美國二手車
市場價格分析

每個人心中都有一輛夢幻車

Group8

組員：
M104020020 王薇琳 M104020034 徐宇欣 M104020037 林元雅 M104020053 莊明輯
M104020059 蔡秉辰 M104111057 高珮瑜 M114020037 曾彗瑀

AGENDA

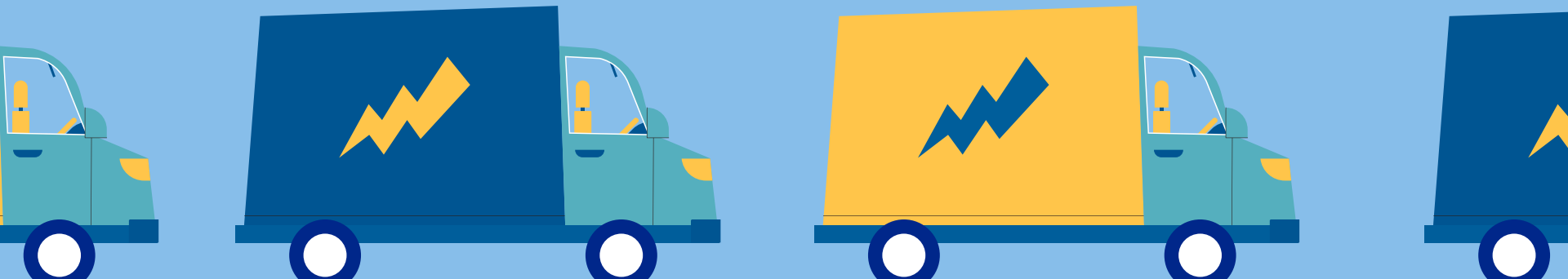
PART ONE
期中回顧

PART TWO
模型訓練

PART THREE
結果與討論

01

期中回顧



資料欄位介紹

變數名稱	說明	資料型態
year	製造年份	int
make	品牌	chr
model	型號	chr
body	車型	chr
transmission	變速箱	chr
state	州	chr

變數名稱	說明	資料型態
condition	車況	num
odometer	里程數	int
color	顏色	chr
interior	內裝顏色	chr
sellingprice	銷售金額	int
saleyear	售出年份	int

資料預處理

刪除NA值

- 刪除有空值的資料

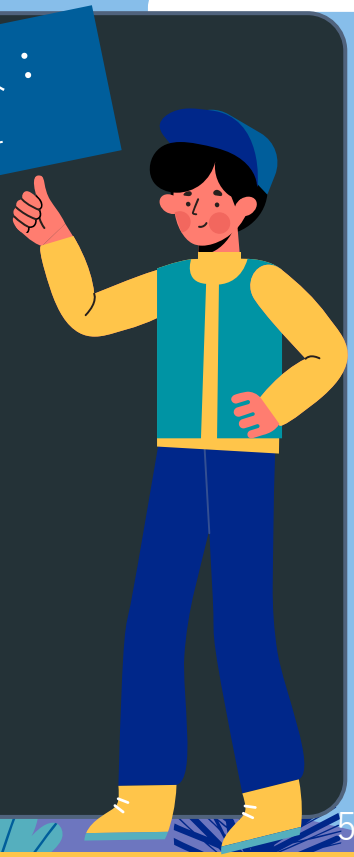
增加欄位

- 增加vehicleage（車齡）欄位，以此紀錄車齡

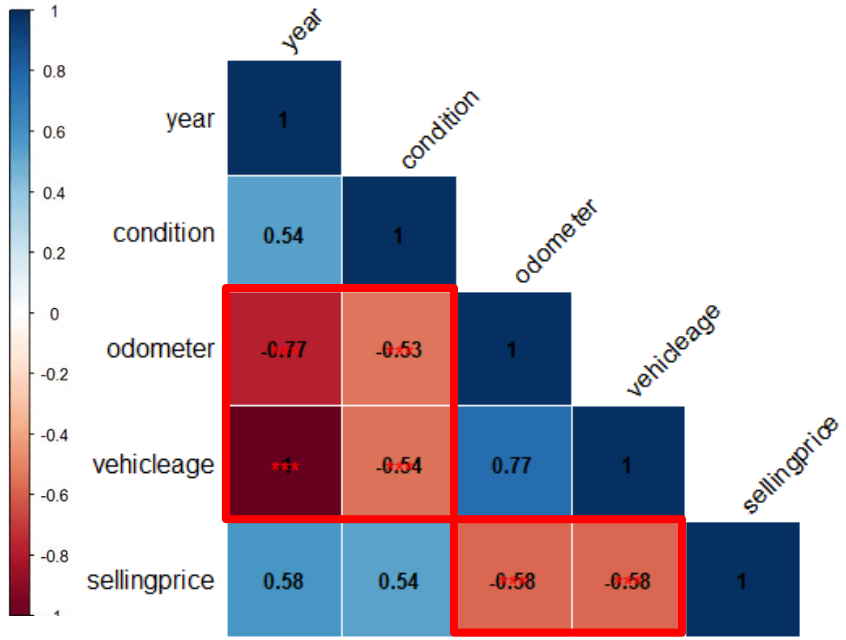
計算方式：以salesyear（售出年份）-year（製造年份）

刪除欄位

- 刪除未使用的欄位，例如：trim, vin, salesyear , year, transmission



變數間的關係



	year	condition	odometer	vehicleage	sellingprice
year	1.0000000	0.5439122	-0.7709191	-0.9975450	0.5805712
condition	0.5439122	1.0000000	-0.5341513	-0.5420983	0.5361078
odometer	-0.7709191	-0.5341513	1.0000000	0.7695592	-0.5774821
vehicleage	-0.9975450	-0.5420983	0.7695592	1.0000000	-0.5788867
sellingprice	0.5805712	0.5361078	-0.5774821	-0.5788867	1.0000000

