Python部分

1. 載入 Churn_Modelling.csv 資料集，並印出哪些欄位含有遺漏值 (missing value)。 (5%)

```
In [2]: df = pd.read_csv('Churn_Modelling.csv')
```

```
In [3]: df.isnull().sum()
```

```
Out[3]: CustomerId         0
        CredRate           0
        Geography          0
        Gender             4
        Age                6
        Tenure             0
        Balance            0
        Prod Number        0
        HasCrCard          0
        ActMem             0
        EstimatedSalary    4
        Exited             0
        dtype: int64
```

2. 以平均值填入 EstimatedSalary 的遺漏值，以眾數填入 Age 與 Gender 的遺漏值。(10%)

```
In [4]: df['EstimatedSalary'].fillna((df['EstimatedSalary'].mean()), inplace=True)
        df['Age'] = df['Age'].fillna(df['Age'].mode()[0])
        df['Gender'] = df['Gender'].fillna(df['Gender'].mode()[0])
```

```
In [5]: df.isnull().sum()
```

```
Out[5]: CustomerId         0
        CredRate           0
        Geography          0
        Gender             0
        Age                0
        Tenure             0
        Balance            0
        Prod Number        0
        HasCrCard          0
        ActMem             0
        EstimatedSalary    0
        Exited             0
        dtype: int64
```

3. 修改欄位名稱，將 CredRate 改成 CreditScore、ActMem 改成 IsActiveMember、Prod Number 改成 NumOfProducts、Exited 改成Churn，以利後續分析資料。 (5%)

```
In [6]: df.rename(columns={'CredRate': 'CreditScore',
                           'ActMem': 'IsActiveMember',
                           'Prod Number': 'NumOfProducts',
                           'Exited': 'Churn'}, inplace=True)
        df.head()
```

Out[6]:

	CustomerId	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Churn
0	15634602	619	France	Female	42.0	2	0.00	1	1	1	101348.88	1
1	15647311	608	Spain	Female	41.0	1	83807.86	1	0	1	112542.58	0
2	15619304	502	France	Female	42.0	8	159660.80	3	1	0	113931.57	1
3	15701354	699	France	Female	39.0	1	0.00	2	0	0	93826.63	0
4	15737888	850	Spain	Female	43.0	2	125510.82	1	1	1	79084.10	0

4. 去除 CustomerId,欄位，並將Geography、Gender、HasCrCard、 Churn、IsActiveMember 修改資料型態為 category，印出所有欄位的資料型態，並存成新的 CSV 檔 (設定index=False)。(5%)

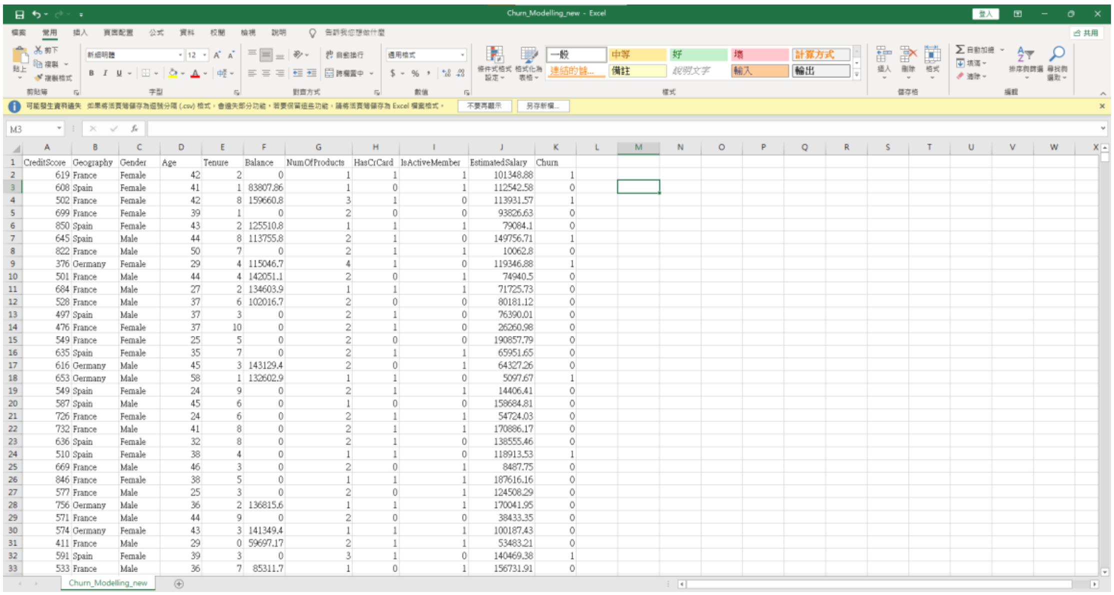
```
In [7]: #刪除CustomerId 欄位
        df = df.drop(['CustomerId'], axis = 1)
```

```
In [8]: #將Geography、Gender、HasCrCard、 Churn、IsActiveMember 修改資料型態為 category
        df['Geography'] = df.Geography.astype('category')
        df['Gender'] = df.Gender.astype('category')
        df['HasCrCard'] = df.HasCrCard.astype('category')
        df['Churn'] = df.Churn.astype('category')
        df['IsActiveMember'] = df.IsActiveMember.astype('category')
        df.dtypes
```

```
Out[8]: CreditScore          int64
        Geography         category
        Gender            category
        Age                float64
        Tenure               int64
        Balance            float64
        NumOfProducts        int64
        HasCrCard         category
        IsActiveMember    category
        EstimatedSalary    float64
        Churn             category
        dtype: object
```

```
In [10]: # 存成新的 CSV 檔
         df.to_csv("Churn_Modelling_new.csv", index=False)
```

