

HWDiamond Price

高嘉妤、柯堯城、吳承恩、趙友誠

2024-11-25

Table of contents

0. 資料簡介	1
1.Data Preprocessing	2
2.Data visualization for exploratory data analysis	4
典型相關分析 (CCA)	11
3.Construct a predictive model for price	12
Data splitting	12
Linear regression model	12
randomForest 參數微調	16

(Kaggle)Diamonds Prices 2022

```
library(readr)
library(psych)
library(Hmisc)
library(DataExplorer)
library(ggplot2)
library(MASS)
library(car)
library(stargazer)
data <- data.table::fread("Diamonds Prices2022.csv")[,-1]
```

0. 資料簡介

Dimension of the Data : **53943 samples x 10 columns**

Variables	Explanation	remark
carat	克拉 (重量)	連續變數 (公克)
cut	切工	類別變數, Fair,Good,Ideal,Premium,Very Good
color	顏色	類別變數, D,E,F,G,H,I,J 無色(D~F), 近乎無色(G~J)
clarity	淨度	類別變數, IF: 內部無暇,VVS1: 極輕微瑕,VS1: 輕微內含物 1,VS2: 輕微內含物 2,SI1: 微內含物 1,SI2: 微內含物 2,I1: 內含物
depth	深度	連續變數 (mm)
table	檯面尺寸	連續變數
price	價格	連續變數

Variables	Explanation	remark
x	鑽石的長	連續變數 (mm)
y	鑽石的寬	連續變數 (mm)
z	鑽石的高	連續變數 (mm)

1. Data Preprocessing

```
latexit(describe(data, title="Diamond Price Dataset"), title="", file="")
```

```

z
  n    missing   distinct   Info   Mean   Gmd   .05   .10   .25   .50   .75   .90   .95
53943     0       375        1  3.539  0.7901  2.65  2.69  2.91  3.53  4.04  4.52  4.73

lowest : 0    1.07 1.41 1.53 2.06, highest: 6.43 6.72 6.98 8.06 31.8

```

由 describe 可以發現 x,y,z(長、寬、高) 當中出現 0，這是不合理的量測數值，因此將其作為遺失值處理。

經檢查，此資料有 20 筆資料的長或寬或高為 0，與五萬多的樣本數相比算小，因此刪除。

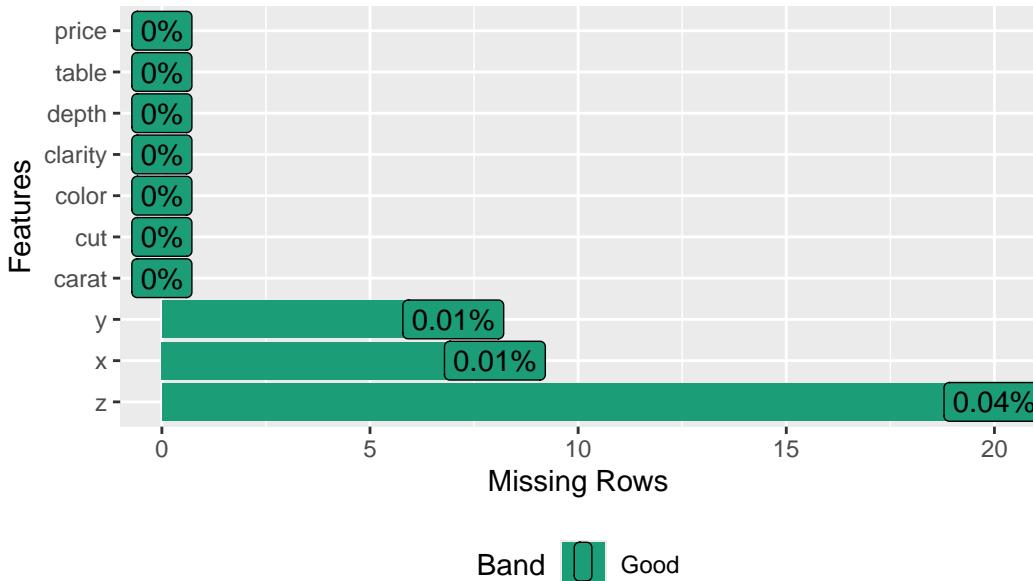
此外還發現反應變數”價格”有離群值，在後續使用模型建立預測模式時需多加考量。

```

data$x[data$x==0] <- NA
data$y[data$y==0] <- NA
data$z[data$z==0] <- NA
DataExplorer::plot_missing(
  data,
  title = "Fig.1 missing proportion before removing NA")

```

Fig.1 missing proportion before removing NA



```
data <- data[!(is.na(data$x) | is.na(data$y) | is.na(data$z)),]
```

我們認為僅僅觀察單一的長、寬、高並不具有任何意義，因此建立新變數體積 = $x \times y \times z$ 。

此外，一些類別型變數以 ordinal 的方式處理，賦予其從 1 開始的整數。

Cut(切工) 從 Fair 到 Very Good 分成 1-5 級

Color(鑽石顏色) 按照 D-J 排序為 1-7

Clarity(鑽石純淨程度) 按照 I1-IF 排序成 1-8

```

# 定義體積 = x * y * z
data$volume <- data$x * data$y * data$z
data$x <- NULL
data$y <- NULL
data$z <- NULL
# 發現體積中有離群值，故將其刪除
data <- data[data$volume < 800]
# 定義類別順序
levelcut <- c("Fair", "Good", "Ideal", "Premium", "Very Good")