發現

由相關矩陣得知odometer(里程數)、vehicleage(車齡)兩變數與 sellingprice(銷售金額)有顯著關係

02

模型訓練

資料處理

資料正規化

- 將數值類變數正規化：condition（車況）、odometer（里程數）、vehicleage（車齡）

condition	odometer	color	interior	vehicleage
0.750	0.037181074	gray	black	0.00000000
0.600	0.036974074	red	black	0.00000000
0.625	0.038529077	white	black	0.04166667
0.750	0.038665077	silver	black	0.00000000
0.675	0.043261087	black	tan	0.00000000

模型訓練

Forward Selection

變數挑選

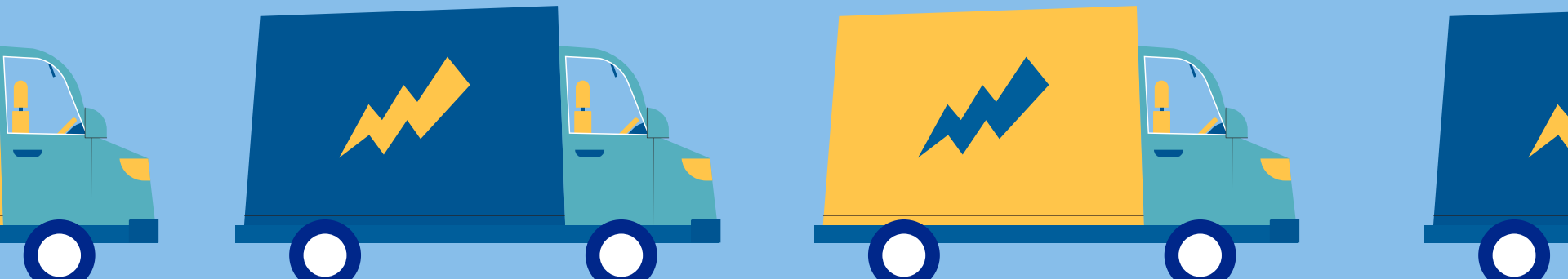
- 結果顯示：模型列入以下變數時，表現較佳

```
> # forward
> forward <- step(min.model, direction='forward', scope=formula(biggest), trace=0)
>
> summary(forward)

Call:
lm(formula = sellingprice ~ odometer + body + condition + vehicleage +
    model + interior + state + color, data = train_d)
```

model

線性回歸



模型訓練

線性回歸

```
Call:
lm(formula = sellingprice ~ ., data = train_d)

Residuals:
   Min     1Q Median     3Q    Max
-19309  -1293   -253    987  96979

Coefficients: (8 not defined because of singularities)
                   Estimate Std. Error  t value Pr(>|t|)
(Intercept)        8705.467   1499.468    5.806 6.44e-09 ***
bodyCoupe           203.448    136.037    1.496 0.134778
bodyextended cab   4987.490    660.364    7.553 4.32e-14 ***
bodyExtended Cab   4191.837    348.513   12.028  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2580 on 63772 degrees of freedom
Multiple R-squared:  0.8411,    Adjusted R-squared:  0.8408
F-statistic:  3590 on 94 and 63772 DF,  p-value: < 2.2e-16
```

MAE(train)

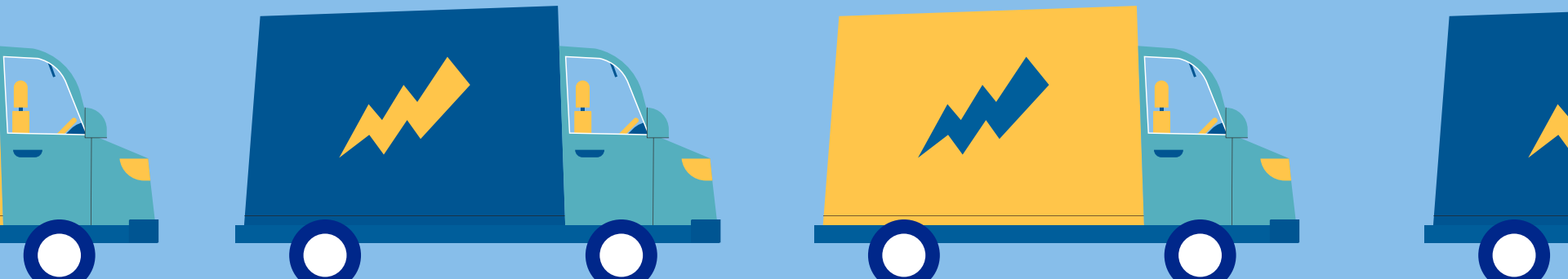
1698.165

MAE(test)

