Mula Anvesh Reddy 03-07-2021 Assignment-3

In [1]:
```
#import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:
```
#reading the dataset
dataset=pd.read_csv(r'C:\Users\Anvesh Mula\OneDrive\Desktop\Internship\Churn_Modelling1.csv')
```

In [3]:
```
dataset
```

Out[3]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActive
0	1	15634602	Hargrave	619	France	Female	42.0	2	0.00	1	1	
1	2	15647311	Hill	608	Spain	Female	41.0	1	83807.86	1	0	
2	3	15619304	Onio	502	France	Female	42.0	8	159660.80	3	1	
3	4	15701354	Boni	699	France	Female	39.0	1	0.00	2	0	
4	5	15737888	Mitchell	850	Spain	Female	NaN	2	125510.82	1	1	
5	6	15574012	Chu	645	Spain	Male	44.0	8	113755.78	2	1	
6	7	15592531	Bartlett	822	France	NaN	50.0	7	0.00	2	1	
7	8	15656148	Obinna	376	Germany	Female	29.0	4	115046.74	4	1	
8	9	15792365	He	501	France	Male	44.0	4	142051.07	2	0	
9	10	15592389	H?	684	France	Male	27.0	2	134603.88	1	1	
10	11	15767821	Bearce	528	France	Male	31.0	6	NaN	2	0	
11	12	15737173	Andrews	497	Spain	Male	24.0	3	0.00	2	1	
12	13	15632264	Kay	476	France	Female	34.0	10	0.00	2	1	
13	14	15691483	Chin	549	France	Female	25.0	5	0.00	2	0	
14	15	15600882	Scott	635	Spain	Female	35.0	7	0.00	2	1	
15	16	15643966	Goforth	616	Germany	Male	45.0	3	143129.41	2	0	
16	17	15737452	Romeo	653	NaN	Male	58.0	1	132602.88	1	1	
17	18	15788218	Henderson	549	Spain	Female	24.0	9	0.00	2	1	
18	19	15661507	Muldrow	587	Spain	Male	45.0	6	0.00	1	0	
19	20	15568982	Hao	726	France	Female	24.0	6	0.00	2	1	
20	21	15577657	McDonald	732	France	Male	41.0	8	0.00	2	1	
21	22	15597945	Dellucci	636	Spain	NaN	32.0	8	0.00	2	1	
22	23	15699309	Gerasimov	510	Spain	Female	38.0	4	0.00	1	1	
23	24	15725737	Mosman	669	France	Male	46.0	3	0.00	2	0	
24	25	15625047	Yen	846	France	Female	38.0	5	0.00	1	1	

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActive
25	26	15738191	Maclean	577	France	Male	25.0	3	0.00	2	0	
26	27	15736816	Young	756	Germany	Male	36.0	2	NaN	1	1	
27	28	15700772	Nebechi	571	France	Male	44.0	9	0.00	2	0	
28	29	15728693	McWilliams	574	Germany	Female	NaN	3	141349.43	1	1	
29	30	15656300	Lucciano	411	France	Male	29.0	0	59697.17	2	1	
...	...	...	...	...	...	...	...	...	...	...	...	