新的excel檔案：



CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Churn
619	France	Female	42	2	0	1	1	1	101348.88	1
608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
502	France	Female	42	8	159660.8	3	1	0	113931.57	1
699	France	Female	39	1	0	2	0	0	93826.63	0
850	Spain	Female	43	2	125510.8	1	1	1	79084.1	0
645	Spain	Male	44	8	113755.8	2	1	0	149756.71	1
822	France	Male	50	7	0	2	1	1	10062.8	0
376	Germany	Female	29	4	115046.7	4	1	0	119346.88	1
501	France	Male	44	4	142051.1	2	0	1	74940.5	0
684	France	Male	27	2	134603.9	1	1	1	71725.73	0
528	France	Male	37	6	102016.7	2	0	0	80181.12	0
497	Spain	Male	37	3	0	2	1	0	76390.01	0
476	France	Female	37	10	0	2	1	0	26260.98	0
549	France	Female	25	5	0	2	0	0	190857.79	0
635	Spain	Female	35	7	0	2	1	1	65951.65	0
616	Germany	Male	45	3	143129.4	2	0	1	64327.26	0
653	Germany	Male	58	1	132602.9	1	1	0	5097.67	1
549	Spain	Female	24	9	0	2	1	1	14406.41	0
587	Spain	Male	45	6	0	1	0	0	158684.81	0
726	France	Female	24	6	0	2	1	1	54724.03	0
732	France	Male	41	8	0	2	1	1	170886.17	0
636	Spain	Female	32	8	0	2	1	0	138555.46	0
510	Spain	Female	38	4	0	1	1	0	118913.53	1
669	France	Male	46	3	0	2	0	1	8487.75	0
846	France	Female	38	5	0	1	1	1	187616.16	0
577	France	Male	25	3	0	2	0	1	124508.29	0
756	Germany	Male	36	2	136815.6	1	1	1	170041.95	0
571	France	Male	44	9	0	2	0	0	38433.35	0
574	Germany	Female	43	3	141349.4	1	1	1	100187.43	0
411	France	Male	29	0	59697.17	2	1	1	53483.21	0
591	Spain	Female	39	3	0	3	1	0	140469.38	1
533	France	Male	36	7	85311.7	1	0	1	156731.91	0

5. 對各個欄位進行分析，了解目前銀行客戶的概況：

(1) 對 HasCrCard 欄位進行分析，說明有多少比例的人持有信用卡，多少比例的人不持有信用卡。(3%)

```
In [11]: df.groupby(['HasCrCard']).size()

Out[11]: HasCrCard
         0    2945
         1    7055
         dtype: int64
```

```
In [12]: print('持有信用卡的人數比例 :', 7055/(2945+7055))
         print('未持有信用卡的人數比例 :', 2945/(2945+7055))

         持有信用卡的人數比例 : 0.7055
         未持有信用卡的人數比例 : 0.2945
```