1714.95

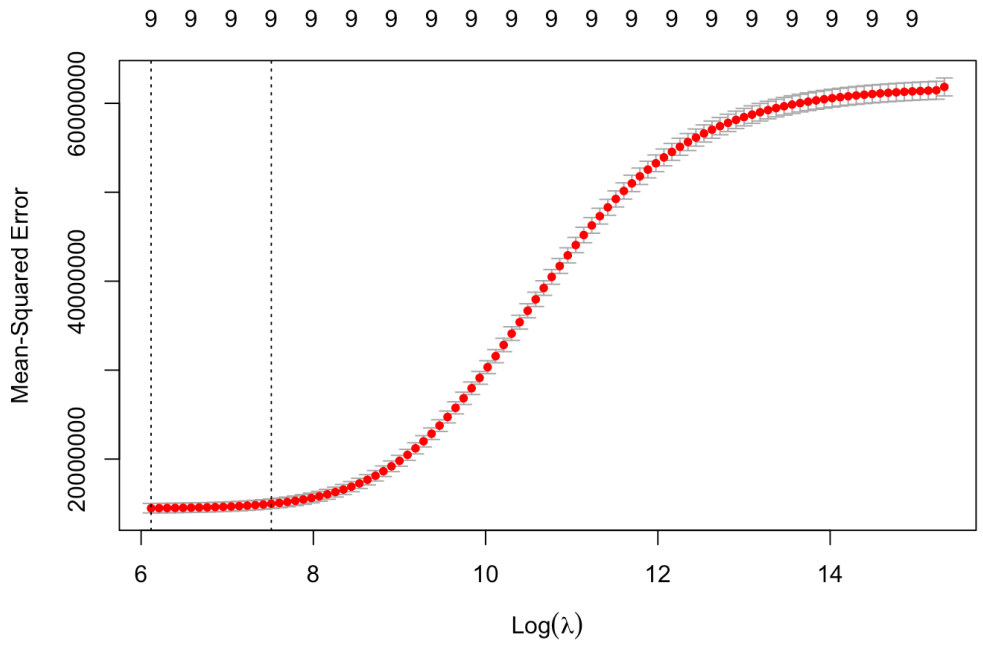
model

Ridge



模型訓練

Ridge-找出最佳lambda

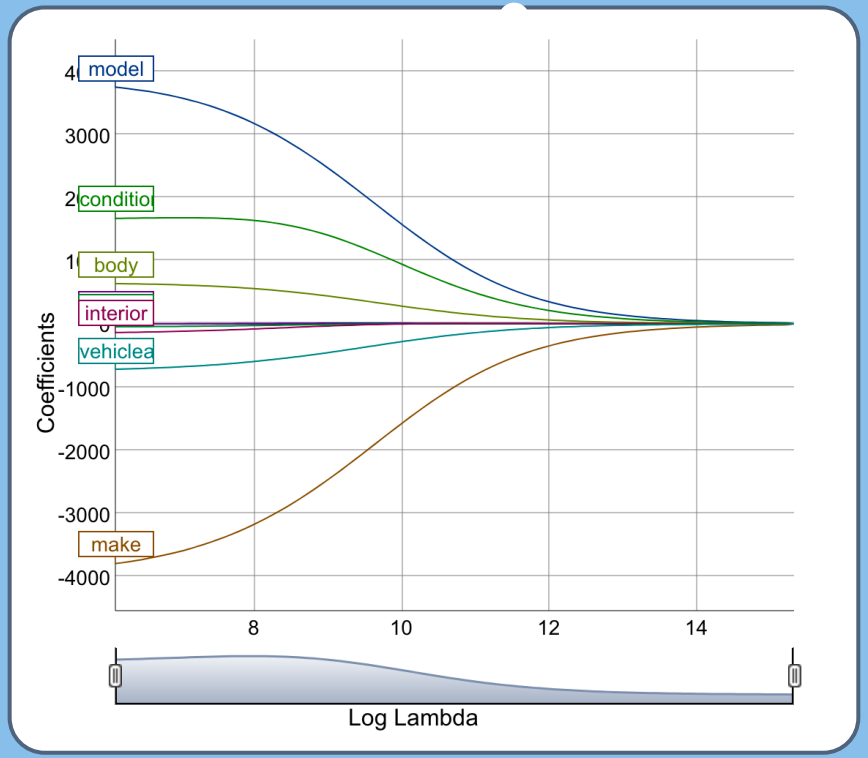



```r
###ridge
ridge <- cv.glmnet(x=x, y=y_train, alpha=0)
plot(ridge)
ridgeSum<-summary(ridge)
ridgeSum
#find coefficients of best model
best_lambda <- ridge$lambda.min
best_lambda #find optimal lambda value that
```

```
> best_lambda <- ridge$lambda.min
> best_lambda #find optimal lambda
[1] 452.976
```

模型訓練

Ridge-變數挑選



模型訓練

Ridge-結果

```
#use fitted best model to make train predictions
predictions_train <- predict(ridge, s = best_lambda, newx = x)

#find SST and SSE
sst <- sum((y_train - mean(y_train))^2) #Sum of Squares Total,
sse <- sum((predictions_train - y_train)^2) #Sum of Squares Err
#find R-Squared
rsq <- 1 - sse/sst
rsq
y_train_RMSE = sqrt(sse/nrow(df2))
y_train_RMSE
MAE(predict(ridge,x),y_train)
MAE(predict(ridge,x_test),y_test)
```

```
> rsq <- 1 - sse/sst
> rsq
[1] 0.766346
> y_train_RMSE = sqrt(sse/nrow(df2))
> y_train_RMSE
[1] 3180.563
> MAE(predict(ridge,x),y_train)
[1] 2678.924
> MAE(predict(ridge,x_test),y_test)
[1] 2668.946
```

