

08/12/2021

Contents

Load in the data

Data from my Subaru Impreza was collected using an OBD reader and was exported into a csv file. This data will be used throughout the report to make meaningful insights about the fuel efficiency of the car.

```r
car <- read_csv("Driving.csv")
```

```
## Rows: 56 Columns: 23
```

```
## -- Column specification ----------------------------------------------------------
## Delimiter: ","
## chr (14): Driving Start Time, Driving Finish Time, Driving Time, Address, Di...
## dbl  (9): Fuel Consumed, Fuel Cost, Rapid Acc. Hard Count, Rapid Acc. Normal...
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Select data and method

It was decided that a ridge regression approach will be used to fit the model, where the fuel efficiency of the car will be the response variable. As fuel efficiency is a numeric variable, the model will be a regression model (outputs a value) not a classification model (classifies the result into a class).

EDA

Before a model can be constructed, it is important to perform explanatory data analysis (EDA) to ensure that the data is clean and in the correct form and also to investigate relationships between predictors and the response variable.

EDA 01: View the data

To begin with, we will view the data to see what needs to be fixed before working with the data.

```r
head(car)
```

```
## # A tibble: 6 x 23
##   ‘Driving Start T~ ‘Driving Finish~ ‘Driving Time‘ Address Distance ‘Max Speed‘
##   <chr>             <chr>            <chr>          <chr>   <chr>    <chr>
## 1 2021.11.01 19:17~ 2021.11.01 19:3~ 16m 17s        West T~ 9.49km   62.0km/h
## 2 2021.11.01 20:56~ 2021.11.01 21:1~ 18m 4s         Quebec~ 9.9km    62.0km/h
## 3 2021.11.04 11:15~ 2021.11.04 11:2~ 11m 39s        939 Ma~ 6.87km   62.0km/h
## 4 2021.11.04 12:04~ 2021.11.04 12:0~ 4m 49s         63 Bel~ 2.73km   58.0km/h
## 5 2021.11.04 13:01~ 2021.11.04 13:0~ 5m 30s         15 Win~ 2.73km   64.0km/h
## 6 2021.11.05 16:58~ 2021.11.05 17:2~ 25m 31s        161-16~ 12.84km  62.0km/h
## # ... with 17 more variables: Avr. Speed <chr>, Max RPM <chr>, Avr. RPM <chr>,
## #   Max Coolant Temp <chr>, Max Engine Oil Temp <chr>, Fuel effciency <chr>,
## #   Fuel Consumed <dbl>, Fuel Cost <dbl>, Rapid Acc. Hard Count <dbl>,
## #   Rapid Acc. Normal Count <dbl>, Rapid Decel. Hard Count <dbl>,
## #   Rapid Decel. Normal Count <dbl>, Speeding Hard count <dbl>,
## #   Speeding Normal count <dbl>, Idling time <dbl>, Safe Driving Score <chr>,
## #   Eco Driving Score <chr>
```

We can omit the missing data points as there are not many of them so omitting them won’t have much of an impact on the results.

```r
car <- car %>%
  na.omit()
```

EDA 02: Remove predictors

We saw that there were 22 predictors in the data set, however, not all of them are useful for the purpose of the task, so we will omit these predictors from the data set.

```r
car <- car %>%
  select(-c(`Driving Start Time`, `Driving Finish Time`, `Max Engine Oil Temp`,
            `Fuel Cost`, `Speeding Normal count`, `Speeding Hard count`,
            `Rapid Acc. Hard Count`, `Rapid Acc. Normal Count`, `Driving Time`))
head(car)
```

