ETC_HW6_107403020 李泳輝

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')
df.head()
```

Out[2]:

	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 [4]:
```
df.isnull().sum()
```

Out[4]:
```
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 [5]:
```
df['EstimatedSalary'] = df['EstimatedSalary'].fillna(df['EstimatedSalary'].mean())
df['Age'] = df['Age'].fillna(df['Age'].mode()[0])
df['Gender'] = df['Gender'].fillna(df['Gender'].mode()[0])
```

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
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 [7]:
```
df.rename(columns={'CredRate' : 'CreditScore',
          'ActMem' : 'IsActiveMember',
          'Prod Number' : 'NumOfProducts',
          'Exited' : 'Churn'}, inplace=True)
```

In [8]:
```
df.head()
```

Out[8]:

	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 [9]:
```
df.drop(['CustomerId'], axis=1, inplace=True)
df.head()
```

Out[9]:

	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 [11]:
```
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')
```

In [12]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   CreditScore      10000 non-null  int64
 1   Geography        10000 non-null  category
 2   Gender           10000 non-null  category
 3   Age              10000 non-null  float64
 4   Tenure           10000 non-null  int64
 5   Balance          10000 non-null  float64
 6   NumOfProducts    10000 non-null  int64
 7   HasCrCard        10000 non-null  category
 8   IsActiveMember   10000 non-null  category
 9   EstimatedSalary  10000 non-null  float64
 10  Churn            10000 non-null  category
dtypes: category(5), float64(3), int64(3)
memory usage: 518.3 KB
```

In [13]:
```
df.to_csv("ETC_HW6_107403020.csv", index=False)
```

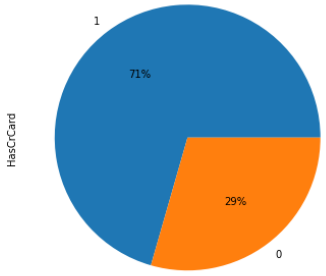
5. 對各個欄位進行分析，了解目前銀行客戶的概況：

(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 [14]:
```
df['HasCrCard'].value_counts()
```

Out[14]:
```
1    7055
0    2945
Name: HasCrCard, dtype: int64
```

In [26]:
```
plt.figure(figsize=(8, 6))
ax = df['HasCrCard'].value_counts().plot.pie(autopct='%1.0f%%')
```

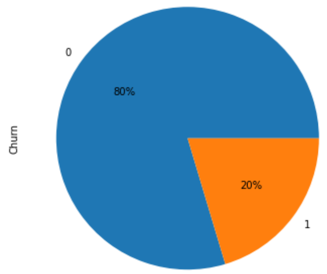


(2) 對 Churn 欄位進行分析，說明有多少比例的客戶流失。(3%)

In [27]:
```
df['Churn'].value_counts()
```

Out[27]:
```
0    7963
1    2037
Name: Churn, dtype: int64
```

In [28]:
```
plt.figure(figsize=(8, 6))
ax = df['Churn'].value_counts().plot.pie(autopct='%1.0f%%')
```

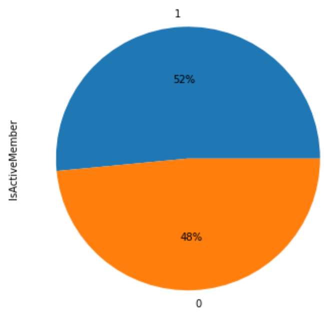


(3) 對 IsActiveMember 欄位進行分析，說明有多少比例的客戶仍是活躍狀態。(3%)

In [29]:
```
df['IsActiveMember'].value_counts()
```

Out[29]:
```
1    5151
0    4849
Name: IsActiveMember, dtype: int64
```

In [30]:
```
plt.figure(figsize=(8, 6))
ax = df['IsActiveMember'].value_counts().plot.pie(autopct='%1.0f%%')
```



(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(6%)

In [31]:
```
df.groupby('Churn').mean()