MAE(train)

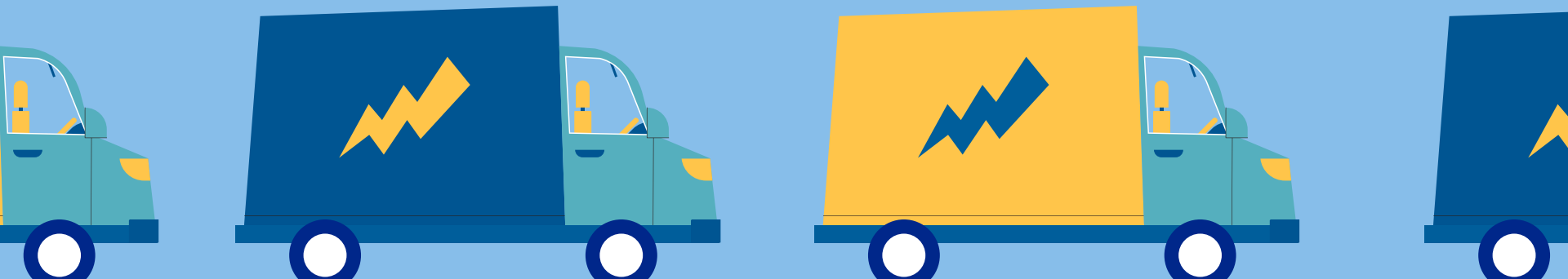
2678.942

MAE(test)

2668.946

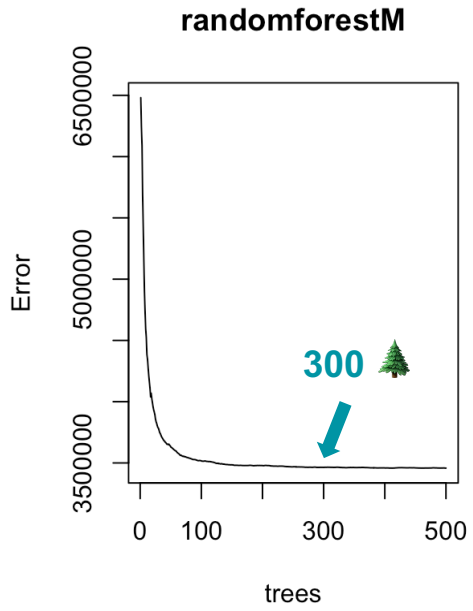
model

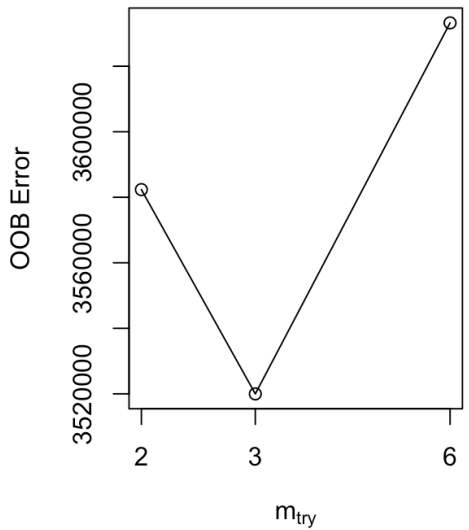
Random Forest



模型訓練

Random Forest-調參






```
mtry = 3  OOB error = 3519991
Searching left ...
mtry = 2        OOB error = 3582340
-0.01771273 0.05
Searching right ...
mtry = 6        OOB error = 3633226
-0.03216897 0.05
```

模型訓練

Random Forest-最終模型

```
> final_randomforest

Call:
 randomForest(formula = sellingprice ~ ., data = train_d, mtry = 3,      ntree = 300, importance = TRUE, do.trace = 100)
               Type of random forest: regression
                     Number of trees: 300
No. of variables tried at each split: 3

          Mean of squared residuals: 3509805
                    % Var explained: 91.61
```

MAE(train)

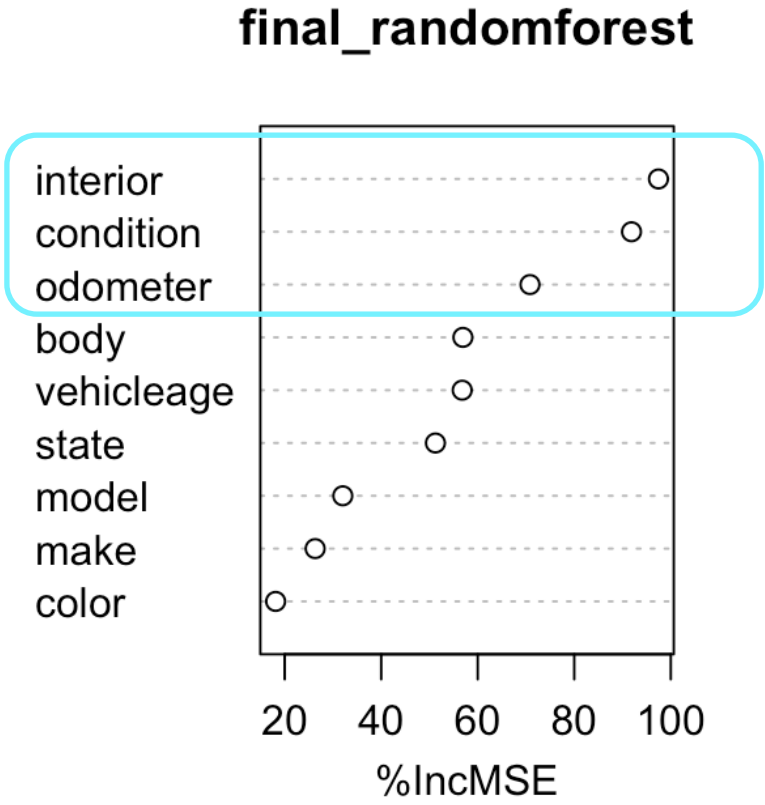
615.9689

MAE(test)