```
## # A tibble: 6 x 14
##   Address                  Distance ‘Max Speed‘ ‘Avr. Speed‘ ‘Max RPM‘ ‘Avr. RPM‘
##   <chr>                    <chr>    <chr>       <chr>        <chr>     <chr>
## 1 West Terrace Adelaide ~ 9.49km   62.0km/h    35.07km/h    3031.25r~ 1352.91rpm
## 2 Quebec Avenue Clapham ~ 9.9km    62.0km/h    32.91km/h    3281.25r~ 1396.97rpm
## 3 939 Marion Road Mitche~ 6.87km   62.0km/h    35.44km/h    2546.75r~ 1370.64rpm
## 4 63 Belair Road Kingswo~ 2.73km   58.0km/h    33.99km/h    2637.5rpm 1352.98rpm
## 5 15 Windermere Avenue C~ 2.73km   64.0km/h    29.85km/h    2806.25r~ 1333.85rpm
## 6 161-169 Esplanade Brig~ 12.84km  62.0km/h    30.22km/h    3771.75r~ 1311.62rpm
## # ... with 8 more variables: Max Coolant Temp <chr>, Fuel effciency <chr>,
## #   Fuel Consumed <dbl>, Rapid Decel. Hard Count <dbl>,
## #   Rapid Decel. Normal Count <dbl>, Idling time <dbl>,
## #   Safe Driving Score <chr>, Eco Driving Score <chr>
```

EDA 03: Clean columns

We will now clean the columns, converting them into the right form and removing the units of measurement for each data point.

```
# Remove units of measurement
car$Distance <- str_remove_all(car$Distance, "km")
car$`Avr. Speed` <- str_remove_all(car$`Avr. Speed`, "km/h")
car$`Max Speed` <- str_remove_all(car$`Max Speed`, "km/h")
car$`Avr. RPM` <- str_remove_all(car$`Avr. RPM`, "rpm")
car$`Max RPM` <- str_remove_all(car$`Max RPM`, "rpm")
car$`Max Coolant Temp` <- str_remove_all(car$`Max Coolant Temp`, "°C")
car$`Fuel effciency` <- str_remove_all(car$`Fuel effciency`, "L/100km")

# Convert to numeric
car$Distance <-  as.numeric(car$Distance)
car$`Avr. Speed` <- as.numeric(car$`Avr. Speed`)
car$`Max Speed` <- as.numeric(car$`Max Speed`)
car$`Avr. RPM` <- as.numeric(car$`Avr. RPM`)
car$`Max RPM` <- as.numeric(car$`Max RPM`)
car$`Max Coolant Temp` <- as.numeric(car$`Max Coolant Temp`)
car$`Fuel effciency` <- as.numeric(car$`Fuel effciency`)
car$`Safe Driving Score` <- as.numeric(car$`Safe Driving Score`)
car$`Eco Driving Score` <- as.numeric(car$`Eco Driving Score`)
```

EDA 04: Skim the data

we will skim the data to show many may rows we have in the data set, to see the type of each column and to get basic summary statistics for each column.