```

```

levelcolor <- c("D", "E", "F", "G", "H", "I", "J")
levelclarity <- c("I1", "SI2", "SI1", "VS2", "VS1", "VVS2", "VVS1", "IF")

# 使用 match 進行編碼
data$cut <- match(data$cut, levelcut)
data$color <- match(data$color, levelcolor)
data$clarity <- match(data$clarity, levelclarity)

```

2.Data visualization for exploratory data analysis

從圖 2 的 Spearman 相關係數可以發現，鑽石的克拉數與體積幾乎與價格是正比，在後續分析的時候可以稍微注意，此外並無其他變數有特別的相關性。

```

corrplot::corrplot(cor(data, method = 'spearman'),
                   method = 'number',
                   mar = c(1, 0, 0, 0))
mtext("Fig.2 Spearman Correlation", side = 1, line = 4, cex = 1)

```

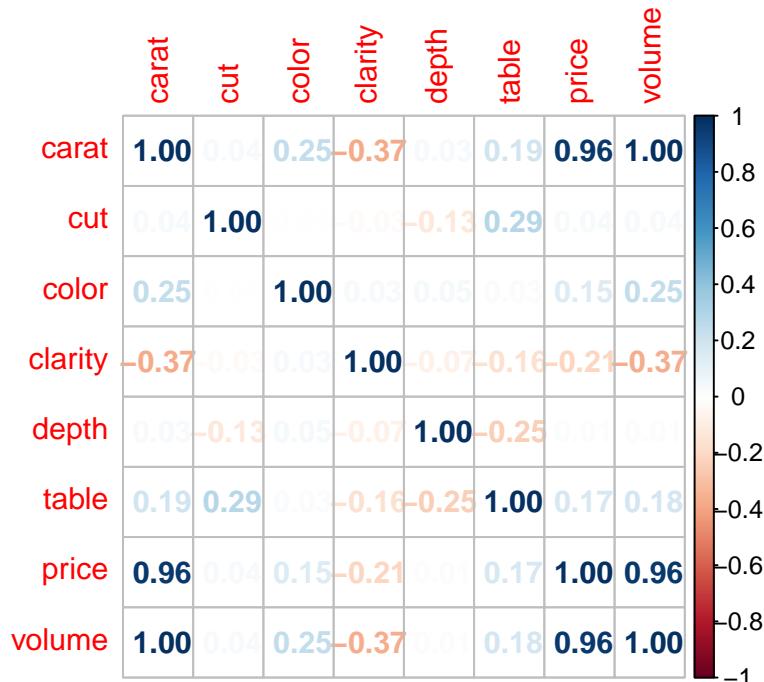


Fig.2 Spearman Correlation

```

# 克拉對價格（加切工）
ggplot(data, aes(x = carat,
                  y = price,
                  color = factor(cut,
                                 levels = c("1", "2", "3", "4", "5"),
                                 labels = c("Fair", "Good", "Ideal", "Premium", "Very Good")))) +
  geom_point(alpha = 0.6) +
  labs(title = "Fig.3 Carat vs Price by Cut",
       x = "Carat",
       y = "Price",
       color = "Cut") +
  scale_color_manual(values = c( "Fair" = "red", "Good" = "blue", "Ideal" = "green",

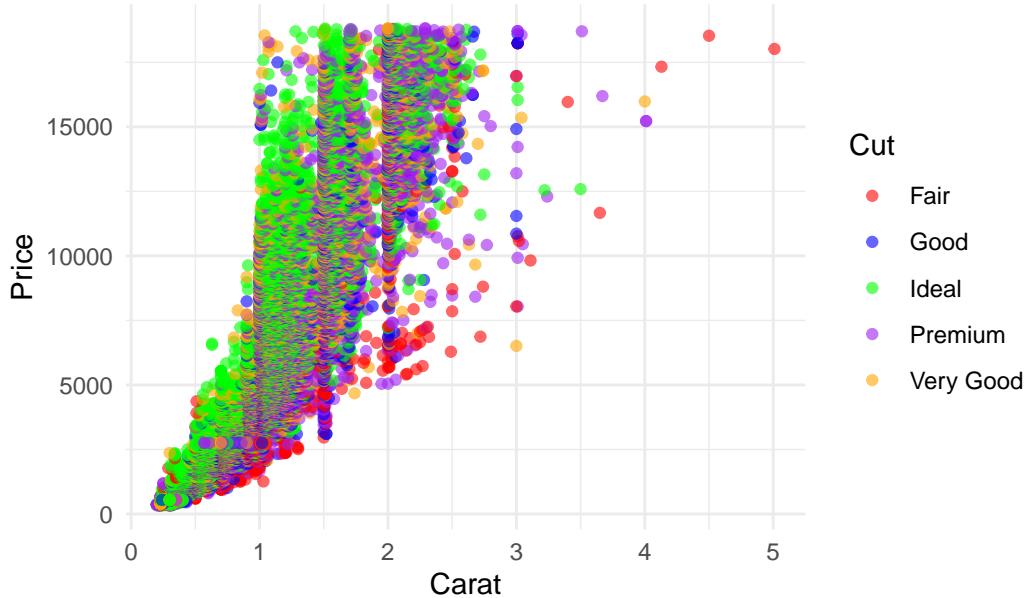
```

```

    "Premium" = "purple", "Very Good" = "orange"))+
theme_minimal()