1168.919

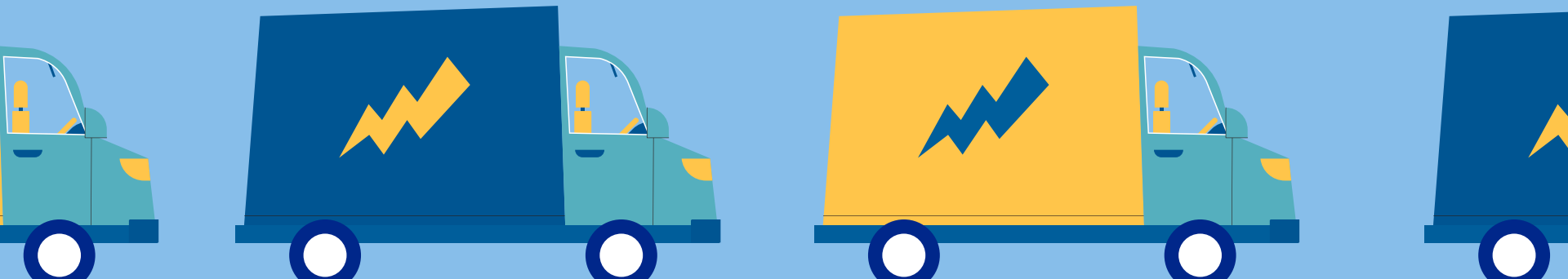
模型訓練

Random Forest-變數重要性



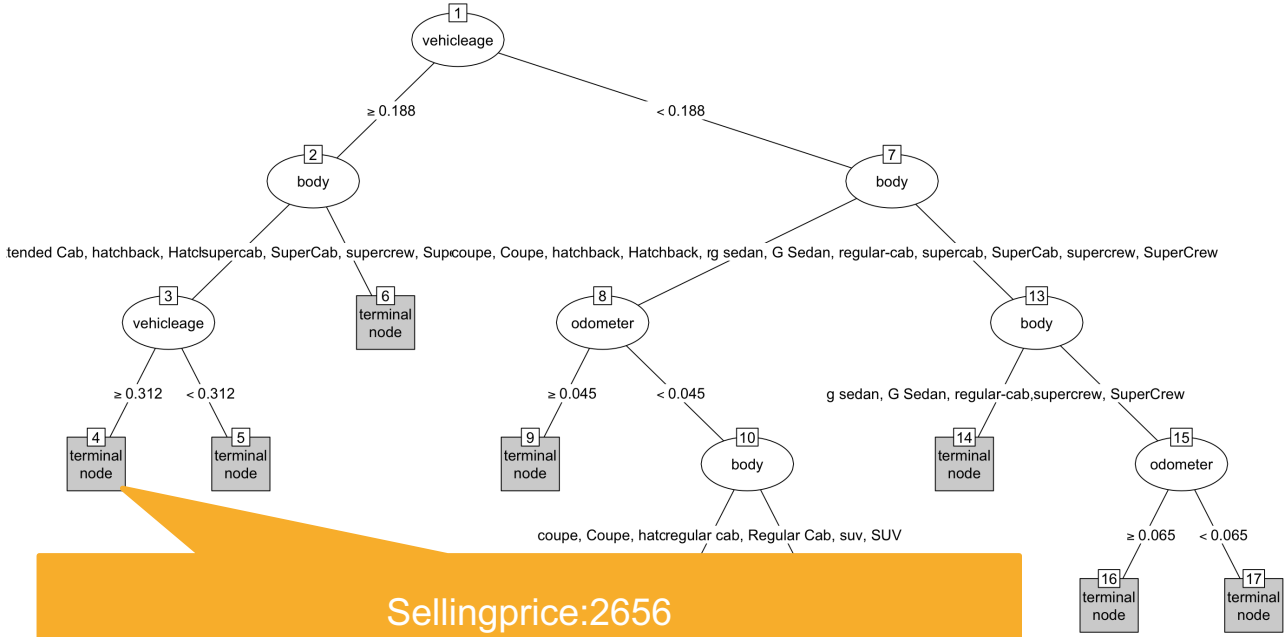
model

決策樹



模型訓練

決策樹



Sellingprice:2656

Body:coupe, extended cab, hatchback, sedan, wagon

MAE(train)