```
skimr::skim_without_charts(car)
```

Table 1: Data summary

Name	car
Number of rows	55
Number of columns	14
Column type frequency:	
character	1
numeric	13
Group variables	None

Variable type: character

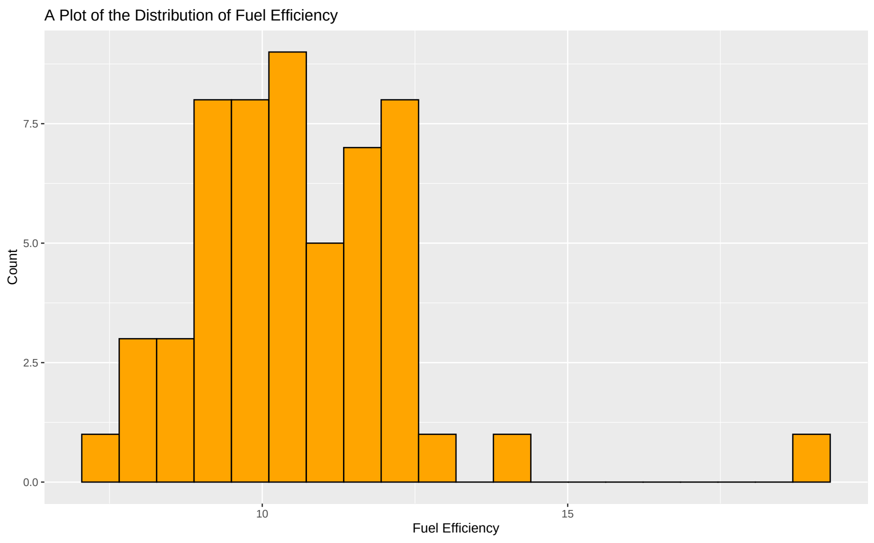
skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Address	0	1	26	50	0	36	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	
Distance	0	1	7.97	4.70	0.69	4.60	7.92	9.98	
Max Speed	0	1	62.51	6.20	46.00	60.00	62.00	64.00	
Avr. Speed	0	1	30.34	6.50	12.15	26.06	30.15	35.32	
Max RPM	0	1	2825.80	480.53	2118.75	2554.62	2700.00	3028.12	48
Avr. RPM	0	1	1322.22	114.05	1061.22	1254.06	1339.29	1396.69	15
Max Coolant Temp	0	1	94.53	5.57	68.00	95.00	96.00	97.00	1
Fuel effciency	0	1	10.59	1.85	7.59	9.37	10.47	11.61	
Fuel Consumed	0	1	0.80	0.41	0.13	0.52	0.84	1.00	
Rapid Decel. Hard Count	0	1	0.09	0.44	0.00	0.00	0.00	0.00	
Rapid Decel. Normal Count	0	1	1.44	1.62	0.00	0.00	1.00	2.00	
Idling time	0	1	192.76	132.72	19.00	98.50	174.00	238.50	7
Safe Driving Score	0	1	83.51	10.73	65.66	74.59	84.30	91.62	1
Eco Driving Score	0	1	84.42	8.52	67.67	77.10	84.77	89.91	1

EDA 05: Distribution of Response Variable

```r
car %>%
  ggplot(aes(`Fuel effciency`)) +
  geom_histogram(col = "black", fill = "orange", bins = 20) +
  xlab("Fuel Efficiency") +
  ylab("Count") +
  ggtitle("A Plot of the Distribution of Fuel Efficiency")
```



EDA 06: Relationship Between Predictors and Response Variable

Now we will investigate the relationships between all of the the predictors and the fuel efficiency.

```
car %>%
  select(-Address) %>%
  gather(-`Fuel effciency`, key = "var", value = "value") %>%
  ggplot(aes(value, `Fuel effciency`)) +
  geom_point() +
  facet_wrap(~var, scales = "free") +
  xlab("Value") +
  ylab("Fuel Efficiency") +
  ggtitle("A Plot of Fuel Efficiency against All Predictors")
```

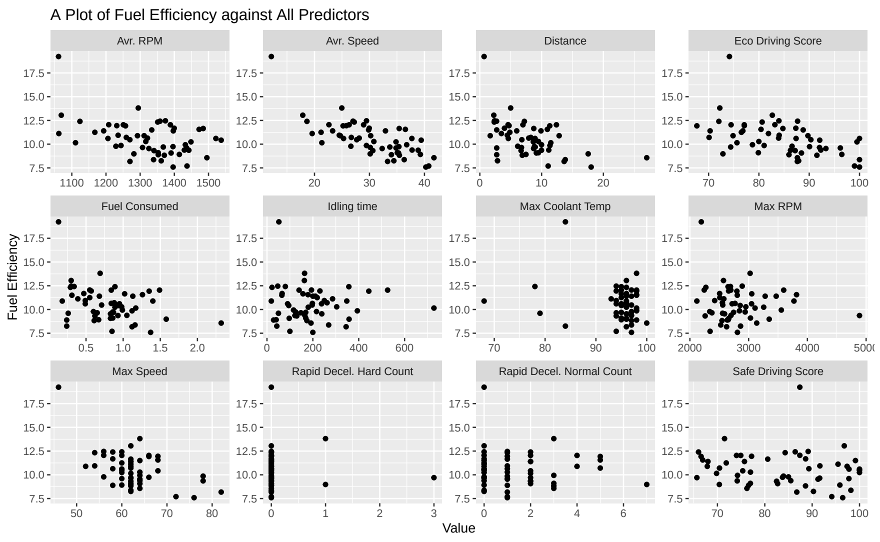


Figure 1: A Plot of Student Popularity against All Predictors

Preprocessing

Now that the data has be cleaned and relationships between the predictors and the response variable have been investigated, it is time to preprocess.

Preprocessing 01: Split into train and test

The data will be split into a training and testing set. The training set will be used to build the model while the testing set will test the predictions made by the model and will ensure that we don't under fit or over fit the model to the data.

