學號：0653440

載入的套件：

```
import pandas as pd
import numpy as np
from sklearn import linear_model
from sklearn.metrics import mean_absolute_error
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
```

一、載入資料表：

```
csv = pd.read_csv("nyc-rolling-sales.csv",encoding = "ISO-8859-1")
```

二、資料處理：

1. 決定各欄去留，及是否要轉 Dummy 及非數值的處理：

```
#各欄處理
csv=csv.drop(csv.columns[0], axis=1)#0
csv = pd.get_dummies(csv, columns=['BOROUGH'])#1
csv = pd.get_dummies(csv, columns=['NEIGHBORHOOD'])#3
csv = pd.get_dummies(csv, columns=['BUILDING CLASS CATEGORY'])#3
csv = pd.get_dummies(csv, columns=['TAX CLASS AT PRESENT'])#4
#05BLOCK
#06LOT
csv=csv.drop('EASE-MENT',axis = 1)#07
csv = pd.get_dummies(csv, columns=['BUILDING CLASS AT PRESENT'])#8
csv=csv.drop('ADDRESS',axis = 1)#9
csv=csv.drop('APARTMENT NUMBER',axis = 1)#10
csv = pd.get_dummies(csv, columns=['ZIP CODE'])#11
csv=csv.drop('RESIDENTIAL UNITS',axis = 1)#12
csv=csv.drop('COMMERCIAL UNITS',axis = 1)#13
#14TOTAL UNITS
csv = csv[csv['LAND SQUARE FEET'].str.contains(" -  ") == False]#15
csv = csv[csv['GROSS SQUARE FEET'].str.contains(" -  ") == False]#16
csv=csv.drop('YEAR BUILT',axis = 1)#17
csv = pd.get_dummies(csv, columns=['TAX CLASS AT TIME OF SALE'])#18
csv = pd.get_dummies(csv, columns=['BUILDING CLASS AT TIME OF SALE'])#19
csv = csv[csv['SALE PRICE'].str.contains(" -  ") == False]#20
```

說明：

(1) Index: 刪除。

(2) Borough: 有 1~5 種可能，故轉 Dummy。

(3) Neighborhood: 轉 Dummy。

(4) Building Class Category: 轉 Dummy。

(5) Tax Class at Present: 轉 Dummy。

(6) Block、Lot: 雖然不是數值但是數字，轉 Dummy 會太多欄，經測試決定以原值轉 float 留下。

(7) Easement: 刪除。

(8) Building Class at Present: 轉 Dummy。

(9) Address: 刪除。

(10) Apartment Number: 刪除。

(11) Zip Code: 轉 Dummy。

(12) Residential Units: 刪除。

(13) Commercial Units: 刪除。

(14) Total Units: 保留原值。

(15) Land Square Feet: 將資料為" - "的列刪除。

(16) Gross Square Feet: 將資料為" - "的列刪除。

(17) Year Built: 刪除。

(18) Tax Class at Time of Sale: 轉 Dummy。

(19) Building Class at Time of Sale: 轉 Dummy。

(20) Sale Price: 將資料為" - "的列刪除。

(21) Sale Date: 刪除。

2. 刪除極端值：

```
m = np.percentile(csv['SALE PRICE'], 22)
M = np.percentile(csv['SALE PRICE'], 83)

csv = csv[csv['SALE PRICE'] > m]
csv = csv[csv['SALE PRICE'] < M]
```

說明：經測試後決定將「SALE PRICE」在前 22%及後 17%的資料刪除。

三、各用「Linear Regression」及「Random Forest Regression」來預測 Sale Price 並計算 Mean Absolute Error：

```
datadic_X = data
datadic_X_train, datadic_X_test, datadic_y_train, datadic_y_test = train_test_split(data, target,train_size=0.75,test_size=0.25)
```

說明：切割 75%為訓練集、25%測試集。

1. Linear Regression

```
regr = linear_model.LinearRegression()
regr.fit(datadic_X_train, datadic_y_train)
datadic_y_pred = regr.predict(datadic_X_test)
print('----Linear Regression----')
print("Mean absolute error:" ,mean_absolute_error(datadic_y_test, datadic_y_pred))
```

2. Random Forest Regression

```
rf = RandomForestRegressor()
rf.fit(datadic_X_train, datadic_y_train.ravel())
datadic_y_pred = rf.predict(datadic_X_test)
print('------------------------------')
print('----Random Forest Regression----')
print("Mean absolute error:",mean_absolute_error(datadic_y_test, datadic_y_pred))
```

四、計算結果，MAE 改善：

1. 刪除極端值前：

```
----Linear Regression----
Mean absolute error: 1065351.25482
------------------------------
----Random Forest Regression----
Mean absolute error: 798214.952892
```

2. 刪除極端值後：

```
----Linear Regression----
Mean absolute error: 140923.129503
------------------------------
----Random Forest Regression----
Mean absolute error: 130497.316365
```