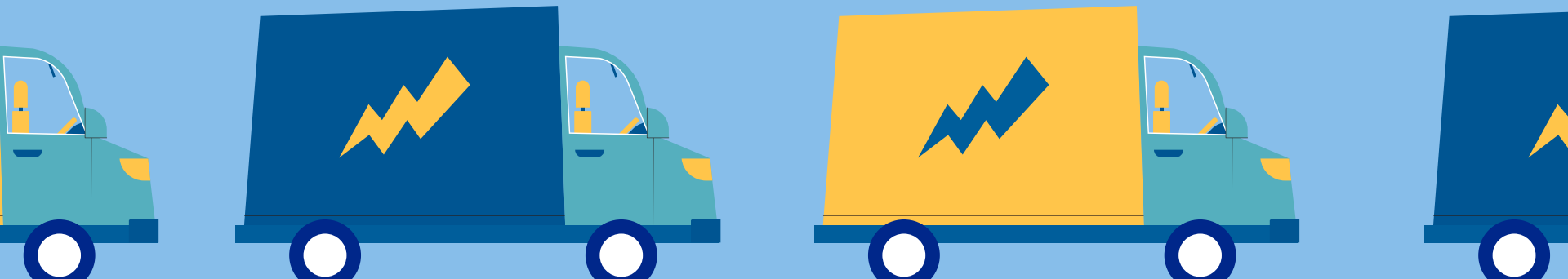
2214.98

MAE(test)