```

Fig.3 Carat vs Price by Cut



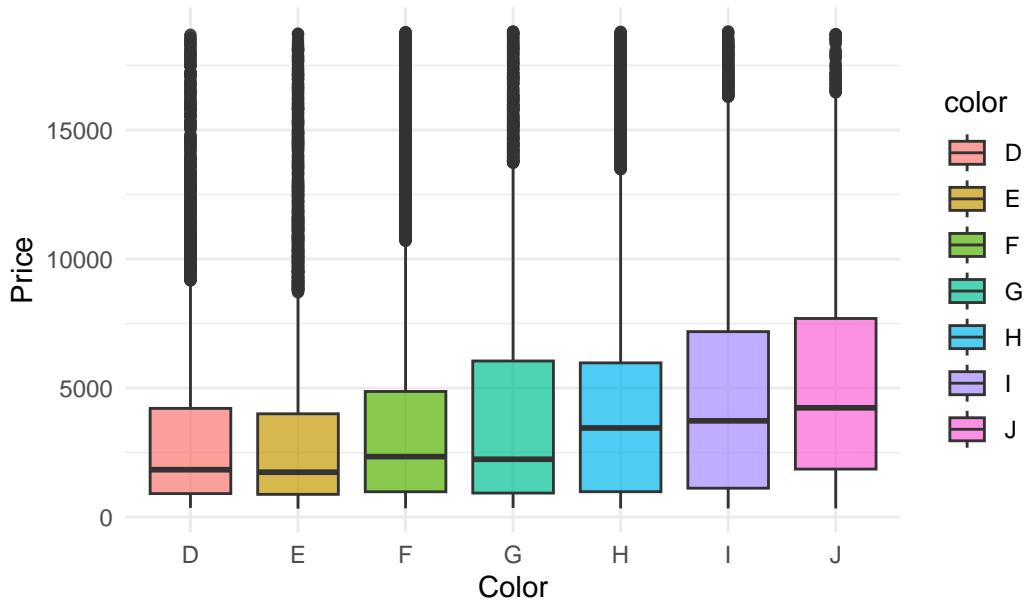
從圖 3 可發現大致上越重的鑽石價格越高。

```

# 顏色對價格圖
ggplot(data, aes(x = factor(color,
                             levels = c("1", "2", "3", "4", "5", "6", "7"),
                             labels = c("D", "E", "F", "G", "H", "I", "J")),
            y = price,
            fill = factor(color,
                          levels = c("1", "2", "3", "4", "5", "6", "7"),
                          labels = c("D", "E", "F", "G", "H", "I", "J")))) +
geom_boxplot(alpha = 0.7) +
labs(title = "Fig.4 Price Distribution by Color",
     x = "Color",
     y = "Price",
     fill = "color") +
theme_minimal()

```

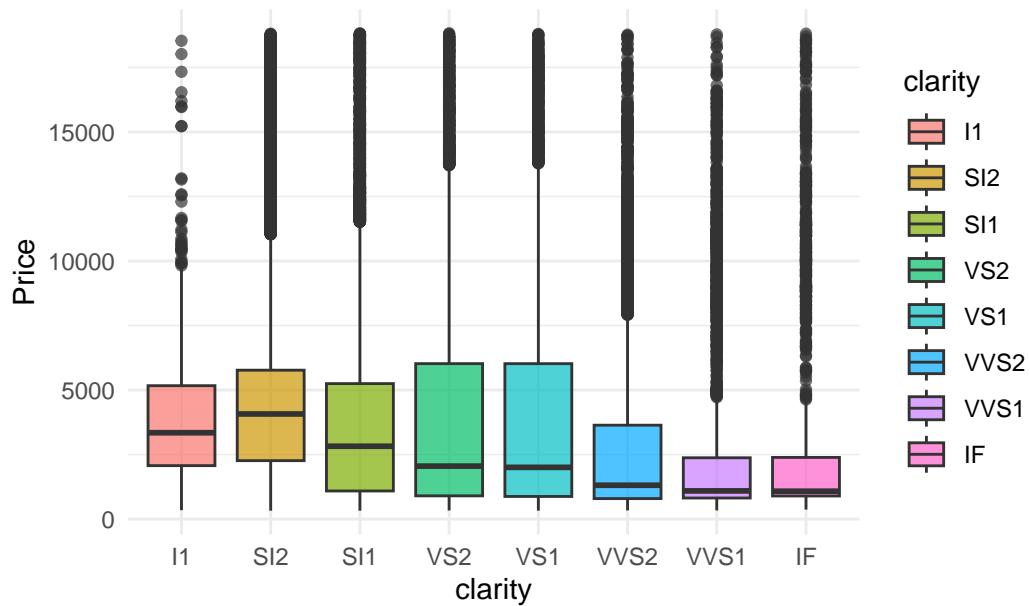
Fig.4 Price Distribution by Color



從圖 4 可發現當分類越靠近接近無色時價格越高。

```
# 淨度對價格
ggplot(data, aes(x = factor(data$clarity,
                             levels = c("1", "2", "3", "4", "5", "6", "7", "8"),
                             labels = c("I1","SI2","SI1","VS2","VS1","VVS2","VVS1","IF")),
         y = price,
         fill = factor(data$clarity,
                       levels = c("1", "2", "3", "4", "5", "6", "7", "8"),
                       labels = c("I1","SI2","SI1","VS2","VS1","VVS2","VVS1","IF")))) +
geom_boxplot(alpha = 0.7) +
labs(title = "Fig.5 Price Distribution by clarity",
     x = "clarity",
     y = "Price",
     fill = "clarity") +
theme_minimal()
```

