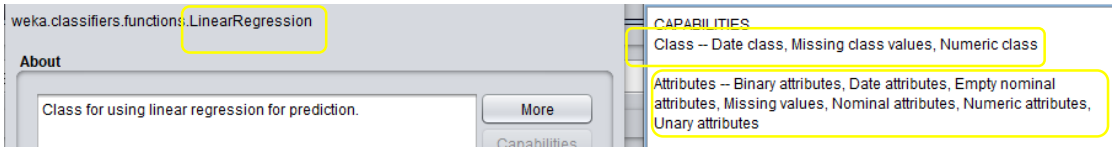
# ECT HW3

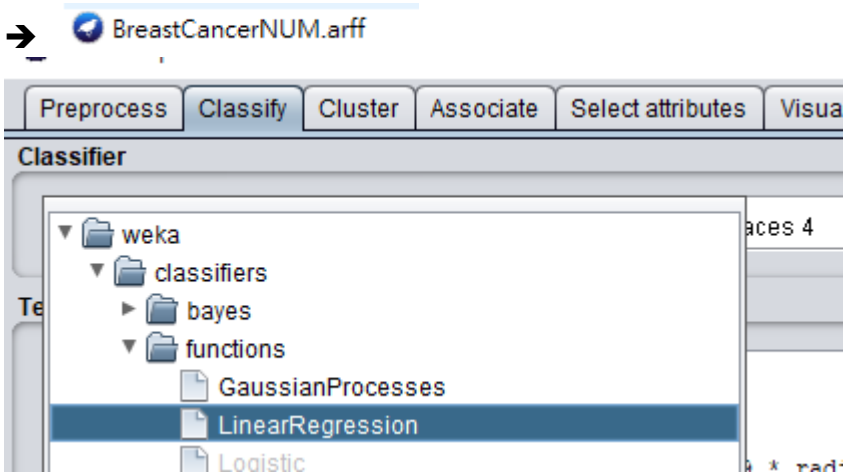
## Q1

LinearRegression:

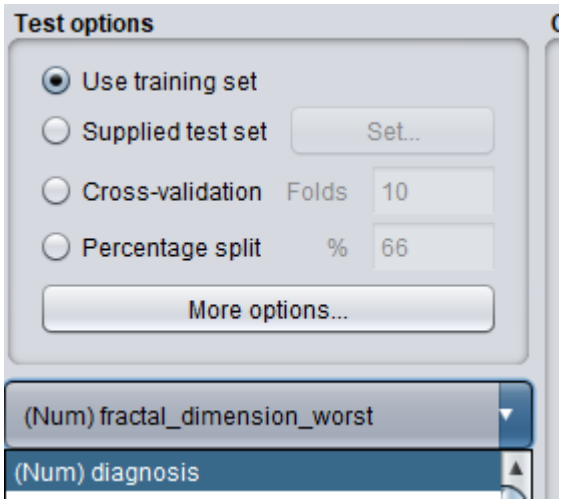



首先，先確認 LinearRegression 的使用條件，可以看出他的 Output 分類需要

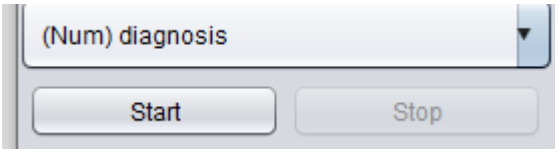
Numeric，因此我們在做 LinearRegression 時用 NUM 的檔案