2233.275

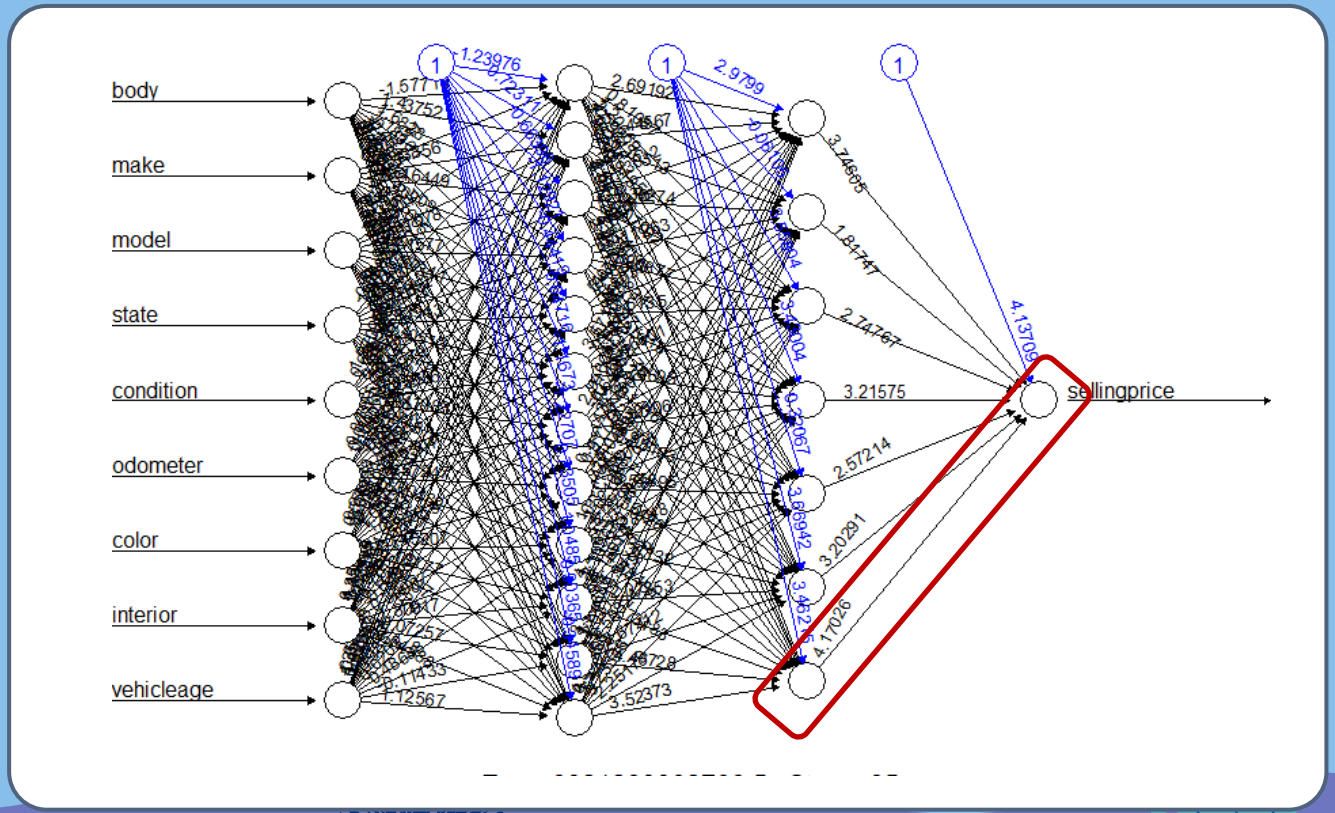
model

Deep Neural Network



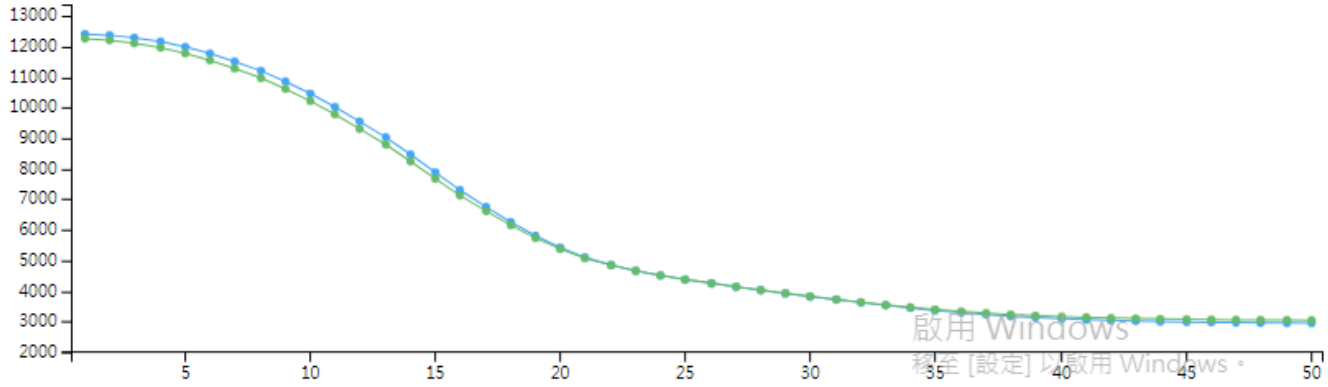
模型
訓練

Deep Neural Network



模型
訓練

Deep Neural Network



MAE(train)

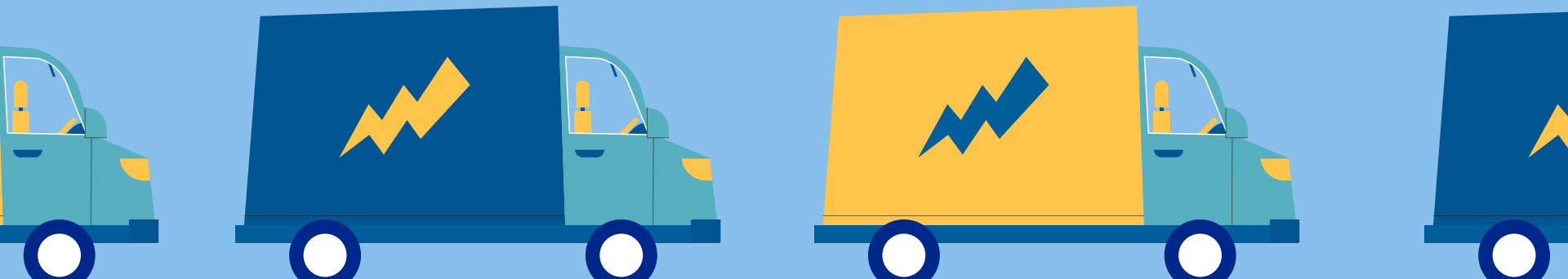
2960

MAE(test)

2932

model

XGBoost



模型訓練

XGBoost-Preprocess

```
#1.將Dataframe轉換為 xgboost 的稀疏矩陣
require(xgboost)
train_d<-catto_freq(train_d)
test_d<-catto_freq(test_d)
dtrain = xgb.DMatrix(data = as.matrix(train_d[,-10]),
                     label = train_d$sellingprice)
dtest = xgb.DMatrix(data = as.matrix(test_d[,-10]),
                    label = test_d$sellingprice)
```

模型訓練

XGBoost-調參

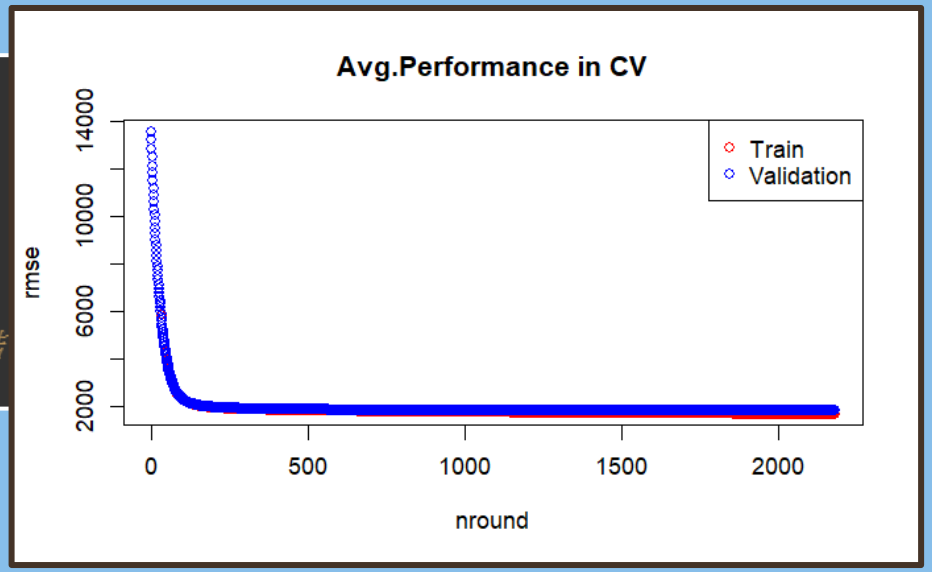
```
# 2.設定xgb.params，也就是 xgboost 裡面的參數
xgb.params = list(
  #col的抽樣比例，越高表示每棵樹使用的col越多，會增加每棵小樹的複雜度
  colsample_bytree = 0.6,
  # row的抽樣比例，越高表示每棵樹使用的col越多，會增加每棵小樹的複雜度
  subsample = 0.6,
  booster = "gbtree",
  # 樹的最大深度，越高表示模型可以長得越深，模型複雜度越高
  max_depth = 4,
  # boosting會增加被分錯的資料權重，而此參數是讓權重不會增加的那麼快，因此越大會讓模型愈保守
  eta = 0.03,
  # 或用'mae'也可以
  eval_metric = "rmse",
  objective = "reg:linear",
  # 越大，模型會越保守，相對的模型複雜度比較低
  gamma = 0)
```