```
car_split <- initial_split(car, strata = `Fuel effciency`)
```

```
## Warning: The number of observations in each quantile is below the recommended
## threshold of 20. Stratification will be done with 2 breaks instead.
```

```
car_train <- training(car_split)
car_test <- testing(car_split)
```

Preprocessing 02: Create the recipe

We will create dummy variables for all the nomial predictors, which in this case is the address.

```
car_recipe <-
  recipe(`Fuel effciency` ~., data = car_train) %>%
  step_dummy(all_nominal()) %>%
  step_normalize(all_predictors()) %>%
  step_center(all_predictors()) %>%
  step_scale(all_predictors())
```

Modelling

Modelling 01: Create the model

We are using ridge regression and we will be tuning the penalty term. For ridge regression the mixture term equals 0, if we were to do lasso regression we would set it equal to 1.

```
car_model <-
  linear_reg(mixture = 0, penalty = tune()) %>%
  set_engine("glmnet")
```

Modelling 02: Create the workflow

Using the recipe and model, create a workflow.

```
car_wf <-
  workflow() %>%
  add_model(car_model) %>%
  add_recipe(car_recipe)
```

Modelling 03: Tune the Penalty Term

A k-cross validation approach will be used for tuning the penalty term. We will use 10 folds.

```
set.seed(2021)
car_cv <- vfold_cv(car_train, strata = `Fuel effciency`)
```

```
## Warning: The number of observations in each quantile is below the recommended
## threshold of 20. Stratification will be done with 2 breaks instead.
```

```
car_cv
```

```
## #  10-fold cross-validation using stratification
## # A tibble: 10 x 2
##    splits         id
##    <list>         <chr>
##  1 <split [36/5]> Fold01
##  2 <split [37/4]> Fold02
##  3 <split [37/4]> Fold03
##  4 <split [37/4]> Fold04
##  5 <split [37/4]> Fold05
##  6 <split [37/4]> Fold06
##  7 <split [37/4]> Fold07
##  8 <split [37/4]> Fold08
##  9 <split [37/4]> Fold09
## 10 <split [37/4]> Fold10
```

Create a tuning grid for the penalty term.

```
car_grid <- grid_regular(penalty(), levels = 10)
```

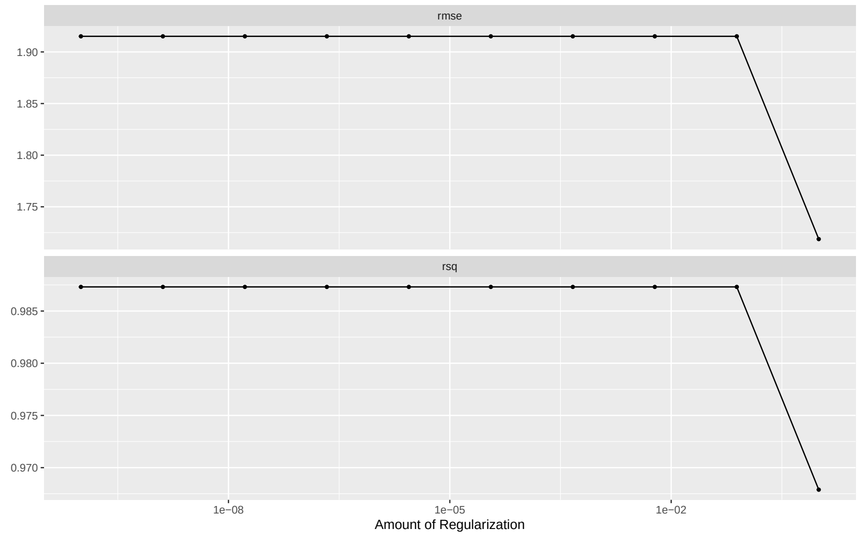
Tune the penalty term hyper parameter.

