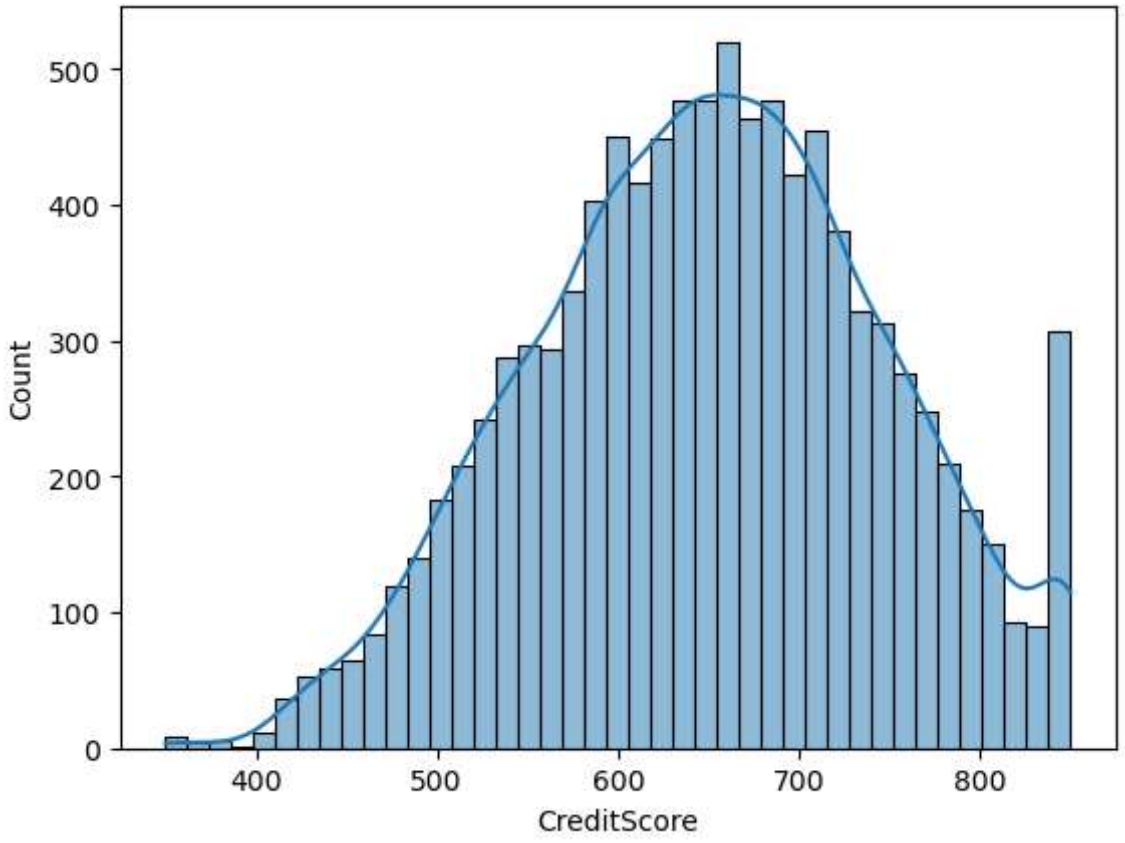
In [2]:
```
import numpy as np
import pandas as pd
df = pd.read_csv("C:\\Users\\ssdha\\Downloads\\Churn_Modelling.csv")
```

In [3]:
```
 import seaborn as sns
sns.histplot(df.CreditScore,kde=True)
```

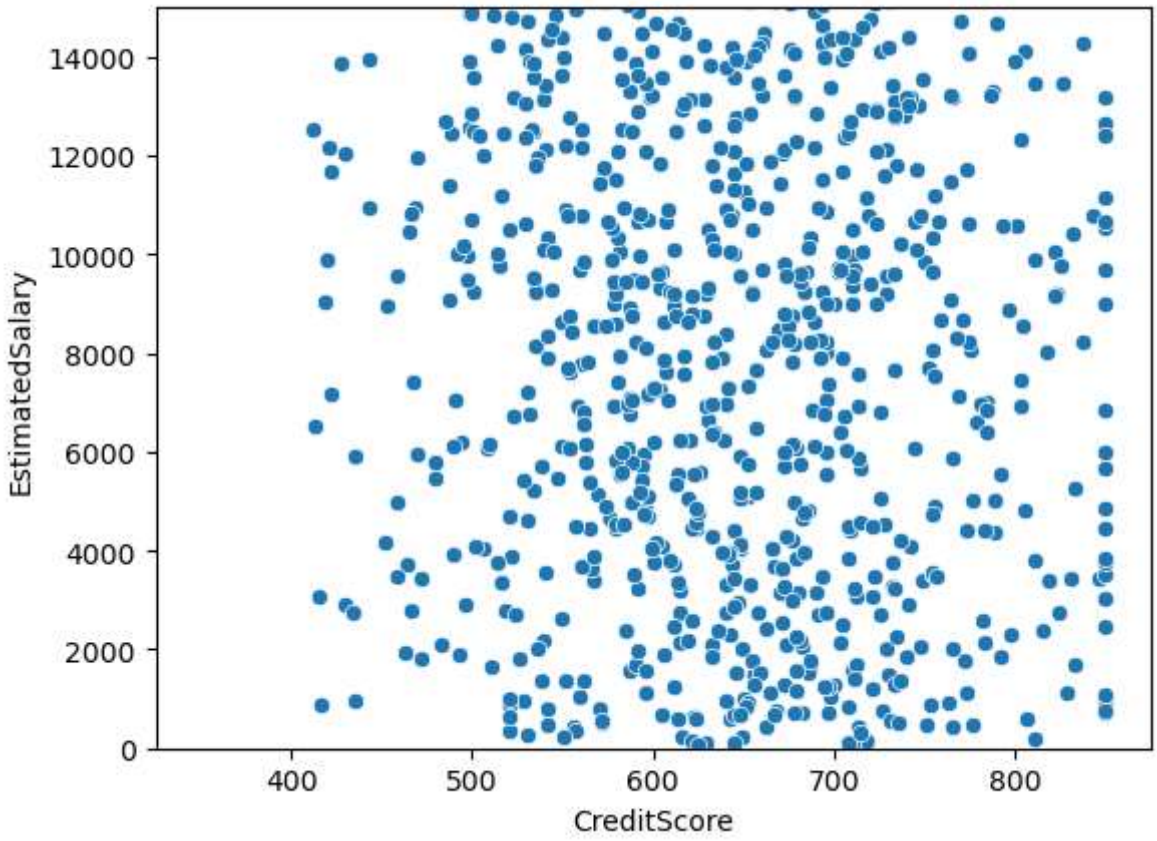
Out[3]:
```
<AxesSubplot:xlabel='CreditScore', ylabel='Count'>
```



In [4]:
```
import seaborn as sns
import matplotlib.pyplot as plt
sns.scatterplot(df.CreditScore,df.EstimatedSalary)
plt.ylim(0,15000)
```

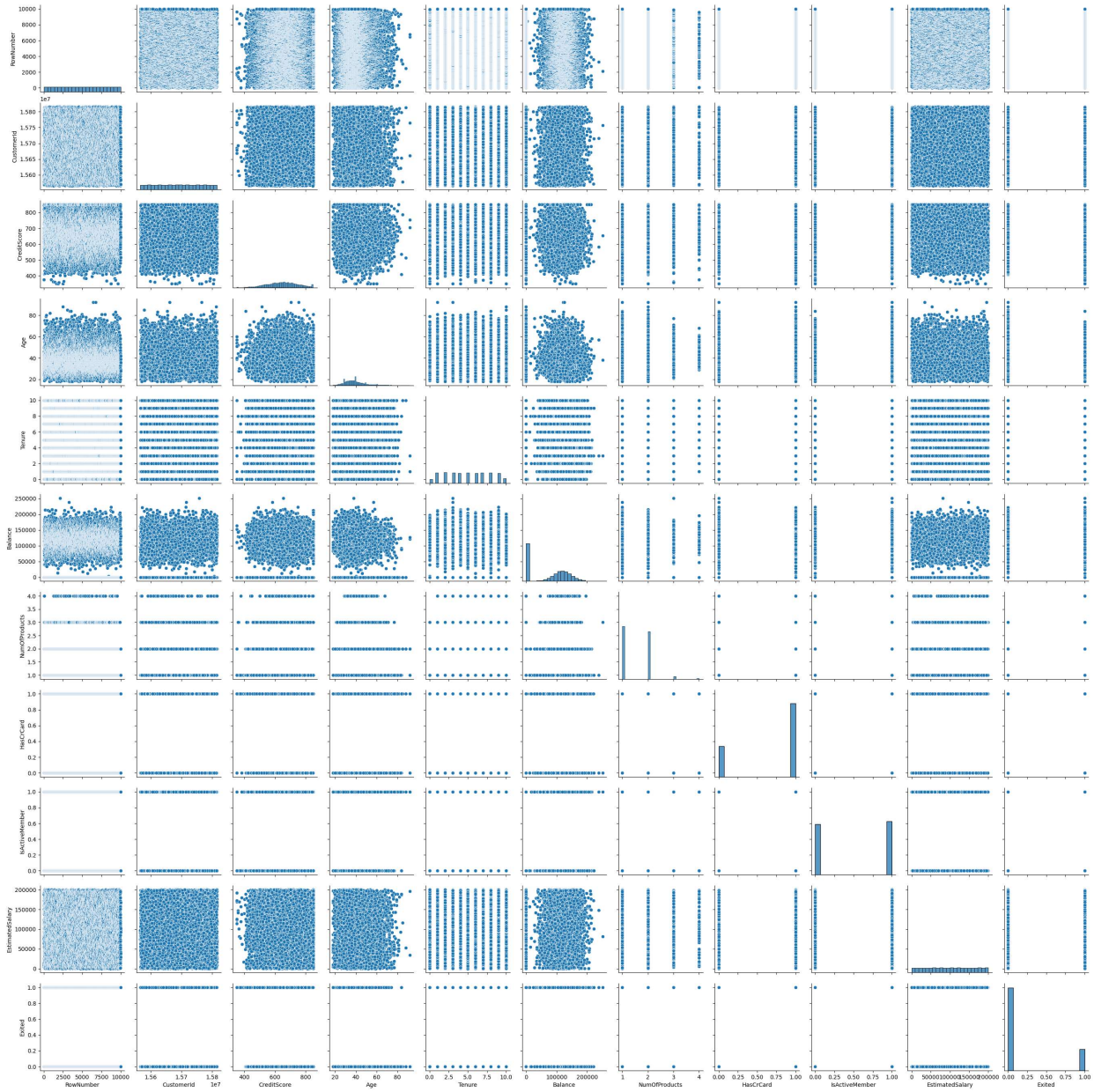
```
C:\Users\ssdha\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
Pass the following variables as keyword args: x, y. From version 0.12, the only valid
positional argument will be `data`, and passing other arguments without an explicit k
eyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[4]:
```
(0.0, 15000.0)
```



In [8]:
```
import seaborn as sns
df=pd.read_csv("C:\\Users\\ssdha\\Downloads\\Churn_Modelling.csv")
sns.pairplot(df)
```

Out[8]:
```
<seaborn.axisgrid.PairGrid at 0x13cadc81130>
```



In [9]:
```
df=pd.read_csv("C:\\Users\\ssdha\\Downloads\\Churn_Modelling.csv")
df.describe(include='all')
```

Out[9]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	
count	10000.00000	1.000000e+04	10000	10000.000000	10000	10000	10000.000000	1000
unique	NaN	NaN	2932	NaN	3	2	NaN	