選好檔案後，在 Classify 的 function 中找到 LinearRegression

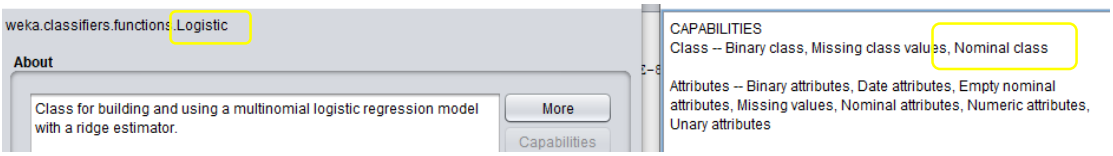



依題目要求，設定 Use training set 和 output 為 diagnosis



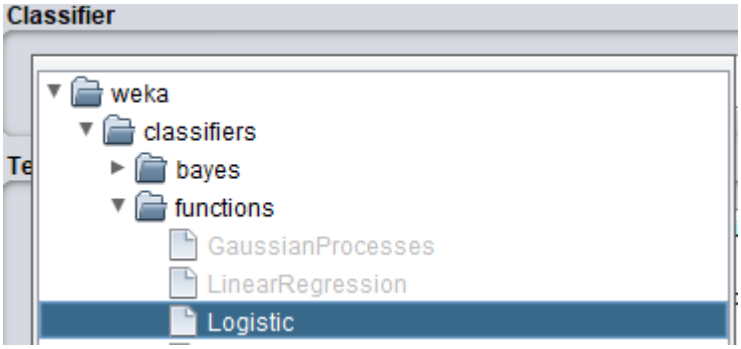

點選 start 則開始分析。

Logistic:

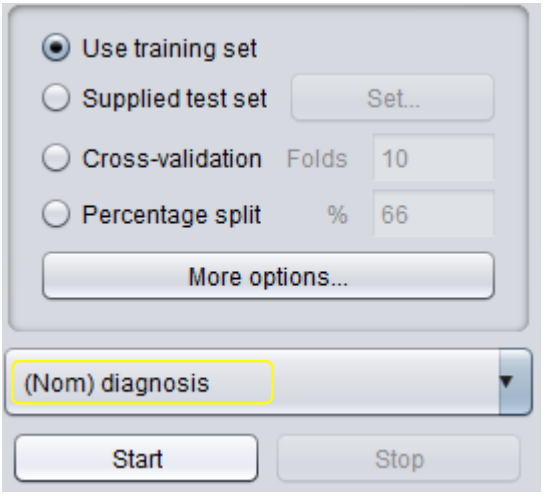



同樣查看使用條件，可以發現他的分類為 Numinal，因此選用 NOM 檔案

➔ BreastCancerNOM.arff




在 Classifier 中選取 Logistic




依題目要求，設定 Use training set 和 output 為 diagnosis。

注意!這裡 output 已經變為 Nominal。點選 start 即可開始分析。

## (a)

```
Linear Regression Model

diagnosis =

     -0.0709 * radius_mean +
     -0.0047 * perimeter_mean +
      0.0008 * area_mean +
      3.8735 * compactness_mean +
     -1.1601 * concavity_mean +
     -1.9936 * concave points_mean +
     -0.969  * radius_se +
      0.0363 * perimeter_se +
      0.0029 * area_se +
    -19.4375 * smoothness_se +
      3.0168 * concavity_se +
     -0.0105 * texture_worst +
     -0.0058 * perimeter_worst +
      0.0002 * area_worst +
     -0.3631 * concavity_worst +
     -1.8387 * concave points_worst +
     -0.811  * symmetry_worst +
     -3.909  * fractal_dimension_worst +
      3.133
```

由 LinearRegression 跑出來的結果如上所示，正相關代表參數值上升，會使 output 值也跟著上升，反之則是負相關，因此可得知上圖中「+」的參數為正相關、「-」的為負相關。整理得知有 6 個因素正相關、12 個因素負相關。

## (b)

```
Attributes:   31
              diagnosis
              radius_mean
              texture_mean
              perimeter_mean
              area_mean
              smoothness_mean
              compactness_mean
              concavity_mean
              concave points_mean
              symmetry_mean
              fractal_dimension_mean
              radius_se
```