```

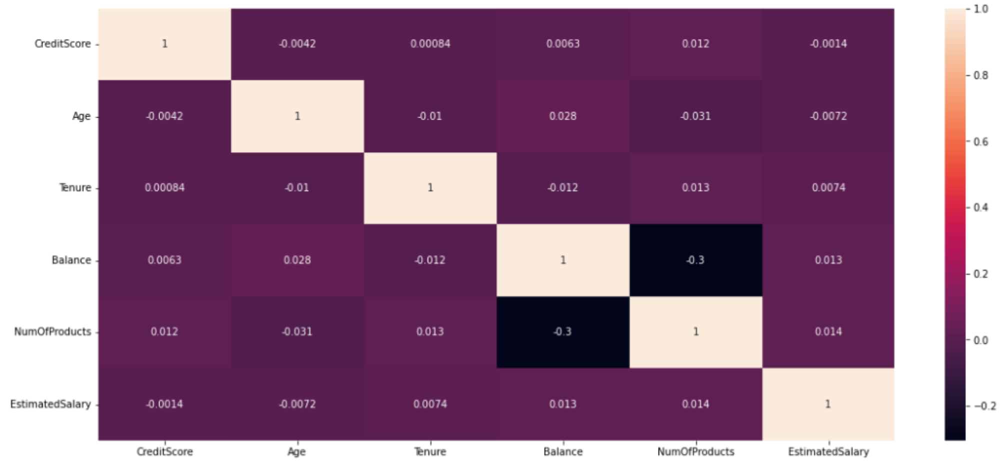
Out[31]:

	CreditScore	Age	Tenure	Balance	NumOfProducts	EstimatedSalary
Churn						
0	651.853196	37.411277	5.033279	72745.296779	1.544267	99718.932023
1	645.351497	44.837997	4.932744	91108.539337	1.475209	101465.677531

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

In [32]:
```
import seaborn as sns
```

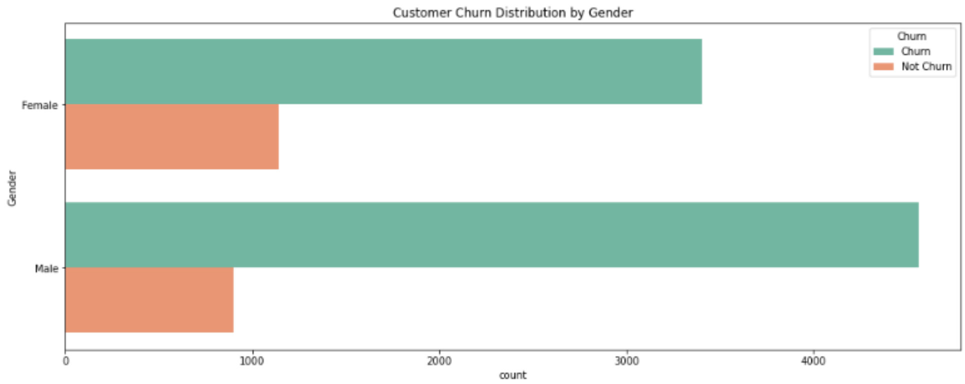
In [110]:
```
columns = ['CreditScore', 'Age', 'Tenure', 'Balance', 'NumOfProducts', 'EstimatedSalary']
tmp = df[columns]
corr = tmp.corr()
plt.figure(figsize=(18, 8))
sns.heatmap(corr, annot=True)
plt.xticks(rotation=0)
plt.show()
```



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%)

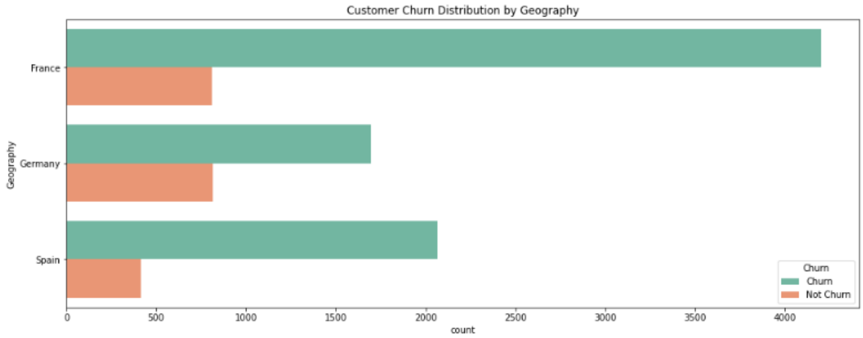
In [25]:
```
plt.figure(figsize=(16, 6))
ax = sns.countplot(data=df, y='Gender', hue='Churn', palette='Set2')
plt.title('Customer Churn Distribution by Gender')
plt.legend(["Churn", "Not Churn"], title='Churn')
plt.xlabel("count")
plt.ylabel("Gender")
plt.show()
```



女性客戶的比例明顯高於男性客戶。

(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%)

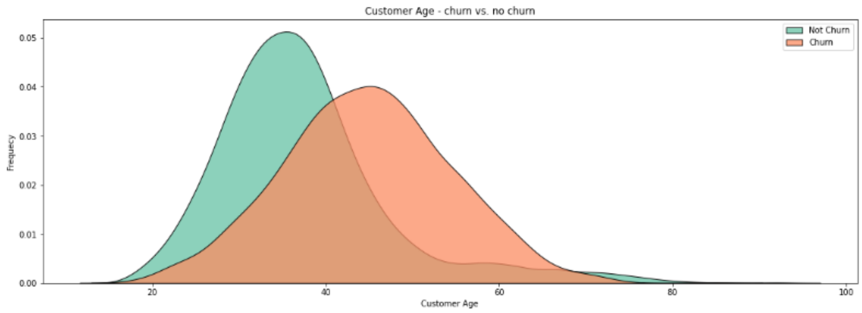
In [26]:
```
plt.figure(figsize=(16, 6))
ax = sns.countplot(data=df, y='Geography', hue='Churn', palette='Set2')
plt.title('Customer Churn Distribution by Geography')
plt.legend(["Churn", "Not Churn"], title='Churn')
plt.xlabel("count")
plt.ylabel("Geography")
plt.show()