top	NaN	NaN	Smith	NaN	France	Male	NaN	
freq	NaN	NaN	32	NaN	5014	5457	NaN	
mean	5000.50000	1.569094e+07	NaN	650.528800	NaN	NaN	38.921800	
std	2886.89568	7.193619e+04	NaN	96.653299	NaN	NaN	10.487806	
min	1.00000	1.556570e+07	NaN	350.000000	NaN	NaN	18.000000	
25%	2500.75000	1.562853e+07	NaN	584.000000	NaN	NaN	32.000000	
50%	5000.50000	1.569074e+07	NaN	652.000000	NaN	NaN	37.000000	
75%	7500.25000	1.575323e+07	NaN	718.000000	NaN	NaN	44.000000	
max	10000.00000	1.581569e+07	NaN	850.000000	NaN	NaN	92.000000	1

In [10]:
```
df.count()
```

Out[10]:
```
RowNumber          10000
CustomerId         10000
Surname            10000
CreditScore        10000
Geography          10000
Gender             10000
Age                10000
Tenure             10000
Balance            10000
NumOfProducts      10000
HasCrCard          10000
IsActiveMember     10000
EstimatedSalary    10000
Exited             10000
dtype: int64
```

In [11]:
```
df['Geography'].value_counts()
```

Out[11]:
```
France     5014
Germany    2509
Spain      2477
Name: Geography, dtype: int64
```

In [13]:
```
from ast import increment_lineno
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(color_codes=True)
df=pd.read_csv("C:\\Users\\ssdha\\Downloads\\Churn_Modelling.csv")
df.head()
```

Out[13]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	N