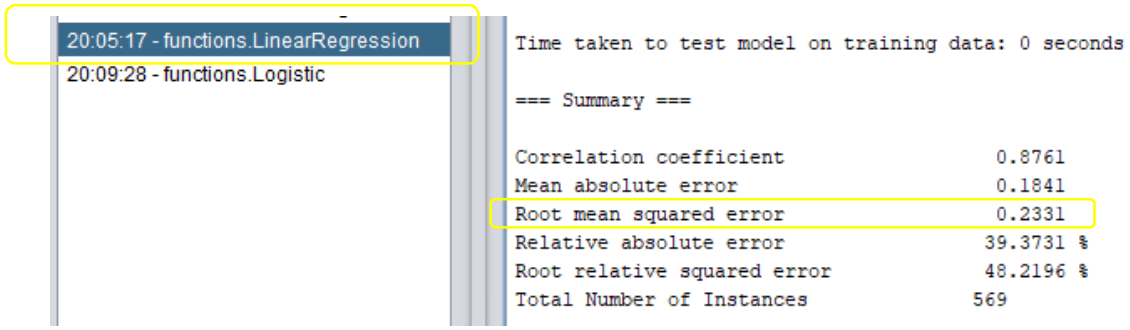
由此圖可知，input + output 的屬性有 31 種，因此是使用 30 種屬性來做預

測。由題(a)可得知用有(6 + 12)個因素會影響 output，因此有 30-18 = 12 個因素不會是無相關的。例如：

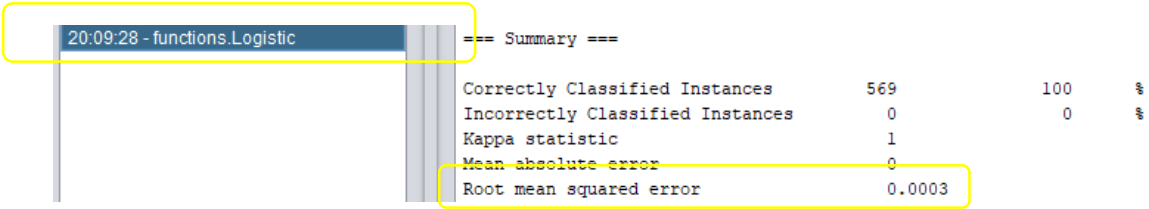
1.texture_mean　2.smoothness_mean　3. symmetry_mean

**(c)**

For LinearRegression:




For Logistic:




可以看出 LinearRegression 的均方根誤差 = 0.2331、Logistic 的均方根誤差 = 0.0003，因此從誤差越小預測越準確的觀點來評價，Logistic 的 model 表現得較好。但我們全部都是使用 training set 的資料來訓練、預測，因此無法保證在實際 testing 時會有相同的結果，雖然應該會差不多。

# Q2

```
import pandas as pd
df = pd.read_csv('BreastCancer.csv')
df
```

| | diagnosis | radius_mean | texture_mean | perimeter_mean | area_mean | smoothness_mean | compactness_mean | concavity_mean | concave points_mean | symmetry_mea |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 17.99 | 10.38 | 122.80 | 1001.0 | 0.11840 | 0.27760 | 0.30010 | 0.14710 | 0.24 |
| 1 | 0 | 20.57 | 17.77 | 132.90 | 1326.0 | 0.08474 | 0.07864 | 0.08690 | 0.07017 | 0.18 |
| 2 | 0 | 19.69 | 21.25 | 130.00 | 1203.0 | 0.10960 | 0.15990 | 0.19740 | 0.12790 | 0.20 |
| 3 | 0 | 11.42 | 20.38 | 77.58 | 386.1 | 0.14250 | 0.28390 | 0.24140 | 0.10520 | 0.25 |
| 4 | 0 | 20.29 | 14.34 | 135.10 | 1297.0 | 0.10030 | 0.13280 | 0.19800 | 0.10430 | 0.18 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 564 | 0 | 21.56 | 22.39 | 142.00 | 1479.0 | 0.11100 | 0.11590 | 0.24390 | 0.13890 | 0.17 |
| 565 | 0 | 20.13 | 28.25 | 131.20 | 1261.0 | 0.09780 | 0.10340 | 0.14400 | 0.09791 | 0.17 |
| 566 | 0 | 16.60 | 28.08 | 108.30 | 858.1 | 0.08455 | 0.10230 | 0.09251 | 0.05302 | 0.15 |
| 567 | 0 | 20.60 | 29.33 | 140.10 | 1265.0 | 0.11780 | 0.27700 | 0.35140 | 0.15200 | 0.23 |
| 568 | 1 | 7.76 | 24.54 | 47.92 | 181.0 | 0.05263 | 0.04362 | 0.00000 | 0.00000 | 0.15 |