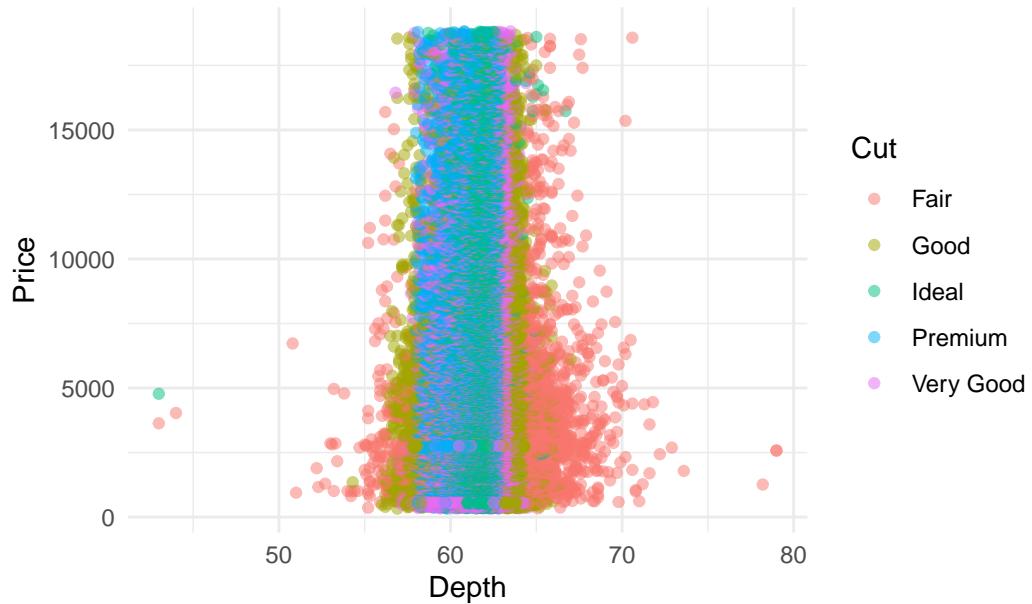
Fig.5 Price Distribution by clarity



從圖 5 可發現單一淨度指標對價格並沒有直接關連，高淨度的鑽石未必會有高價格。

```
# 深度對價格
ggplot(data, aes(x = depth,
                  y = price,
                  color = factor(data$cut,
                                 levels = c("1", "2", "3", "4", "5"),
                                 labels = c("Fair", "Good", "Ideal", "Premium", "Very Good")))) +
  geom_point(alpha = 0.5) +
  labs(title = "Fig.6 Depth vs Price by Cut",
       x = "Depth",
       y = "Price",
       color = "Cut") +
  theme_minimal()
```

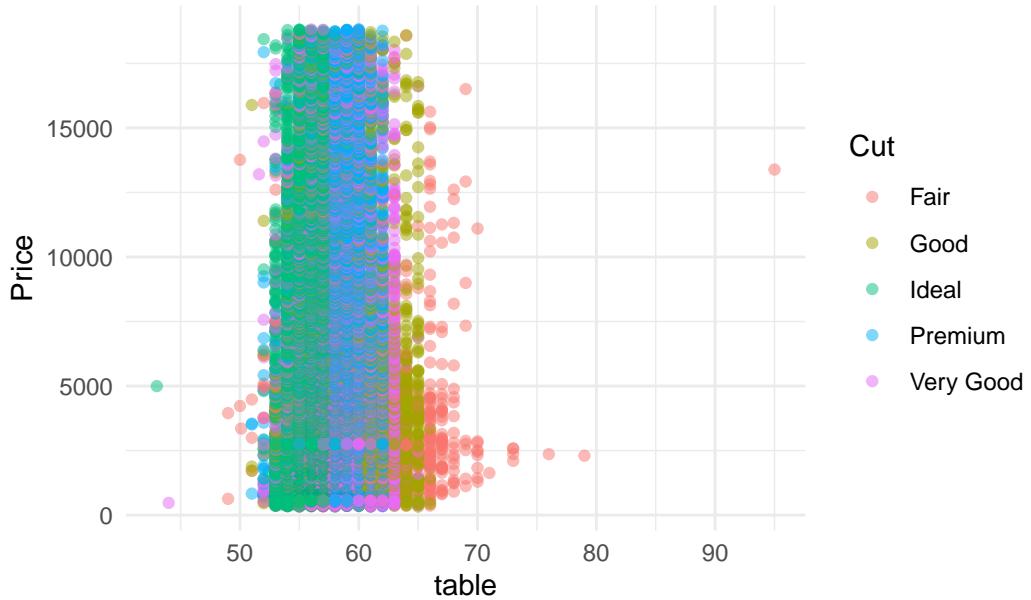
Fig.6 Depth vs Price by Cut



從圖 6 可發現深度和價格沒有相關，且深度大多集中於 60 附近，推測是因為深度比例在此區間能切割出最明亮的鑽石。

```
# 檯面尺寸對價格
ggplot(data, aes(x = table,
                  y = price,
                  color = factor(data$cut,
                                 levels = c("1", "2", "3", "4", "5"),
                                 labels = c("Fair", "Good", "Ideal", "Premium", "Very Good")))) +
  geom_point(alpha = 0.5) +
  labs(title = "Fig.7 table vs Price by Cut",
       x = "table",
       y = "Price",
       color = "Cut") +
  theme_minimal()
```