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	
2	3	15619304	Onio	502	France	Female	42	8	159660.80	
3	4	15701354	Boni	699	France	Female	39	1	0.00	
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	

In [14]:
```
df.isnull().sum()
```

Out[14]:
```
RowNumber          0
CustomerId         0
Surname            0
CreditScore        0
Geography          0
Gender             0
Age                0
Tenure             0
Balance            0
NumOfProducts      0
HasCrCard          0
IsActiveMember     0
EstimatedSalary    0
Exited             0
dtype: int64
```

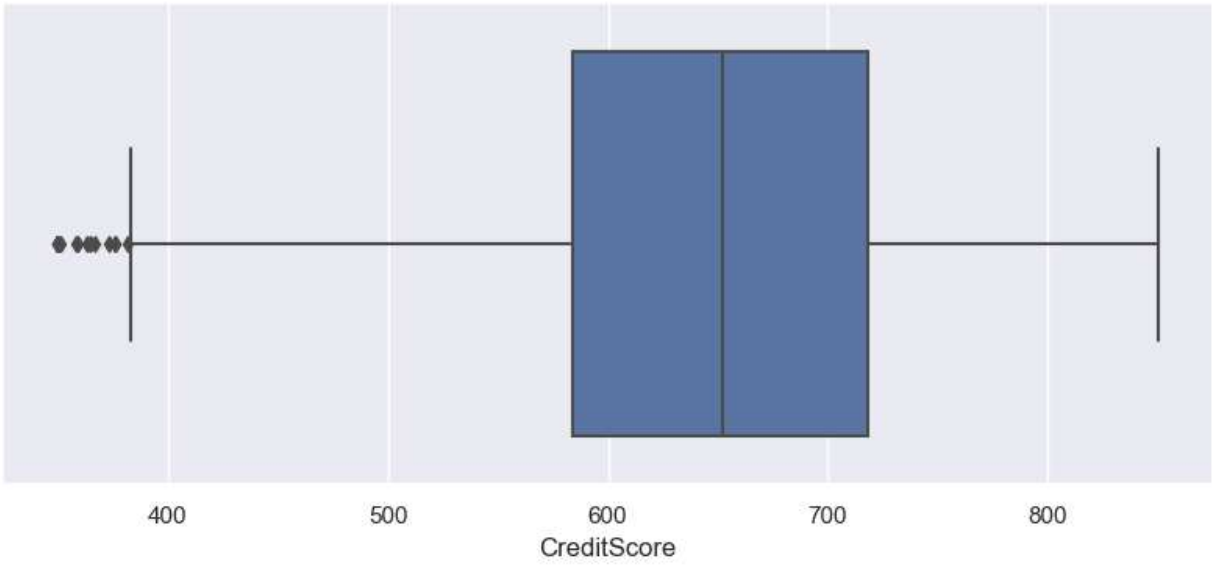
In [15]:
```
import pandas as pd
import matplotlib
from matplotlib import pyplot as pyplot
%matplotlib inline
matplotlib.rcParams['figure.figsize']=(10,4)
df=pd.read_csv("C:\\Users\\ssdha\\Downloads\\Churn_Modelling.csv")
df.sample(5)
```

Out[15]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance
5082	5083	15793307	Calabresi	724	Spain	Female	41	4	142880.28
9340	9341	15605804	Watson	737	France	Male	45	10	0.00
7013	7014	15599440	McGregor	748	France	Female	34	8	0.00
803	804	15712825	Howells	511	Spain	Female	29	9	0.00
9859	9860	15615430	Adams	678	Germany	Male	55	4	129646.91

In [16]:
```
sns.boxplot(x='CreditScore', data=df)
```

Out[16]:
```
<AxesSubplot:xlabel='CreditScore'>
```



In [17]:
