

1.

```

In [148]: 1 import pandas as pd
          2 import numpy as np
          3 from sklearn.model_selection import cross_val_score
          4
          5 data=pd.read_csv('C:/Users/Admin/Downloads/Churn_Modelling.csv')
          6 data.head()

Out[148]:
   CustomerId  CredRate Geography Gender Age  Tenure  Balance  Prod Number  HasCrCard  ActMem  EstimatedSalary  Exited
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

In [149]: 1 data.isnull().any()

Out[149]: CustomerId      False
CredRate      False
Geography      False
Gender        True
Age           True
Tenure        False
Balance       False
Prod Number   False
HasCrCard     False
ActMem        False
EstimatedSalary  True
Exited        False
dtype: bool

```

2.

```

In [150]: 1 data.EstimatedSalary.fillna(data.EstimatedSalary.mean())
          2

Out[150]: 0      101348.88
          1      112542.58
          2      113931.57
          3      93826.63
          4      79084.10
          ...
          9995     96270.64
          9996     101699.77
          9997      42085.58
          9998      92888.52
          9999      38190.78
          Name: EstimatedSalary, Length: 10000, dtype: float64

In [151]: 1 from scipy import stats
          2 stats.mode(data['Age'])
          3 data.Age.fillna(37)

Out[151]: 0      42.0
          1      41.0
          2      42.0
          3      39.0
          4      43.0
          ...
          9995      39.0
          9996      35.0
          9997      36.0
          9998      42.0
          9999      28.0
          Name: Age, Length: 10000, dtype: float64

```

```

Out[151]: 0      42.0
          1      41.0
          2      42.0
          3      39.0
          4      43.0
          ...
          9995    39.0
          9996    35.0
          9997    36.0
          9998    42.0
          9999    28.0
          Name: Age, Length: 10000, dtype: float64

In [152]: 1 stats.mode(data['Gender'])
          2 data.Gender.fillna('Male')

Out[152]: 0      Female
          1      Female
          2      Female
          3      Female
          4      Female
          ...
          9995      Male
          9996      Male
          9997      Female
          9998      Male
          9999      Female
          Name: Gender, Length: 10000, dtype: object

In [153]: 1 data=data.rename(columns={'CredRate':'CreditScore'})
          2 data=data.rename(columns={'ActMem':'IsActiveMember'})
          3 data=data.rename(columns={'Prod Number':'NumOfProducts'})
          4 data=data.rename(columns={'Exited':'Churn'})

```

3、4

```

Name: Gender, Length: 10000, dtype: object

In [153]: 1 data=data.rename(columns={'CredRate':'CreditScore'})
          2 data=data.rename(columns={'ActMem':'IsActiveMember'})
          3 data=data.rename(columns={'Prod Number':'NumOfProducts'})
          4 data=data.rename(columns={'Exited':'Churn'})

In [154]: 1 data=data.drop(['CustomerId'],axis=1)

In [155]: 1 data['Geography'] = data['Geography'].astype('category')
          2 data['Gender'] = data['Gender'].astype('category')
          3 data['HasCrCard'] = data['HasCrCard'].astype('category')
          4 data['Churn'] = data['Churn'].astype('category')
          5 data['IsActiveMember'] = data['IsActiveMember'].astype('category')
          6 data.head()

Out[155]:
   CreditScore  Geography  Gender  Age  Tenure  Balance  NumOfProducts  HasCrCard  IsActiveMember  EstimatedSalary  Churn
0           619    France  Female  42.0      2     0.00             1           1             1         101348.88      1
1           608     Spain  Female  41.0      1    83807.86             1           0             1         112542.58      0
2           502    France  Female  42.0      8   159660.80             3           1             0         113931.57      1
3           699    France  Female  39.0      1      0.00             2           0             0          93826.63      0
4           850     Spain  Female  43.0      2  125510.82             1           1             1          79084.10      0

In [156]: 1 data.dtypes

Out[156]: CreditScore      int64
Geography      category
Gender      category
Age      float64

```

1	608	Spain	Female	41.0	1	83807.86	1	0	1	112542.58	0
2	502	France	Female	42.0	8	159660.80	3	1	0	113931.57	1
3	699	France	Female	39.0	1	0.00	2	0	0	93826.63	0
4	850	Spain	Female	43.0	2	125510.82	1	1	1	79084.10	0

In [156]: 1 data.dtypes

Out[156]: 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 [175]: 1 data.to\_csv('C:/Users/Admin/Downloads/Result222.csv',index=False)