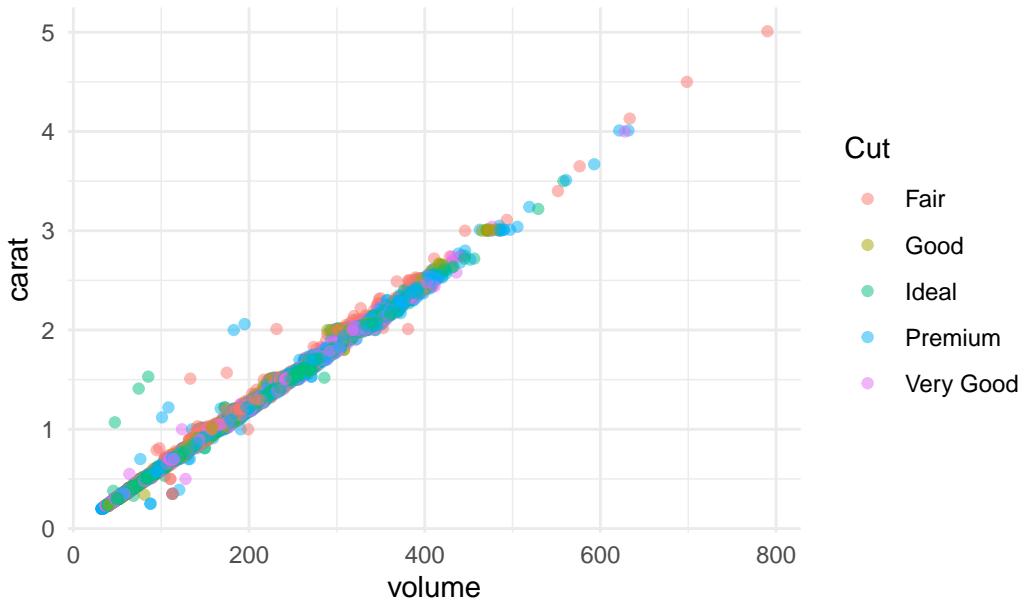
Fig.7 table vs Price by Cut



從圖 7 可發現檯面尺寸對價格沒有相關，且檯面尺寸約集中在 56~62 之間，推測也是在這個區間中能切割出最好的鑽石

```
# 體積對重量
ggplot(data,aes(x = volume,
                 y = carat,
                 color = factor(data$cut,
                               levels = c("1", "2", "3", "4", "5"),
                               labels = c("Fair", "Good", "Ideal", "Premium", "Very Good"))))+geom_point(alpha = 0.5)+labs(title ="Fig.8 volume vs carat by Cut",
x ="volume",
y ="carat",color ="Cut") +theme_minimal()
```

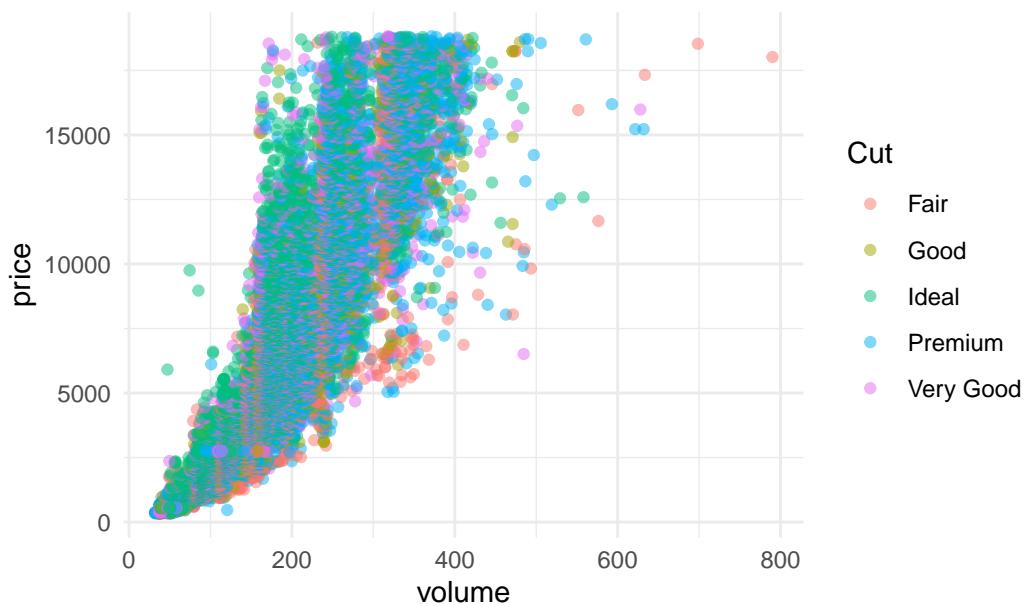
Fig.8 volume vs carat by Cut



由圖 8 可發現體積和重量（克拉）有近乎正比關係

```
ggplot(data, aes(x = volume,
                  y = price,
                  color = factor(data$cut,
                                 levels = c("1", "2", "3", "4", "5"),
                                 labels = c("Fair", "Good", "Ideal", "Premium", "Very Good")))) +
  geom_point(alpha = 0.5) +
  labs(title = "Fig.9 volume vs carat by Cut",
       x = "volume",
       y = "price",
       color = "Cut") +
  theme_minimal()
```