```
df=pd.read_csv("C:\\Users\\ssdha\\Downloads\\Churn_Modelling.csv")
df.columns
import pandas as pd
import numpy as np
headers=['RowNumber','CustomerID','Surname','CreditScore','Geography',
'Gender','Age','Tenure','Balance','NumofProducts','HasCard'
'IsActiveMember','EstimatedSalary','Exited']
import seaborn as sns
df.head()
```

Out[17]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	N
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	
2	3	15619304	Onio	502	France	Female	42	8	159660.80	
3	4	15701354	Boni	699	France	Female	39	1	0.00	
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	

In [18]:
```
#Splitting the Dataset into the Independent Feature Matrix:
X = df.iloc[:, :-1].values
print(X)
```

```
[[1 15634602 'Hargrave' ... 1 1 101348.88]
 [2 15647311 'Hill' ... 0 1 112542.58]
 [3 15619304 'Onio' ... 1 0 113931.57]
 ...
 [9998 15584532 'Liu' ... 0 1 42085.58]
 [9999 15682355 'Sabbatini' ... 1 0 92888.52]
 [10000 15628319 'Walker' ... 1 0 38190.78]]
```

In [19]:
```
#Extracting the Dataset to Get the Dependent Vector
Y = df.iloc[:, -1].values
print(Y)
```

```
[1 0 1 ... 1 1 0]
```

In [21]:
```
from sklearn.preprocessing import StandardScaler
```

In [25]:
```
from sklearn.model_selection import train_test_split
# split the dataset
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.05, random_state