➔ 先讀取 csv 檔，並查看資料大致樣貌。

```
#x:input
x = df.loc[:,"radius_mean":]
print(x)
#y:output
y = df.loc[:,["diagnosis"]]
print(y)

x_name = []
for name in x:
    x_name.append(name)
y_name = []
for name in y:
    y_name.append(name)
# print(x_name)
# print(y_name)
```

➔ 用 loc 函數把資料切成 input、output，並且把每個 attribute 記錄下來，

方便之後比對各 attribute 的係數。

```
model_linear = LinearRegression()
model_linear.fit(x, y)
```

➔ 用 LinearRegression 初始化 model，再用 fit 函數把 input、output 放進

去訓練

```
for i,j in zip(x_name, model_linear.coef_[0]):
    print(i,":",j)
```

```
radius_mean : 0.2177720556001026
texture_mean : -0.004545468674191476
perimeter_mean : -0.023739860969339062
area_mean : -0.000317834750195312
smoothness_mean : -0.08468913708552854
compactness_mean : 4.222035251614448
concavity_mean : -1.3979972832584853
concave points_mean : -2.1418330270055286
symmetry_mean : -0.10270920015983347
fractal_dimension_mean : -0.033261609551094265
radius_se : -0.4349559322344023
texture_se : 0.006758472331847354
perimeter_se : 0.022520257685144932
area_se : 0.0009232178860695204
smoothness_se : -15.854320748151878
compactness_se : -0.06490340897230286
concavity_se : 3.565467985633931
concave points_se : -10.567951307787231
symmetry_se : -1.6973406942713358
fractal_dimension_se : 7.146440155040075
radius_worst : -0.19518312138034666
texture_worst : -0.0071593751990308915
```

➔ 用之前建立好的 attribute_name 列表，加上 coef 函數取得每個 attribute 的係數，來印出 attribute – coefficient 的關係。

```
# get the most influential attribute
MaxAbsCoef_name = x_name[0] # intial the max absolute coefficient attribute name
max_Abs_Coef = 0 # intial the coefficient
actual_Coef = 0 # store the actual coefficient

# iterate the name and its coefficient to find the max
for i,j in zip(x_name, model_linear.coef_[0]):
    if(abs(j) > max_Abs_Coef):
        max_Abs_Coef = abs(j)
        MaxAbsCoef_name = i
        actual_Coef = j
```

➔ 此段程式碼是為了找出「影響程度最大」的參數，也就是「係數的絕對值」最大的參數。前 3 行為初始化參數，用以記錄目前的最大值，並記錄實際的係數為何；後段的 for 迴圈用以找出絕對值後最大的係數。

```
print("Most influential attribute:", MaxAbsCoef_name, "\nCoefficient:", actual_Coef)
print("Score:", model_linear.score(x,y))
```

```
Most influential attribute: smoothness_se
Coefficient: -15.854320748151878
Score: 0.7743246526421793
```

➔ 純粹用以觀察結果，可以看出最有影響力的係數為何、係數值為多少，並為此模型進行評分。

```
from sklearn.linear_model import LogisticRegression
model_logistic = LogisticRegression(solver="liblinear")
model_logistic.fit(x,y.values.ravel())
```

