```
Name : Sagar Kapase
Roll No : BEA-07
```

Group B: Machine Learning

Assignment B3

Given a bank customer, build a neural network-based classifier that can determine whether they will leave or not in the next 6 months.

Dataset Description: The case study is from an open-source dataset from Kaggle. The dataset contains 10,000 sample points with 14 distinct features such as CustomerId, CreditScore, Geography, Gender, Age, Tenure, Balance, etc.

Link to the Kaggle project: https://www.kaggle.com/barelydedicated/bank-customer-churn-modeling (https://www.kaggle.com/barelydedicated/bank-customer-churn-modeling)

Perform following steps:

1. Read the dataset.
2. Distinguish the feature and target set and divide the data set into training and test sets.
3. Normalize the train and test data.
4. Initialize and build the model. Identify the points of improvement and implement the same.
5. Print the accuracy score and confusion matrix.

In [ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

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

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

Out[316]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance
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 [ ]:
```
df.shape
```

Out[317]:
```
(10000, 14)
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   RowNumber        10000 non-null  int64  
 1   CustomerId       10000 non-null  int64  
 2   Surname          10000 non-null  object 
 3   CreditScore      10000 non-null  int64  
 4   Geography        10000 non-null  object 
 5   Gender           10000 non-null  object 
 6   Age              10000 non-null  int64  
 7   Tenure           10000 non-null  int64  
 8   Balance          10000 non-null  float64
 9   NumOfProducts    10000 non-null  int64  
 10  HasCrCard        10000 non-null  int64  
 11  IsActiveMember   10000 non-null  int64  
 12  EstimatedSalary  10000 non-null  float64
 13  Exited           10000 non-null  int64  
dtypes: float64(2), int64(9), object(3)
memory usage: 1.1+ MB
```

In [ ]:
```
df['RowNumber'].value_counts()
```

Out[319]:
```
1        1
6671     1
6664     1
6665     1
6666     1
        ..
3334     1
3335     1
3336     1
3337     1
10000    1
Name: RowNumber, Length: 10000, dtype: int64
```

In [ ]:
```
df['RowNumber'].nunique()
```

Out[320]:
```
10000
```

In [ ]:
```
df['CustomerId'].nunique()
```

Out[321]:
```
10000
```

In [ ]:
```
df.drop(['RowNumber','CustomerId','Surname'],axis=1,inplace=True)
```

In [ ]:
```
df.shape
```

Out[323]:
```
(10000, 11)
```

In [ ]:
```
df.duplicated().sum()
```

Out[324]:
```
0
```

In [ ]:
```
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  object 
 2   Gender           10000 non-null  object 
 3   Age              10000 non-null  int64  
 4   Tenure           10000 non-null  int64  
 5   Balance          10000 non-null  float64
 6   NumOfProducts    10000 non-null  int64  
 7   HasCrCard        10000 non-null  int64  
 8   IsActiveMember   10000 non-null  int64  
 9   EstimatedSalary  10000 non-null  float64
 10  Exited           10000 non-null  int64  
dtypes: float64(2), int64(7), object(2)
memory usage: 859.5+ KB
```

In [ ]:
```
df['Gender'].value_counts()
```

Out[326]:
```
Male      5457
Female    4543
Name: Gender, dtype: int64
```

In [ ]:
```
grp = df.groupby('Gender')['Exited'].value_counts()
grp
```

Out[327]:
```
Gender  Exited
Female  0         3404
        1         1139
Male    0         4559
        1          898
Name: Exited, dtype: int64
```

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

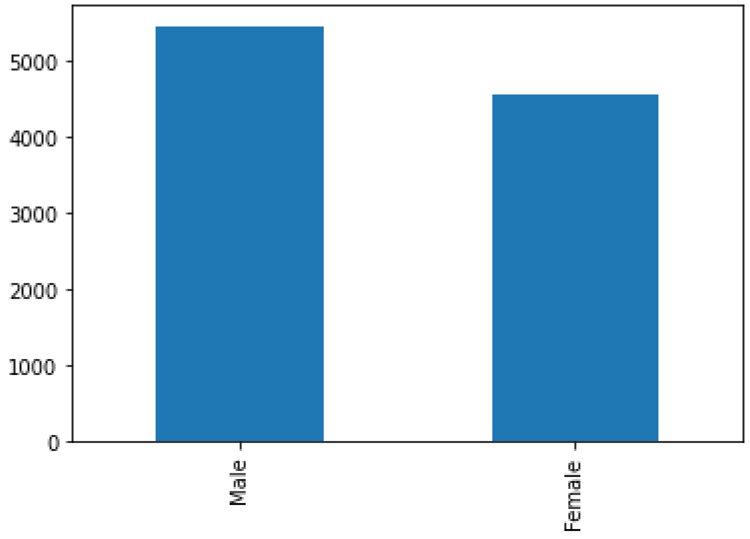
Out[328]:
```
France     5014
Germany    2509
Spain      2477
Name: Geography, dtype: int64
```

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

Out[329]:
```
Geography  Exited
France     0         4204
           1          810
Germany    0         1695
           1          814
Spain      0         2064
           1          413
Name: Exited, dtype: int64
```