9970	9971	15587133	Thompson	518	France	Male	42.0	7	151027.05	2	1	
9971	9972	15721377	Chou	833	France	Female	34.0	3	144751.81	1	0	
9972	9973	15747927	Ch'in	758	France	Male	26.0	4	155739.76	1	1	
9973	9974	15806455	Miller	611	France	Male	27.0	7	0.00	2	1	
9974	9975	15695474	Barker	583	France	Male	33.0	7	122531.86	1	1	
9975	9976	15666295	Smith	610	Germany	Male	50.0	1	113957.01	2	1	
9976	9977	15656062	Azikiwe	637	France	Female	33.0	7	103377.81	1	1	
9977	9978	15579969	Mancini	683	France	Female	32.0	9	0.00	2	1	
9978	9979	15703563	P'eng	774	France	Male	40.0	9	93017.47	2	1	
9979	9980	15692664	Diribe	677	France	Female	58.0	1	90022.85	1	0	
9980	9981	15719276	T'ao	741	Spain	Male	35.0	6	74371.49	1	0	
9981	9982	15672754	Burbidge	498	Germany	Male	42.0	3	152039.70	1	1	
9982	9983	15768163	Griffin	655	Germany	Female	46.0	7	137145.12	1	1	
9983	9984	15656710	Cocci	613	France	Male	40.0	4	0.00	1	0	
9984	9985	15696175	Echezonachukwu	602	Germany	Male	35.0	7	90602.42	2	1	
9985	9986	15586914	Nepean	659	France	Male	36.0	6	123841.49	2	1	
9986	9987	15581736	Bartlett	673	Germany	Male	47.0	1	183579.54	2	0	
9987	9988	15588839	Mancini	606	Spain	Male	30.0	8	180307.73	2	1	
9988	9989	15589329	Pirozzi	775	France	Male	30.0	4	0.00	2	1	
9989	9990	15605622	McMillan	841	Spain	Male	28.0	4	0.00	2	1	

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActive
9990	9991	15798964	Nkemakonam	714	Germany	Male	33.0	3	35016.60	1	1	
9991	9992	15769959	Ajuluchukwu	597	France	Female	53.0	4	88381.21	1	1	
9992	9993	15657105	Chukwualuka	726	Spain	Male	36.0	2	0.00	1	1	
9993	9994	15569266	Rahman	644	France	Male	28.0	7	155060.41	1	1	
9994	9995	15719294	Wood	800	France	Female	29.0	2	0.00	2	0	
9995	9996	15606229	Obijiaku	771	France	Male	39.0	5	0.00	2	1	
9996	9997	15569892	Johnstone	516	France	Male	35.0	10	57369.61	1	1	
9997	9998	15584532	Liu	709	France	Female	36.0	7	0.00	1	0	
9998	9999	15682355	Sabbatini	772	Germany	Male	42.0	3	75075.31	2	1	
9999	10000	15628319	Walker	792	France	Female	28.0	4	130142.79	1	1	

10000 rows × 14 columns

In [4]:
```
type(dataset)
```

Out[4]:
```
pandas.core.frame.DataFrame
```

In [5]:
```
dataset.isnull().any()
```

Out[5]:
```
RowNumber          False
CustomerId         False
Surname            False
CreditScore        False
Geography           True
Gender              True
Age                 True
Tenure             False
Balance             True
NumOfProducts      False
HasCrCard          False
IsActiveMember     False
EstimatedSalary    False
Exited             False
dtype: bool
```

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

Out[6]:
```
RowNumber          0
CustomerId         0
Surname            0
CreditScore        0
Geography          3
Gender             3
Age                4
Tenure             0
Balance            2
NumOfProducts      0
HasCrCard          0
IsActiveMember     0
EstimatedSalary    0
Exited             0
dtype: int64
```

In [7]:
```
dataset[dataset['Age'].isnull()].index.tolist()
```

Out[7]:
```
[4, 28, 43, 59]
```

In [8]:
```
dataset[dataset['Gender'].isnull()].index.tolist()
```

Out[8]:
```
[6, 21, 32]
```

In [9]:
```
dataset[dataset['Geography'].isnull()].index.tolist()
```

Out[9]:
```
[16, 30, 41]
```

In [10]:
```
dataset[dataset['Balance'].isnull()].index.tolist()
```

Out[10]:
```
[10, 26]
```

In [11]:
```
dataset['Age'].fillna(dataset['Age'].mean(),inplace=True)