➔ 進行 logistic training 時也是先用 LogisticRegression()函數初始化 model，再用 fit 把 input、output 放進去。注意這裡有使用 ravel()函數用來把數據拉成 1 維。雖然不使用這個函數仍然可以運行正確，但會跑出一個警告，告訴我們她期望的數據形式是 1 維陣列，因此做此修正。

```
for i,j in zip(x_name, model_logistic.coef_[0]):
    print(i,":",j)
print(model_logistic.score(x,y))
```

```
texture_se : 1.2529957816759405
perimeter_se : 0.002090822460772107
area_se : -0.0948209590492166
smoothness_se : -0.017014462640756244
compactness_se : 0.004571449653399707
concavity_se : -0.04862958087335596
concave points_se : -0.04039845840974738
symmetry_se : -0.042591970067049956
fractal_dimension_se : 0.0063224179436422775
radius_worst : 1.2467684957496594
texture_worst : -0.34605024148752933
perimeter_worst : -0.12450042457343312
area_worst : -0.02408806417723571
smoothness_worst : -0.2869472227912671
compactness_worst : -1.1451701678519965
concavity_worst : -1.5952697989200615
concave points_worst : -0.6589910375574494
symmetry_worst : -0.6965238758168915
fractal_dimension_worst : -0.11531342848055402
Score: 0.9595782073813708
```

➔ 一樣印出結果來觀察各個係數、model 的分數

## (a)

```
for i,j in zip(x_name, model_linear.coef_[0]):
    print(i,":",j)
```

```
radius_mean : 0.2177720556001026
texture_mean : -0.004545468674191476
perimeter_mean : -0.023739860969339062
area_mean : -0.000317834750195312
smoothness_mean : -0.08468913708552854
compactness_mean : 4.222035251614448
concavity_mean : -1.3979972832584853
concave points_mean : -2.1418330270055286
symmetry_mean : -0.10270920015983347
fractal_dimension_mean : -0.033261609551094265
radius_se : -0.4349559322344023
texture_se : 0.006758472331847354
perimeter_se : 0.022520257685144932
area_se : 0.0009232178860695204
smoothness_se : -15.854320748151878
compactness_se : -0.06490340897230286
concavity_se : 3.565467985633931
concave points_se : -10.567951307787231
symmetry_se : -1.6973406942713358
fractal_dimension_se : 7.146440155040075
radius_worst : -0.19518312138034666
texture_worst : -0.0071593751990308915
perimeter_worst : 0.002435050570355807
area_worst : 0.0010112233180037346
smoothness_worst : -0.5428568612985524
compactness_worst : -0.06715829411820029
concavity_worst : -0.38119121482416984
concave points_worst : -0.4643098953922134
symmetry_worst : -0.5567875460100401
```

➔ 由上圖可知 LinearRegression model 中

「area_mean」的係數 = -0.000317834750195312

## (b)

題目要求為「影響最大的因素」，此影響應該無關「正」、「負」，因此在此定義影響最大的因素為「attribute 的係數取絕對值後的最大值」，也就是哪一個維度的屬性變化後，能最有效的影響 output。

```
# get the most influential attribute
MaxAbsCoef_name = x_name[0] # intial the max absolute coefficient attribute name
max_Abs_Coef = 0 # intial the coefficient
actual_Coef = 0 # store the actual coefficient

# iterate the name and its coefficient to find the max
for i,j in zip(x_name, model_linear.coef_[0]):
    if(abs(j) > max_Abs_Coef):
        max_Abs_Coef = abs(j)
        MaxAbsCoef_name = i
        actual_Coef = j
```

```
print("Most influential attribute:", MaxAbsCoef_name, "\nCoefficient:", actual_Coef)
```

```
Most influential attribute: smoothness_se
Coefficient: -15.854320748151878
```

➔ 如上圖所示，初始化 3 個參數來，分別記錄當前取絕對值後的最大值、屬性名稱、實際係數。用 for 迴圈來做比較並更新參數。最後把它印出來。

由此可知影響最大的因素為「smoothness_se」，它的影響程度約有-15.85 左右的係數大小，只要這個增加一個小單位，就會使 output 下降許多。

**(c)**

```
# For Q2 (c)
print(model_linear.score(x,y))
print(model_logistic.score(x,y))
```

```
0.7743246526421793
0.9595782073813708
```

➔ 由此圖可看出，logictic model 的正確率較高