(2) 對 Churn 欄位進行分析，說明有多少比例的客戶流失。(3%)

```
In [13]: df.groupby(['Churn']).size()

         # 0 = 未流失， 1 = 流失

Out[13]: Churn
         0    7963
         1    2037
         dtype: int64
```

```
In [14]: print('客戶流失比例 :', 2037/(2037+7963))

         客戶流失比例 : 0.2037
```

(3) 對 IsActiveMember 欄位進行分析，說明有多少比例的客戶仍是活躍狀態。

In [15]:
```
df.groupby(['IsActiveMember']).size()
```

Out[15]:
```
IsActiveMember
0    4849
1    5151
dtype: int64
```

In [16]:
```
print('客戶活躍比例 :', 5151/(5151+4849))
```

```
客戶活躍比例 : 0.5151
```

(4)對Churn 進行分析，觀察流失客戶跟未流失客戶的資料平均值

將Churn = 0 與 1的狀況分別用不同dataframe儲存，並使用describe()查看分別的統計數據

In [26]:
```
import copy

churn_1 = copy.deepcopy(df[df['Churn'].isin([1])])
churn_1.describe()
```

Out[26]:

	CreditScore	Age	Tenure	Balance	NumOfProducts	EstimatedSalary
count	2037.000000	2037.000000	2037.000000	2037.000000	2037.000000	2037.000000
mean	645.351497	44.837997	4.932744	91108.539337	1.475209	101465.677531
std	100.321503	9.761562	2.936106	58360.794816	0.801521	57912.418071
min	350.000000	18.000000	0.000000	0.000000	1.000000	11.580000
25%	578.000000	38.000000	2.000000	38340.020000	1.000000	51907.720000
50%	646.000000	45.000000	5.000000	109349.290000	1.000000	102460.840000
75%	716.000000	51.000000	8.000000	131433.330000	2.000000	152422.910000
max	850.000000	84.000000	10.000000	250898.090000	4.000000	199808.100000

In [18]:
```
churn_0 = copy.deepcopy(df[df['Churn'].isin([0])])
churn_0.describe()
```

Out[18]:

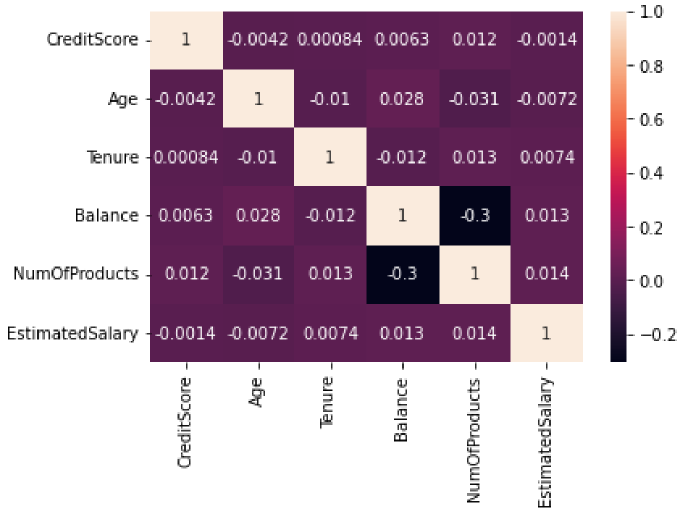
	CreditScore	Age	Tenure	Balance	NumOfProducts	EstimatedSalary
count	7963.000000	7963.000000	7963.000000	7963.000000	7963.000000	7963.000000
mean	651.853196	37.411277	5.033279	72745.296779	1.544267	99718.932023
std	95.653837	10.123714	2.880658	62848.040701	0.509536	57397.636600
min	405.000000	18.000000	0.000000	0.000000	1.000000	90.070000
25%	585.000000	31.000000	3.000000	0.000000	1.000000	50783.490000
50%	653.000000	36.000000	5.000000	92072.680000	2.000000	99645.040000
75%	718.000000	41.000000	7.000000	126410.280000	2.000000	148596.500000
max	850.000000	92.000000	10.000000	221532.800000	3.000000	199992.480000

(5)計算屬性間的相關係數，並用seaborn繪製出熱力圖(heatmap) (8%)

```
In [19]: import matplotlib.pyplot as plt
         import seaborn as sns
```

```
In [20]: corrM = df.corr()
         sns.heatmap(corrM, annot = True)
```

```
Out[20]: <AxesSubplot:>
```