```

In [12]:
```
dataset['Balance'].fillna(dataset['Balance'].mean(),inplace=True)
```

In [13]:
```
dataset.isnull().any()
```

Out[13]:
```
RowNumber          False
CustomerId         False
Surname            False
CreditScore        False
Geography           True
Gender              True
Age                False
Tenure             False
Balance            False
NumOfProducts      False
HasCrCard          False
IsActiveMember     False
EstimatedSalary    False
Exited             False
dtype: bool
```

In [14]:
```
#Geography and Gender are non-numerical or categorical values
dataset['Geography']=dataset['Geography'].fillna(dataset['Geography'].mode()[0])
```

In [15]:
```
dataset['Gender']=dataset['Gender'].fillna(dataset['Gender'].mode()[0])
```

In [16]:
```
dataset.isnull().any()
```

Out[16]:
```
RowNumber          False
CustomerId         False
Surname            False
CreditScore        False
Geography          False
Gender             False
Age                False
Tenure             False
Balance            False
NumOfProducts      False
HasCrCard          False
IsActiveMember     False
EstimatedSalary    False
Exited             False
dtype: bool
```

In [17]:
```
dataset
```

Out[17]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard
0	1	15634602	Hargrave	619	France	Female	42.000000	2	0.000000	1	1
1	2	15647311	Hill	608	Spain	Female	41.000000	1	83807.860000	1	0
2	3	15619304	Onio	502	France	Female	42.000000	8	159660.800000	3	1
3	4	15701354	Boni	699	France	Female	39.000000	1	0.000000	2	0
4	5	15737888	Mitchell	850	Spain	Female	38.918768	2	125510.820000	1	1
5	6	15574012	Chu	645	Spain	Male	44.000000	8	113755.780000	2	1
6	7	15592531	Bartlett	822	France	Male	50.000000	7	0.000000	2	1
7	8	15656148	Obinna	376	Germany	Female	29.000000	4	115046.740000	4	1
8	9	15792365	He	501	France	Male	44.000000	4	142051.070000	2	0
9	10	15592389	H?	684	France	Male	27.000000	2	134603.880000	1	1
10	11	15767821	Bearce	528	France	Male	31.000000	6	76477.301512	2	0
11	12	15737173	Andrews	497	Spain	Male	24.000000	3	0.000000	2	1
12	13	15632264	Kay	476	France	Female	34.000000	10	0.000000	2	1
13	14	15691483	Chin	549	France	Female	25.000000	5	0.000000	2	0
14	15	15600882	Scott	635	Spain	Female	35.000000	7	0.000000	2	1
15	16	15643966	Goforth	616	Germany	Male	45.000000	3	143129.410000	2	0
16	17	15737452	Romeo	653	France	Male	58.000000	1	132602.880000	1	1
17	18	15788218	Henderson	549	Spain	Female	24.000000	9	0.000000	2	1
18	19	15661507	Muldrow	587	Spain	Male	45.000000	6	0.000000	1	0
19	20	15568982	Hao	726	France	Female	24.000000	6	0.000000	2	1
20	21	15577657	McDonald	732	France	Male	41.000000	8	0.000000	2	1
21	22	15597945	Dellucci	636	Spain	Male	32.000000	8	0.000000	2	1
22	23	15699309	Gerasimov	510	Spain	Female	38.000000	4	0.000000	1	1
23	24	15725737	Mosman	669	France	Male	46.000000	3	0.000000	2	0
24	25	15625047	Yen	846	France	Female	38.000000	5	0.000000	1	1

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard
25	26	15738191	Maclean	577	France	Male	25.000000	3	0.000000	2	0
26	27	15736816	Young	756	Germany	Male	36.000000	2	76477.301512	1	1
27	28	15700772	Nebechi	571	France	Male	44.000000	9	0.000000	2	0
28	29	15728693	McWilliams	574	Germany	Female	38.918768	3	141349.430000	1	1
29	30	15656300	Lucciano	411	France	Male	29.000000	0	59697.170000	2	1
...	...	...	...	...	...	...	...	...	...	...	...