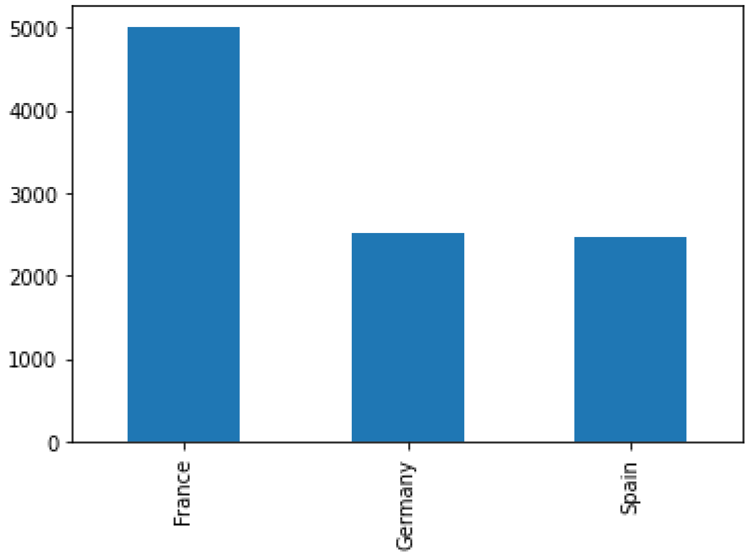
In [ ]:
```
df['Gender'].value_counts().plot(kind='bar')
```

Out[330]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd674b1d250>
```



In [ ]:
```
df['Geography'].value_counts().plot(kind='bar')
```

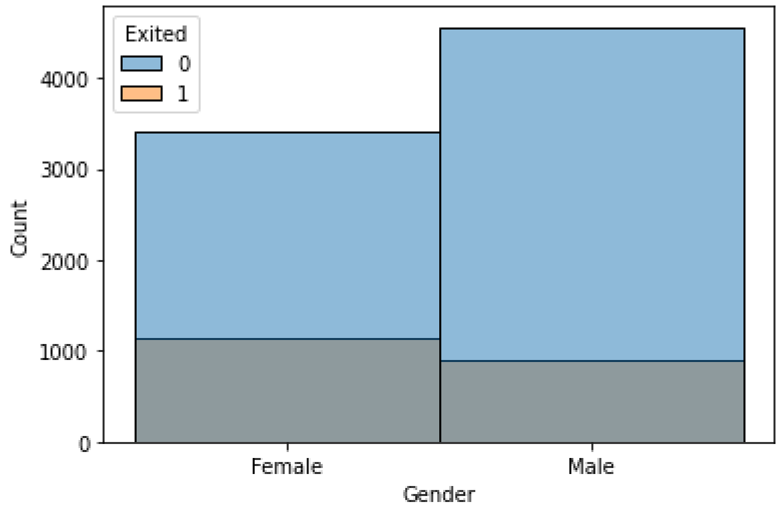
Out[331]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd674a33210>
```



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

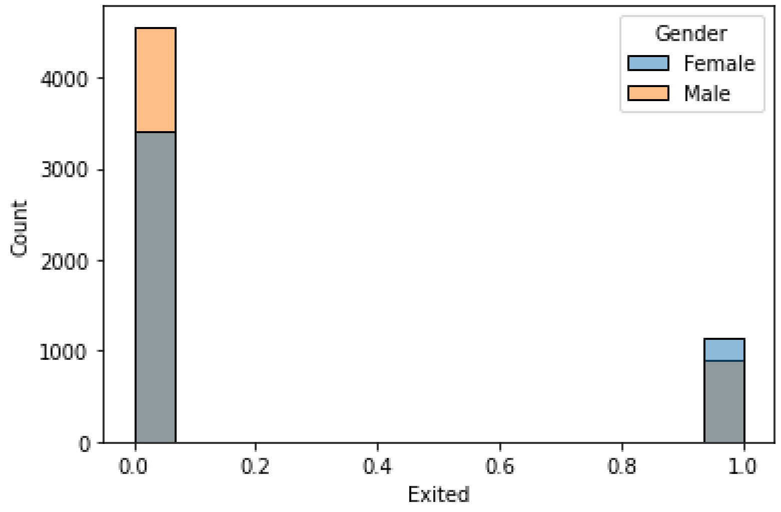
```
In [ ]: sns.histplot(df,x='Gender',hue='Exited')
```

```
Out[333]: <matplotlib.axes._subplots.AxesSubplot at 0x7fd6749f7350>
```



```
In [ ]: sns.histplot(df,x='Exited',hue='Gender')
```

```
Out[334]: <matplotlib.axes._subplots.AxesSubplot at 0x7fd6748fd7d0>
```



```
In [ ]: df.Exited.value_counts()
```

```
Out[335]: 0    7963
          1    2037
          Name: Exited, dtype: int64
```

```
In [ ]: df.to_csv('analytical_base_table.csv', index=None)
```