)
```

In [26]:
```
X_train
```

Out[26]:
```
array([[800, 15567367, 'Tao', ..., 0, 1, 103315.74],
       [1070, 15628674, 'Iadanza', ..., 1, 0, 31904.31],
       [8411, 15609913, 'Clark', ..., 1, 0, 113436.08],
       ...,
       [3265, 15574372, 'Hoolan', ..., 1, 0, 181429.87],
       [9846, 15664035, 'Parsons', ..., 1, 1, 148750.16],
       [2733, 15592816, 'Udokamma', ..., 1, 0, 118855.26]], dtype=object)
```

In [27]:
```
Y_train
```

Out[27]:
```
array([0, 1, 0, ..., 0, 0, 1], dtype=int64)
```

In [28]:
```
X_test
```

Out[28]:
```
array([[9395, 15615753, 'Upchurch', ..., 1, 1, 192852.67],
       [899, 15654700, 'Fallaci', ..., 1, 0, 128702.1],
       [2399, 15633877, 'Morrison', ..., 1, 1, 75732.25],
       ...,
       [492, 15699005, 'Martin', ..., 1, 1, 9983.88],
       [2022, 15795519, 'Vasiliev', ..., 0, 0, 197322.13],
       [4300, 15711991, 'Chiawuotu', ..., 0, 0, 3183.15]], dtype=object)
```

In [29]:
```
Y_test
```

Out[29]:
```
array([0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1,
       0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0,
       1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0,
       1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0,
       0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0], dtype=int64)
```

In [ ]:
```
```