模型訓練

XGBoost - CV

```
# 3. 使用xgb.cv()，tune 出最佳的決策樹數量
cv.model = xgb.cv(
  params = xgb.params,
  data = dtrain,
  nfold = 5,      # 5-fold cv
  nrounds=4000,    # 測試各個樹總數下的模型
  early_stopping_rounds = 30,
  print_every_n = 20 # 每20個單位才顯示一次結
)
```

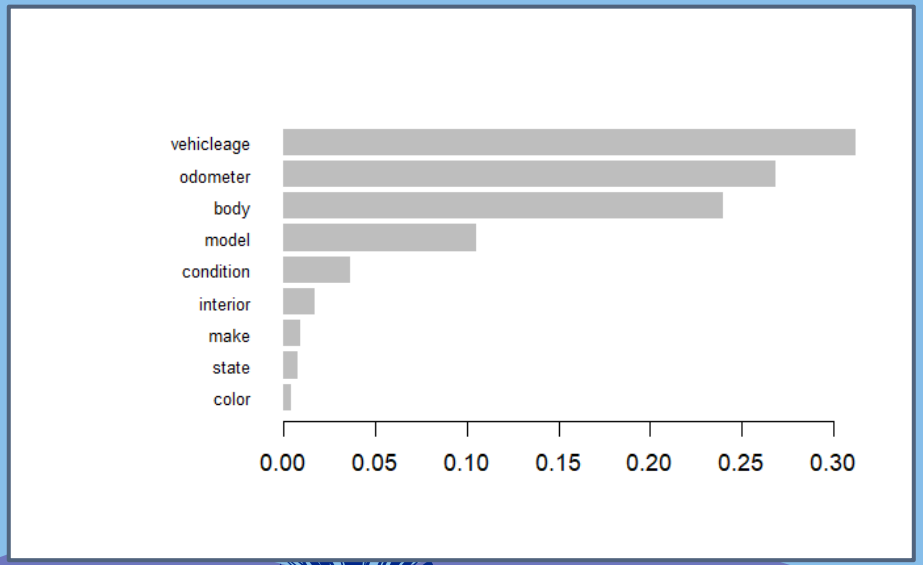
```
> best.nrounds = cv.model$best_iteration
> best.nrounds
[1] 2151
```



模型訓練

XGBoost – 建模

```
# 4. 用xgb.train()建立模型
xgb.model = xgb.train(paras = xgb.params,
                      data = dtrain,
                      nrounds = best.nrounds)
```



MAE(train)

448.402

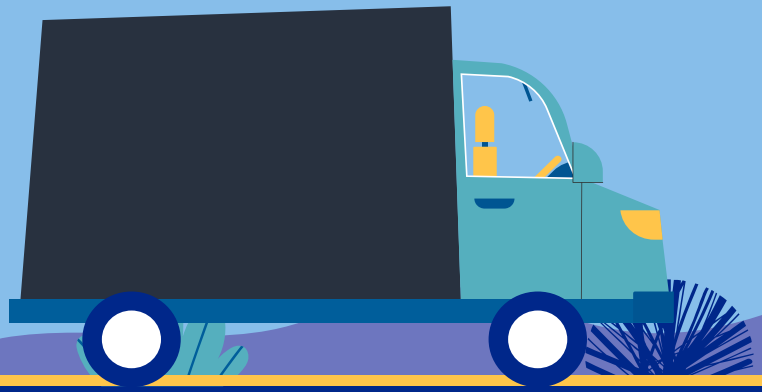
MAE(test)

1271.026

	rmse
Train	1694.124
Test	1925.386

03

結果與討論



模型比較

	MAE(train)	MAE(test)
線性回歸	1698.165	1714.95
Ridge	2678.924	2668.946
隨機森林	615.968	1168.919
決策樹	2214.98	2233.275
DNN	2960	2932
XGBoost	448.402	1271.026

重要變數

	重要變數1	重要變數2	重要變數3
線性回歸	odometer	vehicleage	condition
Ridge	condition	model	make
隨機森林	interior	condition	odometer
決策樹	body	odometer	model
DNN	vehicleage	body	condition
XGBoost	vehicleage	odometer	body

什麼變數影響二手車售價?

- 里程數 odometer
- 車況 condition
- 車齡 vehicleage
- 車型 body

研究限制

類別型變數種類過多，使用編碼處理效果不佳

變數不足，而因資料範圍受限，難以找到相關外部資訊

分工職掌

- 資管碩二 王薇琳：簡報製作、分析 、建模
- 資管碩二 徐宇欣：簡報製作、分析 、建模
- 資管碩二 林元雅：簡報製作、分析 、建模
- 資管碩二 莊明輯：簡報製作、分析 、建模
- 資管碩二 蔡秉辰：簡報製作、分析 、建模
- 企管碩二 高珮瑜：簡報製作、分析 、建模
- 資管碩一 曾彗瑀：簡報製作、分析 、建模

Thank for your attention

GROUP 8