```
doParallel::registerDoParallel()
car_tune <- tune_grid(
  object = car_wf,
  resamples = car_cv,
  grid = car_grid
)
```

Modelling 04: Tuning Results

The plot below is a visual representation of the parameter that we are tuning.

```
car_tune %>% autoplot()
```



Show the best penalty terms based on the residual mean square error (RMSE).

```
show_best(car_tune, metric = "rmse")
```

```
## # A tibble: 5 x 7
##          penalty .metric .estimator  mean     n std_err .config
##            <dbl> <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1 1              rmse    standard    1.72     9   0.408 Preprocessor1_Model10
## 2 0.0000000001   rmse    standard    1.92     9   0.386 Preprocessor1_Model01
## 3 0.00000000129 rmse    standard    1.92     9   0.386 Preprocessor1_Model02
## 4 0.0000000167   rmse    standard    1.92     9   0.386 Preprocessor1_Model03
## 5 0.000000215    rmse    standard    1.92     9   0.386 Preprocessor1_Model04
```

Modelling 05: Finalise the workflow

Finalise the workflow using the best penalty parameter

```
car_wf <- car_wf %>%
          finalize_workflow(select_best(car_tune, metric = 'rmse'))
```

Modelling 06: Fit the model

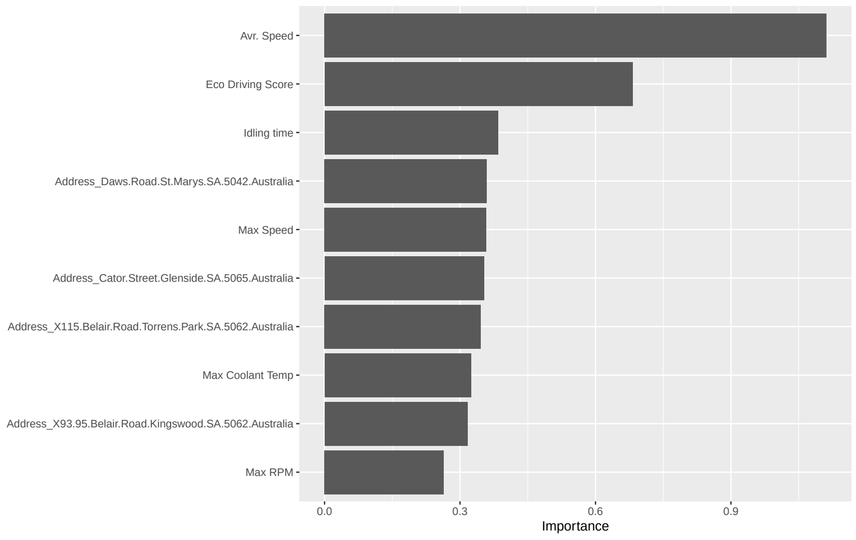
We will fit the training data to the finalized workflow to create model 1.

```
M1 <- car_wf %>% fit(car_train)
```

Modelling 06: Look at the predictors

Look at important variables in the model to see what predictors have a large impact on the fuel efficiency of the car.

```
M1 %>%
  extract_fit_parsnip() %>%
  vip()
```



From this plot we can see that the average speed is the predictor that has the largest impact on the fuel efficiency. Other key predictors include distnace and eco driving score.

Prediction

Prediction 01: Last Fit in Workflow

Put the last fit into the workflow and collect the metrics

```
car_wf %>%
  last_fit(car_split) %>%
  collect_metrics()
```

```
## ! train/test split: preprocessor 1/1, model 1/1 (predictions): There are new levels in a fac...
```

```
## # A tibble: 2 x 4
##   .metric .estimator .estimate .config
##   <chr>   <chr>          <dbl> <chr>
## 1 rmse    standard       0.941 Preprocessor1_Model1
## 2 rsq     standard       0.267 Preprocessor1_Model1
```

Prediction 02: Predict using Training Data

We will make predictions using the training data.