Fig.9 volume vs carat by Cut



由圖 9 可發現與圖 3 非常類似，可知道從重量和從體積對上價格能得到相同結果

典型相關分析 (CCA)

分析：幾何特性 vs. 做工及價格之間的關係

```
# 選擇兩組變數
X <- data[, c("carat", "color", "clarity", "volume")]
Y <- data[, c("price", "cut", "depth", "table")]
cca <- cancor(X,Y)
print(cca)

$cor
[1] 0.95172554 0.52416801 0.07949970 0.01126333

$xcoef
 [,1]      [,2]      [,3]      [,4]
carat -3.613020e-03 0.1871277417 0.0571111515 0.0344228776
color  3.584621e-04 0.0002197235 0.0005444205 -0.0025843239
clarity -5.881181e-04 -0.0006368735 0.0025672725 0.0008290882
volume -3.915824e-05 -0.0011613396 -0.0003465559 -0.0001993115

$ycoef
 [,1]      [,2]      [,3]      [,4]
price -1.074556e-06 -1.219673e-07 1.343038e-07 4.872832e-09
cut    8.470587e-05 -3.778534e-04 -4.155291e-04 -4.256938e-03
depth -4.714159e-05 2.760516e-03 1.327289e-03 -8.880334e-04
table -7.900934e-05 1.419096e-03 -1.454370e-03 2.112245e-04

$xcenter
 carat      color      clarity      volume
 0.7976816  3.5939911  4.0514651 129.8015976

$ycenter
 price      cut      depth      table
3930.845122 3.553097 61.749483 57.457007

cca$cor
[1] 0.95172554 0.52416801 0.07949970 0.01126333

cca$xcoef;cca$ycoef

 [,1]      [,2]      [,3]      [,4]
carat -3.613020e-03 0.1871277417 0.0571111515 0.0344228776
color  3.584621e-04 0.0002197235 0.0005444205 -0.0025843239
clarity -5.881181e-04 -0.0006368735 0.0025672725 0.0008290882
volume -3.915824e-05 -0.0011613396 -0.0003465559 -0.0001993115

 [,1]      [,2]      [,3]      [,4]
price -1.074556e-06 -1.219673e-07 1.343038e-07 4.872832e-09
cut    8.470587e-05 -3.778534e-04 -4.155291e-04 -4.256938e-03
depth -4.714159e-05 2.760516e-03 1.327289e-03 -8.880334e-04
table -7.900934e-05 1.419096e-03 -1.454370e-03 2.112245e-04
```

發現：

第一典型相關變數：最大典型相關係數為 0.9513，第一典型變數主要由 carat 和 table 貢獻組成

第二典型相關變數:最大典型相關係數為0.2112(相關性低)

```
X_loadinds <- cor(X,as.matrix(X) %*% cca$xcoef)
Y_loadinds <- cor(Y,as.matrix(Y) %*% cca$ycoef)
X_loadinds;Y_loadinds
```

	[,1]	[,2]	[,3]	[,4]
carat	-0.9712718	0.10329533	-0.1462553	-0.1567498
color	-0.1821301	0.08305418	0.2819089	-0.9383272
clarity	0.1613851	-0.31157473	0.9171247	0.1890985
volume	-0.9725448	0.05867045	-0.1556928	-0.1627085

	[,1]	[,2]	[,3]	[,4]
price	-0.999094752	-0.03261789	0.01970337	-0.01890836
cut	-0.022582242	-0.16211424	-0.29373299	-0.94176958
depth	0.003096597	0.71910138	0.68255501	-0.13039280
table	-0.159873694	0.43667075	-0.88418926	0.04436677

發現:

第一典型變數主要受carat(-),volume(-)和price(-)影響

第二典型變數主要受clarity(+),depth(-)和table(-)影響

3. Construct a predictive model for price

Data splitting

資料集以 7:3 比例切割成訓練集和測試集。

```
set.seed(123)
train_ratio <- 0.7
n <- nrow(data)
train_indices <- sample(1:n, size = floor(train_ratio * n))
train_data <- data[train_indices, ]
test_data <- data[-train_indices, ]
```

Linear regression model

我們採用線性回歸模型搭配 AIC 的逐步回歸進行變數選取。

```
model.1 <- stepAIC(
  lm(price ~ ., data = train_data),
  direction = "both",
  trace = 0)
summary(model.1)
```

Call:

```
lm(formula = price ~ carat + cut + color + clarity + depth +
  table + volume, data = train_data)
```

Residuals:

Min	1Q	Median	3Q	Max
-14594.5	-692.2	-171.0	554.0	8902.5

```

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -240.830    433.162 -0.556   0.578
carat        2805.484   325.212  8.627 < 2e-16 ***
cut          75.969     6.287 12.084 < 2e-16 ***
color       -320.259    3.924 -81.608 < 2e-16 ***
clarity      526.283    4.175 126.043 < 2e-16 ***
depth        -28.831    5.204 -5.540 3.04e-08 ***
table        -40.811    3.239 -12.599 < 2e-16 ***
volume       37.058     2.008 18.454 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Residual standard error: 1224 on 37736 degrees of freedom
 Multiple R-squared: 0.9055, Adjusted R-squared: 0.9055
 F-statistic: 5.167e+04 on 7 and 37736 DF, p-value: < 2.2e-16

```
vif(model.1)
```

	carat	cut	color	clarity	depth	table	volume
594.410725	1.052485	1.115091	1.189357	1.413459	1.316505	590.073121	