```



France 的客戶人數最多，而 Germany 客戶流失的比例最高。

(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%)

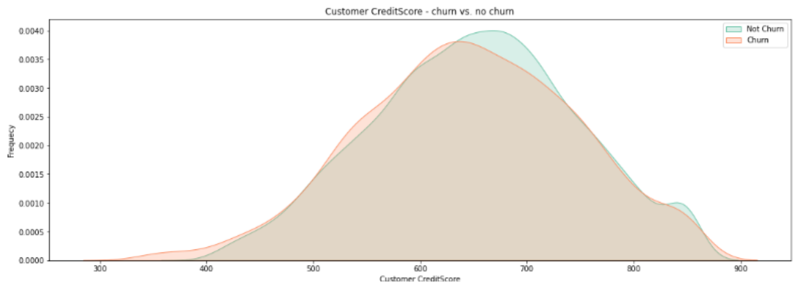
In [27]:
```
plt.figure(figsize=(18, 6))
ax = sns.set_palette('Set2')
ax1 = sns.kdeplot(data=df[df['Churn']==0], x='Age', hue='Churn', multiple='stack')
ax2 = sns.kdeplot(data=df[df['Churn']==1], x='Age', hue='Churn', multiple='stack')
plt.title('Customer Age - churn vs. no churn')
plt.legend(["Not Churn", "Churn"])
plt.xlabel('Customer Age')
plt.ylabel('Frequecy')
plt.show()
```



流失的客戶年齡平均較高。

(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%)

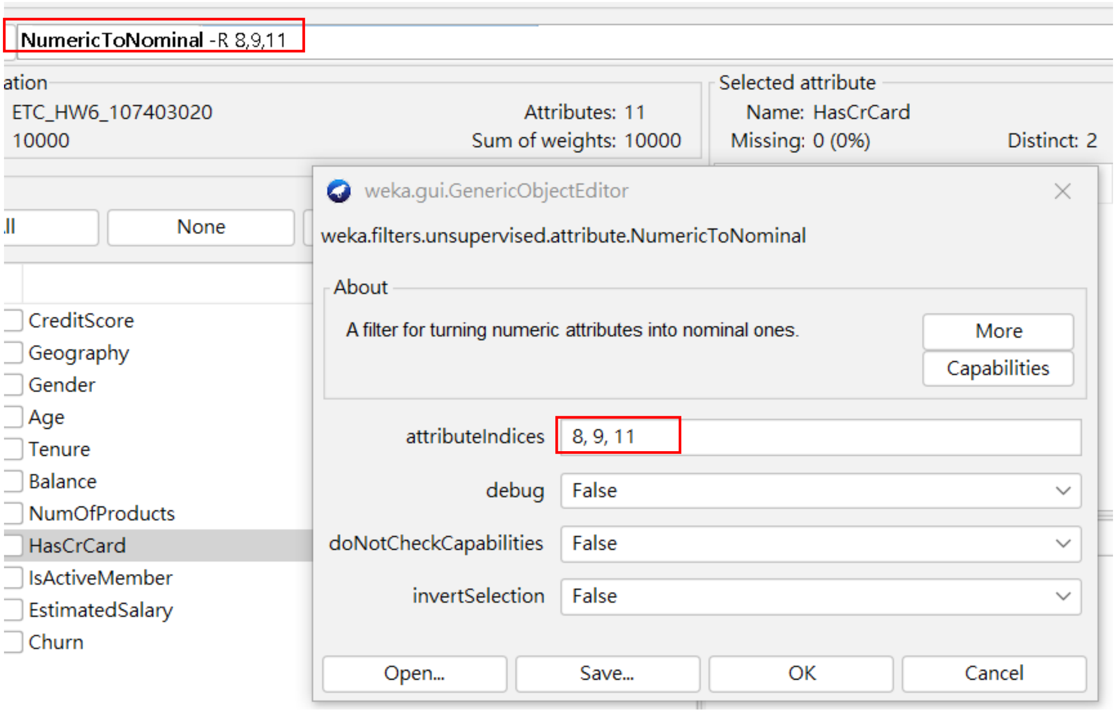
In [29]:
```
plt.figure(figsize=(18, 6))
ax = sns.set_palette('Set2')
ax1 = sns.kdeplot(data=df[df['Churn']==0], x='CreditScore', shade=True, hue='Churn')
ax2 = sns.kdeplot(data=df[df['Churn']==1], x='CreditScore', shade=True, hue='Churn')
plt.title('Customer CreditScore - churn vs. no churn')
plt.legend(["Not Churn", "Churn"])
plt.xlabel('Customer CreditScore')
plt.ylabel('Frequecy')
plt.show()
```



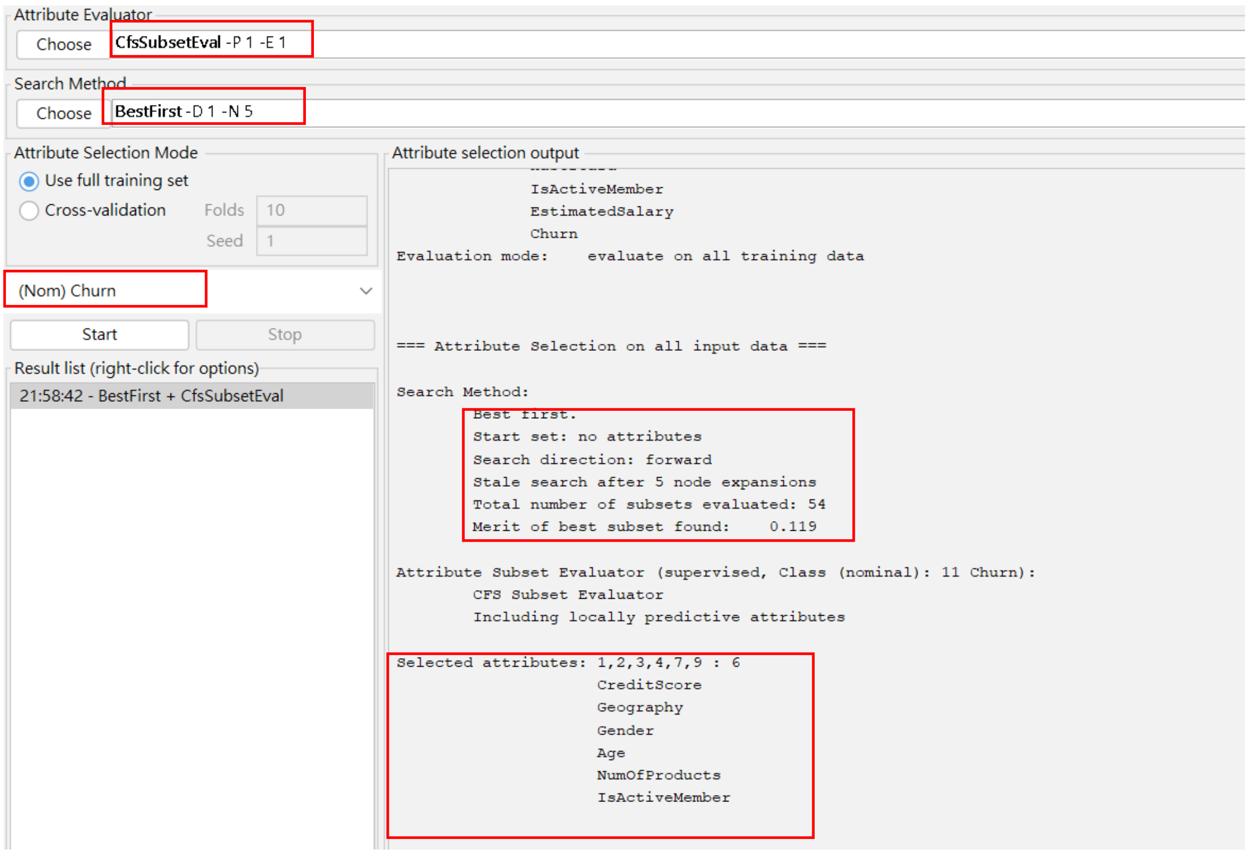
未流失的客戶群體有較高的信用分數。

WEKA

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



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



BestFirst 的預設是 forward，也就是從沒有 attributes 開始挑選屬性。
Total number of subsets evaluated 代表的是算的次數。
這題就後挑選出的屬性由最好到最低依序為 CreditScore, Geography, Gender, Age, NumOfProducts, IsActive，共 6 個。