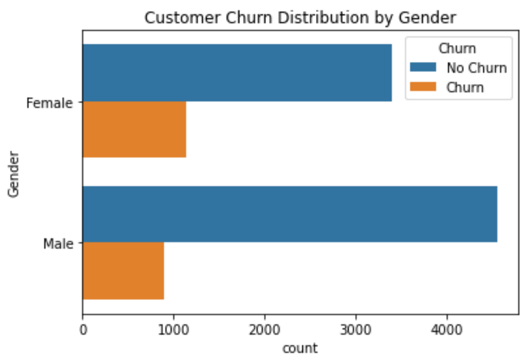


6.運用資料視覺化來幫助分析：

(1)繪出Gender與Churn 的數量關係，分析不同性別於客戶流失的關係，如下圖所示。(Hint: seaborn.countplot())(10%)

```
In [21]: churn = copy.deepcopy(df['Churn'])
         churn = churn.replace(to_replace = [0, 1], value = ['No Churn', 'Churn'])
```

```
In [22]: sns.countplot(y = df['Gender'], hue = churn)
         plt.title('Customer Churn Distribution by Gender')
         plt.show()
```

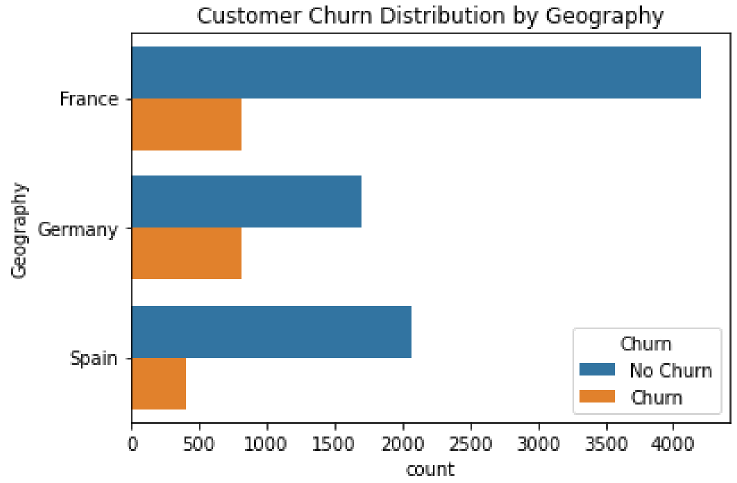


將Churn的欄位0與1分別以Churn和No Churn替代，並丟進hue中

(2)繪出Geography與Churn 的數量關係，分析不同地區於客戶流失的關係。

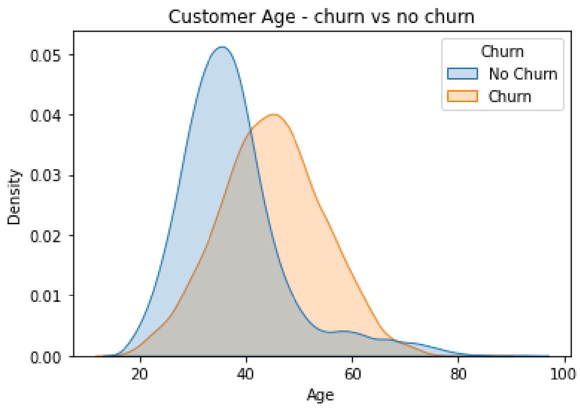
(Hint: seaborn.countplot())(5%)

In [23]:
```
sns.countplot(y = df['Geography'], hue = churn)
plt.title('Customer Churn Distribution by Geography')
plt.show()
```



(3) 繪出 Age 分布與 Churn 的關係，分析不同年齡於客戶流失率的關係，如下圖所示。 (Hint: seaborn.kdeplot()) (10%)

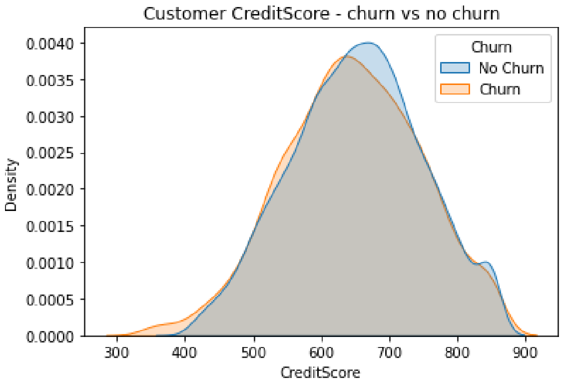
In [24]:
```
# common_norm=False  => 區域下的面積總和分別等於一

sns.kdeplot(x = df['Age'],  hue = churn , fill=True, common_norm=False)
plt.title('Customer Age - churn vs no churn')
plt.show()
```