In [158]: 1 data['HasCrCard'].value\_counts()  
2 #2945 没信用卡

Out[158]: 1 7055  
0 2945  
Name: HasCrCard, dtype: int64

In [159]: 1 data['Churn'].value\_counts()  
MySQL Workbench 8.0 CE 2 #2037 顧客流失

## 5.(1)(2)(3)(4)

In [158]: 1 data['HasCrCard'].value\_counts()  
2 #2945 没信用卡

Out[158]: 1 7055  
0 2945  
Name: HasCrCard, dtype: int64

In [159]: 1 data['Churn'].value\_counts()  
2 #2037 顧客流失

Out[159]: 0 7963  
1 2037  
Name: Churn, dtype: int64

In [160]: 1 data['IsActiveMember'].value\_counts()  
2 #5151 顧客活躍

Out[160]: 1 5151  
0 4849  
Name: IsActiveMember, dtype: int64

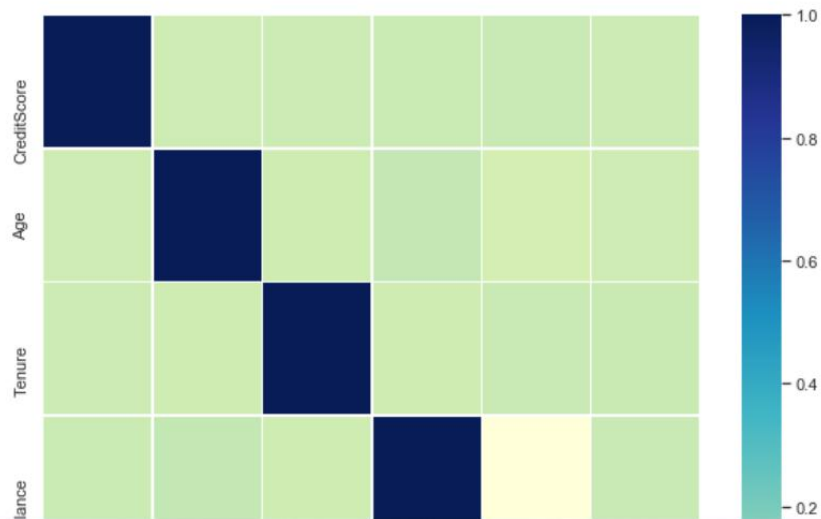
In [161]: 1 data1=data.groupby('Churn').aggregate(['mean'])  
2 data1.head()

Out[161]:

	CreditScore	Age	Tenure	Balance	NumOfProducts	EstimatedSalary
	mean	mean	mean	mean	mean	mean
Churn						
0	651.853196	37.411587	5.033279	72745.296779	1.544267	99718.753200
1	645.351497	44.837997	4.932744	91108.539337	1.475209	101465.677531

```
] 1 data2=data.corr()
```

```
] 1 import seaborn as sns
2 import matplotlib.pyplot as plt
3 %matplotlib inline
4
5 sns.set(style='whitegrid', color_codes=True)
6 plt.figure(figsize=(10,10))
7 ax = sns.heatmap(data2, fmt='d', linewidths=.5, cmap='YlGnBu')
```

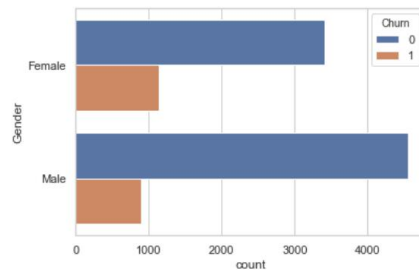


5.(5)

6.(1)(2)

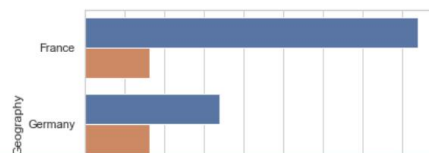
```
In [164]: 1 sns.countplot(y = 'Gender',hue = 'Churn',data=data)
```

```
Out[164]: <matplotlib.axes._subplots.AxesSubplot at 0x2af05964ec8>
```

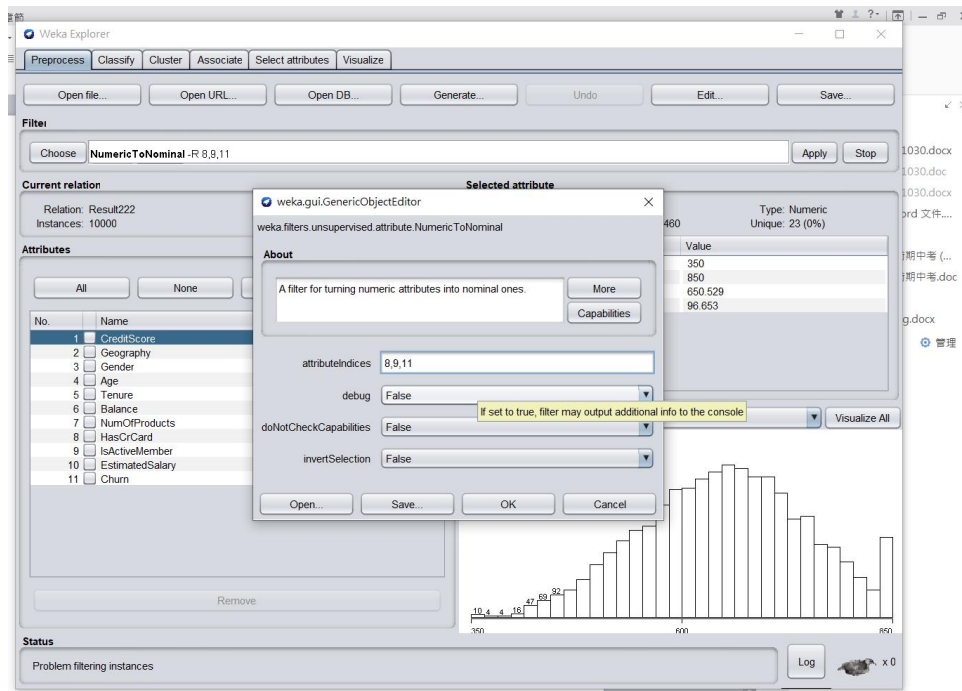


```
In [165]: 1 sns.countplot(y = 'Geography',hue = 'Churn',data=data)
```

```
Out[165]: <matplotlib.axes._subplots.AxesSubplot at 0x2af074c3cc8>
```



7.(1)



## 7.(2)

根據屬性子集中每一個特徵的預測能力以及它們之間的關聯性進行評估，並採用 greedy algo 選擇當前最好的屬性，根據結果 churn，選取 6 種預測能力最好的屬性(最有代表性)