9970	9971	15587133	Thompson	518	France	Male	42.000000	7	151027.050000	2	1
9971	9972	15721377	Chou	833	France	Female	34.000000	3	144751.810000	1	0
9972	9973	15747927	Ch'in	758	France	Male	26.000000	4	155739.760000	1	1
9973	9974	15806455	Miller	611	France	Male	27.000000	7	0.000000	2	1
9974	9975	15695474	Barker	583	France	Male	33.000000	7	122531.860000	1	1
9975	9976	15666295	Smith	610	Germany	Male	50.000000	1	113957.010000	2	1
9976	9977	15656062	Azikiwe	637	France	Female	33.000000	7	103377.810000	1	1
9977	9978	15579969	Mancini	683	France	Female	32.000000	9	0.000000	2	1
9978	9979	15703563	P'eng	774	France	Male	40.000000	9	93017.470000	2	1
9979	9980	15692664	Diribe	677	France	Female	58.000000	1	90022.850000	1	0
9980	9981	15719276	T'ao	741	Spain	Male	35.000000	6	74371.490000	1	0
9981	9982	15672754	Burbidge	498	Germany	Male	42.000000	3	152039.700000	1	1
9982	9983	15768163	Griffin	655	Germany	Female	46.000000	7	137145.120000	1	1
9983	9984	15656710	Cocci	613	France	Male	40.000000	4	0.000000	1	0
9984	9985	15696175	Echezonachukwu	602	Germany	Male	35.000000	7	90602.420000	2	1
9985	9986	15586914	Nepean	659	France	Male	36.000000	6	123841.490000	2	1
9986	9987	15581736	Bartlett	673	Germany	Male	47.000000	1	183579.540000	2	0
9987	9988	15588839	Mancini	606	Spain	Male	30.000000	8	180307.730000	2	1
9988	9989	15589329	Pirozzi	775	France	Male	30.000000	4	0.000000	2	1
9989	9990	15605622	McMillan	841	Spain	Male	28.000000	4	0.000000	2	1

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard
9990	9991	15798964	Nkemakonam	714	Germany	Male	33.000000	3	35016.600000	1	1
9991	9992	15769959	Ajuluchukwu	597	France	Female	53.000000	4	88381.210000	1	1
9992	9993	15657105	Chukwualuka	726	Spain	Male	36.000000	2	0.000000	1	1
9993	9994	15569266	Rahman	644	France	Male	28.000000	7	155060.410000	1	1
9994	9995	15719294	Wood	800	France	Female	29.000000	2	0.000000	2	0
9995	9996	15606229	Obijiaku	771	France	Male	39.000000	5	0.000000	2	1
9996	9997	15569892	Johnstone	516	France	Male	35.000000	10	57369.610000	1	1
9997	9998	15584532	Liu	709	France	Female	36.000000	7	0.000000	1	0
9998	9999	15682355	Sabbatini	772	Germany	Male	42.000000	3	75075.310000	2	1
9999	10000	15628319	Walker	792	France	Female	28.000000	4	130142.790000	1	1

10000 rows × 14 columns

In [18]:
```
dataset['Age']=dataset['Age'].round()
```

In [20]:
```
dataset['Gender'].mode()
```

Out[20]:
```
0    Male
dtype: object
```

In [21]:
```
dataset['Gender'][6]
```

Out[21]:
```
'Male'
```

In [22]:
```
dataset['Surname'].unique
```

Out[22]: <bound method Series.unique of 0 Hargrave
1 Hill
2 Onio
3 Boni
4 Mitchell
5 Chu
6 Bartlett
7 Obinna
8 He
9 H?
10 Bearce
11 Andrews
12 Kay
13 Chin
14 Scott
15 Goforth
16 Romeo
17 Henderson
18 Muldrow
19 Hao
20 McDonald
21 Dellucci
22 Gerasimov
23 Mosman
24 Yen
25 Maclean
26 Young
27 Nebechi
28 McWilliams
29 Lucciano
...
9970 Thompson
9971 Chou
9972 Ch'in
9973 Miller
9974 Barker
9975 Smith
9976 Azikiwe
9977 Mancini
9978 P'eng
9979 Diribe

```
9980             T'ao
9981          Burbidge
9982           Griffin
9983             Cocci
9984    Echezonachukwu
9985            Nepean
9986          Bartlett
9987           Mancini
9988           Pirozzi
9989          McMillan
9990        Nkemakonam
9991       Ajuluchukwu
9992       Chukwualuka
9993            Rahman
9994              Wood
9995          Obijiaku
9996         Johnstone
9997               Liu
9998          Sabbatini
9999             Walker
Name: Surname, Length: 10000, dtype: object>
```

In [23]:
```
dataset['Surname'].unique()
```

Out[23]:
```
array(['Hargrave', 'Hill', 'Onio', ..., 'Kashiwagi', 'Aldridge',
       'Burbidge'], dtype=object)
```