可見年齡較低(約20~40歲)的顧客相較於年齡較高(約45歲左右)的顧客不易流失

(4) 繪出 CreditScore 與 Churn 的關係，分析客戶信用分數於客戶流失率的關係。(Hint: seaborn.kdeplot()) (7%)

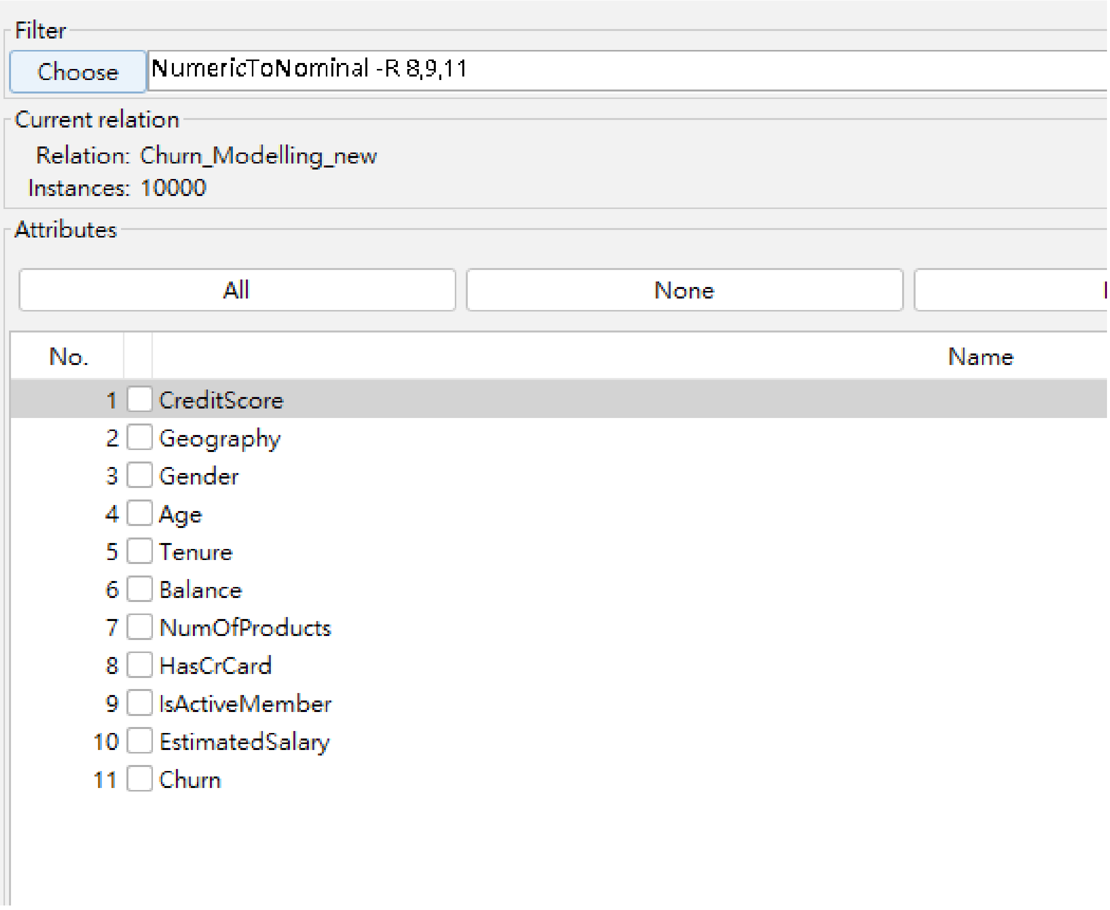
In [25]:
```
sns.kdeplot(x = df['CreditScore'],  hue = churn , fill=True, common_norm=False)
plt.title('Customer CreditScore - churn vs no churn')
plt.show()
```