```
car_train %>%
  add_column(
    M1 %>%
    predict(new_data = car_train)) %>%
  ungroup() %>%
  rmse(`Fuel effciency`, .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       0.837
```

Prediction 03: Predict using Testing Data

We will make predictions using the testing data.

```
car_test %>%
  add_column(
    M1 %>%
    predict(new_data = car_test)) %>%
  ungroup() %>%
  rmse(`Fuel effciency`, .pred)
```

```
## Warning: There are new levels in a factor: 16 Leonard Terrace Torrens Park
## SA 5062 Australia, 203-206 Esplanade Seacliff SA 5049 Australia, 307 Young
## Street Wayville SA 5034 Australia, 311 Young Street Wayville SA 5034 Australia,
## 12 Hill Avenue Cumberland Park SA 5041 Australia, South Road Edwardstown SA
## 5039 Australia, 71 Devereux Road Linden Park SA 5065 Australia, 237 Esplanade
## Seacliff SA 5049 Australia, 76 Devereux Road Beaumont SA 5066 Australia
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       0.941
```

Final Fit

Fit the final workflow to all the training data and then predict for the test data. This is the final model, the single model that will be used for prediction

```
car_fit <- car_wf %>% last_fit(split = car_split)
```

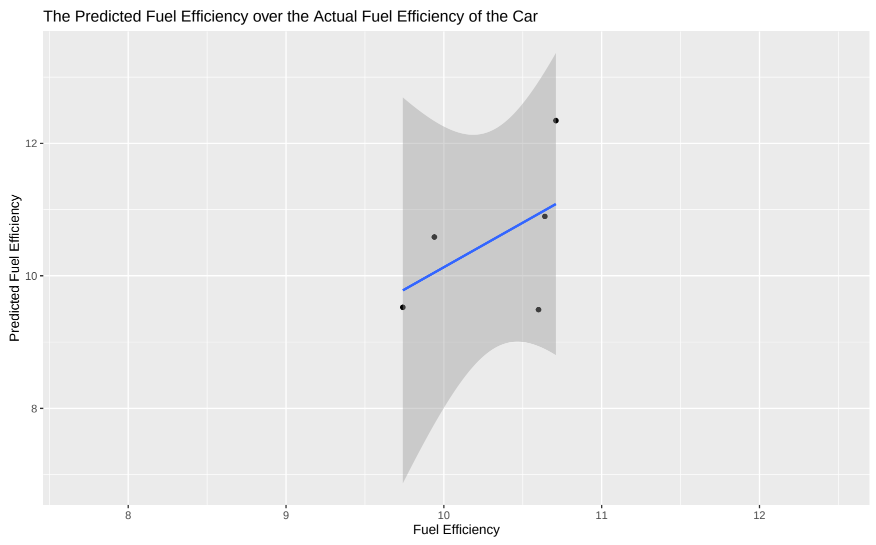
```
## ! train/test split: preprocessor 1/1, model 1/1 (predictions): There are new levels in a fac...
```

```
car_fit %>%
  collect_predictions() %>%
  ggplot(aes(`Fuel effciency`, .pred)) +
  geom_point() +
  xlab("Fuel Efficiency") +
  ylab("Predicted Fuel Efficiency") +
  ggtitle("The Predicted Fuel Efficiency over the Actual Fuel Efficiency of the Car") +
  geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula 'y ~ x'

## Warning: Removed 9 rows containing non-finite values (stat_smooth).

## Warning: Removed 9 rows containing missing values (geom_point).
```


The Predicted Fuel Efficiency over the Actual Fuel Efficiency of the Car
Predicted Fuel Efficiency
Fuel Efficiency