In [24]:
```
dataset['Geography'].unique()
```

Out[24]:
```
array(['France', 'Spain', 'Germany'], dtype=object)
```

In [25]:
```
dataset['Gender'].unique()
```

Out[25]:
```
array(['Female', 'Male'], dtype=object)
```

In [26]:
```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
dataset['Geography']=le.fit_transform(dataset['Geography'])
dataset['Gender']=le.fit_transform(dataset['Gender'])
dataset['Surname']=le.fit_transform(dataset['Surname'])
```

In [28]:
```
x = dataset.iloc[:,0:13].values
```

In [29]:
```
x
```

Out[29]:
```
array([[1.0000000e+00, 1.5634602e+07, 1.1150000e+03, ..., 1.0000000e+00,
        1.0000000e+00, 1.0134888e+05],
       [2.0000000e+00, 1.5647311e+07, 1.1770000e+03, ..., 0.0000000e+00,
        1.0000000e+00, 1.1254258e+05],
       [3.0000000e+00, 1.5619304e+07, 2.0400000e+03, ..., 1.0000000e+00,
        0.0000000e+00, 1.1393157e+05],
       ...,
       [9.9980000e+03, 1.5584532e+07, 1.5700000e+03, ..., 0.0000000e+00,
        1.0000000e+00, 4.2085580e+04],
       [9.9990000e+03, 1.5682355e+07, 2.3450000e+03, ..., 1.0000000e+00,
        0.0000000e+00, 9.2888520e+04],
       [1.0000000e+04, 1.5628319e+07, 2.7510000e+03, ..., 1.0000000e+00,
        0.0000000e+00, 3.8190780e+04]])
```

In [30]:
```
x=np.array(x)
```

In [31]:
```
x
```

Out[31]:
```
array([[1.0000000e+00, 1.5634602e+07, 1.1150000e+03, ..., 1.0000000e+00,
        1.0000000e+00, 1.0134888e+05],
       [2.0000000e+00, 1.5647311e+07, 1.1770000e+03, ..., 0.0000000e+00,
        1.0000000e+00, 1.1254258e+05],
       [3.0000000e+00, 1.5619304e+07, 2.0400000e+03, ..., 1.0000000e+00,
        0.0000000e+00, 1.1393157e+05],
       ...,
       [9.9980000e+03, 1.5584532e+07, 1.5700000e+03, ..., 0.0000000e+00,
        1.0000000e+00, 4.2085580e+04],
       [9.9990000e+03, 1.5682355e+07, 2.3450000e+03, ..., 1.0000000e+00,
        0.0000000e+00, 9.2888520e+04],
       [1.0000000e+04, 1.5628319e+07, 2.7510000e+03, ..., 1.0000000e+00,
        0.0000000e+00, 3.8190780e+04]])
```

In [32]:
```
y=dataset.iloc[:,-1:].values
```

In [33]:
```
y
```

Out[33]:
```
array([[1],
       [0],
       [1],
       ...,
       [1],
       [1],
       [0]], dtype=int64)
```

In [34]:
```
#convert numerical data into binary data using onehotencoder
from sklearn.preprocessing import OneHotEncoder
oh=OneHotEncoder()
```

In [37]:
```
z=oh.fit_transform(x[:,0:1]).toarray()
z
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\preprocessing\_encoders.py:371: FutureWarning: The handling of int
eger data will change in version 0.22. Currently, the categories are determined based on the range [0, max(values)],
while in the future they will be determined based on the unique values.
If you want the future behaviour and silence this warning, you can specify "categories='auto'".
In case you used a LabelEncoder before this OneHotEncoder to convert the categories to integers, then you can now use
the OneHotEncoder directly.
  warnings.warn(msg, FutureWarning)
```

Out[37]:
```
array([[1., 0., 0., ..., 0., 0., 0.],
       [0., 1., 0., ..., 0., 0., 0.],
       [0., 0., 1., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 1., 0., 0.],
       [0., 0., 0., ..., 0., 1., 0.],
       [0., 0., 0., ..., 0., 0., 1.]])
```

In [38]:
```
x=np.concatenate((x,z),axis=1)
```

In [39]:
```
x.shape
```

Out[39]:
```
(10000, 10013)
```

In [40]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```

In [41]:
```
x_train.shape
```

Out[41]: (8000, 10013)

In [42]:
```
x_test.shape
```

Out[42]: (2000, 10013)

In [43]:
```
y_train.shape
```

Out[43]: (8000, 1)

In [44]:
```
y_test.shape
```

Out[44]: (2000, 1)

In [ ]:
```

```