可見客戶流失與否與客戶信用分數無關聯。

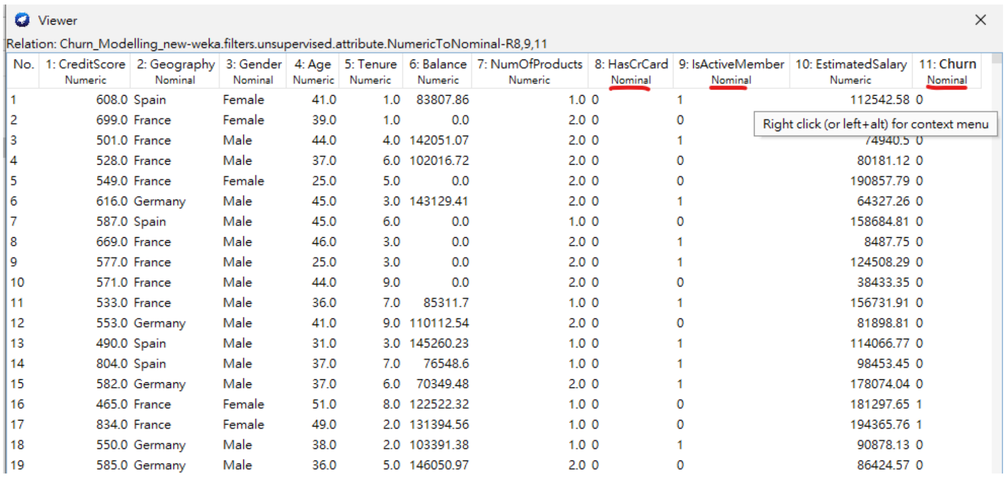
WEKA部分

(1) 將 HasCrCard, IsActiveMember, Churn 轉成 Nominal 屬性。(10%)



針對8, 9, 11欄(HasCrCard, IsActiveMember, Churn)使用NumericToNominal

結果如下：



Viewer

Relation: Churn_Modelling_new-weka.filters.unsupervised.attribute.NumericToNominal-R8,9,11

No.	1: CreditScore Numeric	2: Geography Nominal	3: Gender Nominal	4: Age Numeric	5: Tenure Numeric	6: Balance Numeric	7: NumOfProducts Numeric	8: HasCrCard Nominal	9: IsActiveMember Nominal	10: EstimatedSalary Numeric	11: Churn Nominal
1	608.0	Spain	Female	41.0	1.0	83807.86	1.0	0	1	112542.58	0
2	699.0	France	Female	39.0	1.0	0.0	2.0	0	0	[illegible]	[illegible]
3	501.0	France	Male	44.0	4.0	142051.07	2.0	0	1	74940.5	0
4	528.0	France	Male	37.0	6.0	102016.72	2.0	0	0	80181.12	0
5	549.0	France	Female	25.0	5.0	0.0	2.0	0	0	190857.79	0
6	616.0	Germany	Male	45.0	3.0	143129.41	2.0	0	1	64327.26	0
7	587.0	Spain	Male	45.0	6.0	0.0	1.0	0	0	158684.81	0
8	669.0	France	Male	46.0	3.0	0.0	2.0	0	1	8487.75	0
9	577.0	France	Male	25.0	3.0	0.0	2.0	0	1	124508.29	0
10	571.0	France	Male	44.0	9.0	0.0	2.0	0	0	38433.35	0
11	533.0	France	Male	36.0	7.0	85311.7	1.0	0	1	156731.91	0
12	553.0	Germany	Male	41.0	9.0	110112.54	2.0	0	0	81898.81	0
13	490.0	Spain	Male	31.0	3.0	145260.23	1.0	0	1	114066.77	0
14	804.0	Spain	Male	37.0	7.0	76548.6	1.0	0	1	98453.45	0
15	582.0	Germany	Male	37.0	6.0	70349.48	2.0	0	1	178074.04	0
16	465.0	France	Female	51.0	8.0	122522.32	1.0	0	0	181297.65	1
17	834.0	France	Female	49.0	2.0	131394.56	1.0	0	0	194365.76	1
18	550.0	Germany	Male	38.0	2.0	103391.38	1.0	0	1	90878.13	0
19	585.0	Germany	Male	36.0	5.0	146050.97	2.0	0	0	86424.57	0

Right click (or left+alt) for context menu

(2) 使用 Attribute Selection，以 CfsSubsetEval 及 BestFirst 來篩選屬性，並說明屬性篩選結果。 (10%)