```
# remove carat
model.2 <- stepAIC(
  lm(price ~ .-carat , data = train_data),
  direction = "both",
  trace = 0)
summary(model.2)
```

Call:
`lm(formula = price ~ (carat + cut + color + clarity + depth +
 table + volume) - carat, data = train_data)`

Residuals:

Min	1Q	Median	3Q	Max
-14394.9	-693.4	-171.1	554.5	10270.3

Coefficients:

```

            Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.020e+03 3.813e+02 -5.297 1.18e-07 ***
cut          7.312e+01 6.284e+00 11.636 < 2e-16 ***
color       -3.197e+02 3.928e+00 -81.402 < 2e-16 ***
clarity      5.251e+02 4.177e+00 125.703 < 2e-16 ***
depth        -8.981e+00 4.672e+00 -1.922 0.0546 .
table        -3.110e+01 3.040e+00 -10.229 < 2e-16 ***
volume       5.436e+01 9.369e-02 580.220 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Residual standard error: 1226 on 37737 degrees of freedom
 Multiple R-squared: 0.9053, Adjusted R-squared: 0.9053
 F-statistic: 6.016e+04 on 6 and 37737 DF, p-value: < 2.2e-16

```
vif(model.2)
```

cut	color	clarity	depth	table	volume
-----	-------	---------	-------	-------	--------

```

1.049590 1.114809 1.188033 1.137104 1.157381 1.282051
oldformula <- model.2$formula

# remove depth
newformula <- update(oldformula, .~.-depth)
model.2 <- lm(newformula, data = train_data)
summary(model.2)

```

Call:
`lm(formula = newformula, data = train_data)`

Residuals:

Min	1Q	Median	3Q	Max
-14427.1	-694.1	-170.9	555.7	10191.6

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)							
(Intercept)	-2.677e+03	1.683e+02	-15.91	<2e-16 ***							
cut	7.505e+01	6.204e+00	12.10	<2e-16 ***							
color	-3.201e+02	3.922e+00	-81.61	<2e-16 ***							
clarity	5.260e+02	4.152e+00	126.68	<2e-16 ***							
table	-2.946e+01	2.918e+00	-10.09	<2e-16 ***							
volume	5.436e+01	9.369e-02	580.20	<2e-16 ***							

Signif. codes:	0	'***'	0.001	'**'	0.01	'*'	0.05	'.'	0.1	' '	1

Residual standard error: 1226 on 37738 degrees of freedom
Multiple R-squared: 0.9053, Adjusted R-squared: 0.9053
F-statistic: 7.218e+04 on 5 and 37738 DF, p-value: < 2.2e-16

檢查 VIF 可發現 carat 和 volume 值很大，故試將 carat 刪除建立 model.2

移除 carat 之後發現，depth 在顯著水準 0.05 之下變得不顯著，因此移除此變數，再配適一個模型。

接著把剩下的變數加上交互作用項再建立 model.3，可以看到 R^2 增加了 2% 推論增加交互作用項會提升模型準確率因此再用逐步回歸法而 model.2 的 R-squared = 0.9053，VIF 檢查後各變數也無異常。

因此以 model.2 作為預測模型。

以下將測試集資料輸入預測模型並評估表現:

```

predictions <- predict(model.2, newdata = test_data)
# 計算 MSE 和 RMSE
rmse <- sqrt(mean((test_data$price - predictions)^2))
test_r2 <- 1 - sum((predictions - test_data$price)^2) /
    sum((test_data$price - mean(test_data$price))^2)
# 輸出結果
cat("Testing R^2:", test_r2, "\n")

```

Testing R^2: 0.904271
`cat("Root Mean Squared Error (RMSE):", rmse, "\n")`

Root Mean Squared Error (RMSE): 1236.306

經過測試後，這個模型在測試集上的 R-square = 0.9043，RMSE = 1236.306

以線性回歸模型來說，已經是不錯的表現了。

```
model.3 <- lm(price ~ (cut+color+clarity+table+volume)^2 , data = train_data)
summary(model.3)
```

Call:

```
lm(formula = price ~ (cut + color + clarity + table + volume)^2,
  data = train_data)
```

Residuals:

Min	1Q	Median	3Q	Max
-9392.0	-651.2	-127.1	606.5	10414.3

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	3427.30271	687.86342	4.983	6.30e-07 ***
cut	-572.02434	122.72972	-4.661	3.16e-06 ***
color	654.68873	89.58338	7.308	2.76e-13 ***
clarity	-435.68280	95.53375	-4.561	5.12e-06 ***
table	-141.86053	12.05890	-11.764	< 2e-16 ***
volume	29.82969	2.11010	14.137	< 2e-16 ***
cut:color	-1.49396	3.32261	-0.450	0.652977
cut:clarity	-12.94204	3.67755	-3.519	0.000433 ***
cut:table	10.35824	2.07269	4.997	5.83e-07 ***
cut:volume	0.62352	0.08117	7.681	1.61e-14 ***
color:clarity	-84.27672	2.32817	-36.199	< 2e-16 ***
color:table	-5.38292	1.55749	-3.456	0.000549 ***
color:volume	-2.46088	0.04493	-54.773	< 2e-16 ***
clarity:table	14.27778	1.69520	8.422	< 2e-16 ***
clarity:volume	4.32748	0.04907	88.193	< 2e-16 ***
table:volume	0.29527	0.03666	8.054	8.24e-16 ***

Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.'
	0.1	' '	1	

Residual standard error: 1076 on 37728 degrees of freedom

Multiple R-squared: 0.9271, Adjusted R-squared: 0.9271

F-statistic: 3.2e+04 on 15 and 37728 DF, p-value: < 2.2e-16

```
model.4 <- stepAIC(
  model.3,
  direction = "both",
  trace = 0)
summary(model.4)
```

Call:

```
lm(formula = price ~ cut + color + clarity + table + volume +
  cut:clarity + cut:table + cut:volume + color:clarity + color:table +
  color:volume + clarity:table + clarity:volume + table:volume,
  data = train_data)
```

Residuals:

Min	1Q	Median	3Q	Max
-9389.8	-651.9	-127.0	606.5	10413.7

Coefficients:

```

              Estimate Std. Error t value Pr(>|t|)
(Intercept) 3430.46557  687.82017  4.987 6.15e-07 ***
cut          -576.94124 122.24027 -4.720 2.37e-06 ***
color        655.48569  89.56489  7.319 2.56e-13 ***
clarity      -435.45432  95.53139 -4.558 5.17e-06 ***
table        -141.72802 12.05518 -11.757 < 2e-16 ***
volume       29.82514   2.11006 14.135 < 2e-16 ***
cut:clarity -13.11619  3.65706 -3.587 0.000336 ***
cut:table    10.38715  2.07167  5.014 5.36e-07 ***
cut:volume   0.61255  0.07742  7.912 2.60e-15 ***
color:clarity -84.31173  2.32684 -36.234 < 2e-16 ***
color:table  -5.48548  1.54068 -3.560 0.000371 ***
color:volume -2.46140  0.04491 -54.803 < 2e-16 ***
clarity:table 14.28681  1.69506  8.428 < 2e-16 ***
clarity:volume 4.32756  0.04907 88.196 < 2e-16 ***
table:volume  0.29607  0.03662  8.085 6.40e-16 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residual standard error: 1076 on 37729 degrees of freedom
 Multiple R-squared: 0.9271, Adjusted R-squared: 0.9271
 F-statistic: 3.428e+04 on 14 and 37729 DF, p-value: < 2.2e-16

```

predictions <- predict(model.4, newdata = test_data)
# 計算 MSE 和 RMSE
rmse <- sqrt(mean((test_data$price - predictions)^2))
test_r2 <- 1 - sum((predictions - test_data$price)^2) /
  sum((test_data$price - mean(test_data$price))^2)
# 輸出結果
cat("Testing R^2:", test_r2, "\n")
```

Testing R^2: 0.9262337

```
cat("Root Mean Squared Error (RMSE):", rmse, "\n")
```

Root Mean Squared Error (RMSE): 1085.258

接著試試看加入交互作用項 (model.3) 並用逐步回歸法選出 AIC 最低的模型 (model.4)

可看到此模型的 R-square 進步到 92.6%，rmse 下降到 1085.26

可知交互作用存在於此模型，因此我們選擇繼續使用機器學習模型來適配

randomForest 參數微調

依據 RMSE 進行網格搜索，結果為 mtry: 4, min.node.size: 5

```

library(ranger)
library(parallel)
library(doParallel)

# 使用 caret 調參

# train_index <- caret::createDataPartition(data$price, times = 1, p = 0.7, list = FALSE)
# train_data <- data[train_indices, ]
# test_data <- data[-train_indices, ]

control <- trainControl(
```

```

method = "cv",
number = 5,
search = "grid"      # gridsearch
allowParallel = TRUE  # 開啟平行運算
)

# Hyperparameter space
tune_grid <- expand.grid(
  mtry = c(1:6),
  splitrule = "variance",
  min.node.size = c(5,10,20,40,100,200,250)
)

# 依據 RMSE 進行 Grid search
set.seed(123)
rf_model <- train(
  x = train_data[,-7],
  y = train_data$price,
  method = "ranger",
  tuneGrid = tune_grid,
  trControl = control,
  metric = "RMSE",
  num.trees = 1000
)

rf_model$bestTune
# Best parameters of Grid search:
# mtry = 4
# min.node.size = 5

# 使用最佳參數
set.seed(123)
final_rf <- ranger(
  formula = price ~.,
  data = train_data,
  importance = 'permutation',
  num.trees = 500,
  splitrule = 'variance',
  mtry = 4,
  min.node.size = 5,
  num.threads = 5      #ranger support parallel computation
)

predictions <- predict(final_rf, test_data)
rmse <- sqrt(mean((predictions$predictions - test_data$price)^2))
cat(" 測試集 RMSE:", rmse, "\n")

```

測試集 RMSE: 523.0623

可以發現測試集的 RMSE 降低到了 523.06，此模型相較線性回歸模型突出許多。

各變數的重要程度如圖 10 所示。

雖然無法得知哪些變數有交互作用，但 RMSE 下降許多，表示模型已經捕捉到交互作用。

```

library(dplyr)
importance_df <- final_rf$variable.importance %>%
  data.frame(variable = names(.), importance = .) %>%
  arrange(desc(importance))

ggplot(importance_df, aes(x = reorder(variable, importance), y = importance)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  coord_flip() +
  labs(title = "Fig.10 Variable Importance from Random Forest",
       x = "Variables",
       y = "Difference in MSE") +
  theme_minimal()

```

Fig.10 Variable Importance from Random Forest