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

In [ ]:
```
x = df.drop(['Exited'],axis=1)
x.shape
```

Out[338]:
```
(10000, 10)
```

In [ ]:
```
x.head()
```

Out[339]:

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActive
0	619	France	Female	42	2	0.00	1	1	
1	608	Spain	Female	41	1	83807.86	1	0	
2	502	France	Female	42	8	159660.80	3	1	
3	699	France	Female	39	1	0.00	2	0	
4	850	Spain	Female	43	2	125510.82	1	1	

In [ ]:
```
y = df['Exited']
y
```

Out[340]:
```
0       1
1       0
2       1
3       0
4       0
       ..
9995    0
9996    0
9997    1
9998    1
9999    0
Name: Exited, Length: 10000, dtype: int64
```

In [ ]:
```
from sklearn.model_selection import train_test_split
```

In [ ]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.25,random_state=
```

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

Out[343]:
```
(10000, 10)
```

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

Out[344]:
```
(7500, 10)
```

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

Out[345]:
```
(7500,)
```

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

Out[346]:
```
(2500, 10)
```

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

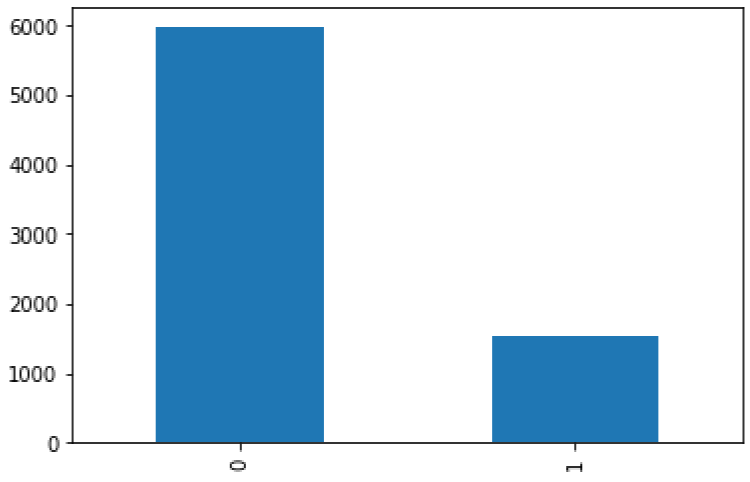
Out[347]:
```
(2500,)
```

In [ ]:
```
y_train.value_counts()
```

Out[348]:
```
0    5972
1    1528
Name: Exited, dtype: int64
```

In [ ]:
```
y_train.value_counts().plot(kind='bar')
```

Out[349]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd6767cf890>
```

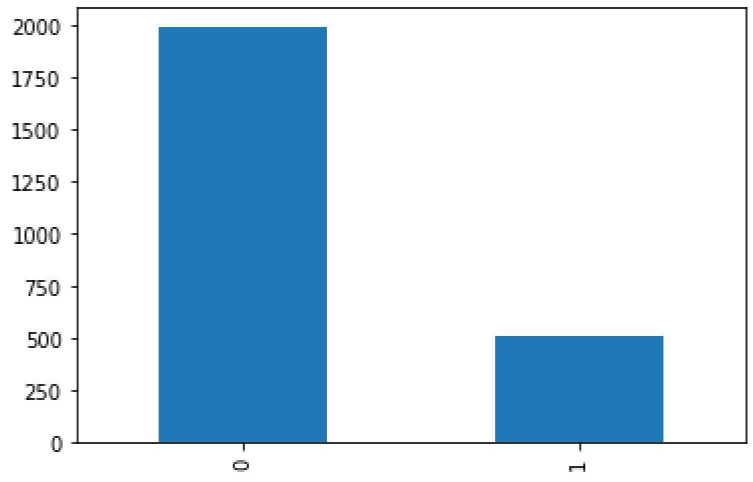


In [ ]:
```
y_test.value_counts()
```

Out[350]:
```
0    1991
1     509
Name: Exited, dtype: int64
```

In [ ]:
```
y_test.value_counts().plot(kind='bar')
```

Out[351]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd674b07d90>
```



In [ ]:
```
x_train.head()
```

Out[352]:

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsAc
1792	686	Spain	Male	41	7	102749.72	1	0	
8733	749	Spain	Male	42	9	222267.63	1	0	
4679	777	Spain	Female	35	3	0.00	2	1	
744	650	France	Male	60	8	0.00	2	1	
7985	696	Germany	Female	27	2	96129.32	2	1	

In [ ]:
```
x_train.reset_index(drop=True,inplace=True)
x_train
```

Out[353]:

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsAc
0	686	Spain	Male	41	7	102749.72	1	0	
1	749	Spain	Male	42	9	222267.63	1	0	
2	777	Spain	Female	35	3	0.00	2	1	
3	650	France	Male	60	8	0.00	2	1	
4	696	Germany	Female	27	2	96129.32	2	1	
...	...	...	...	...	...	...	...	...	
7495	656	France	Female	46	5	113402.14	2	1	
7496	526	Spain	Male	49	2	0.00	1	1	
7497	780	Germany	Male	51	4	126725.25	1	1	
7498	850	Spain	Male	48	2	0.00	1	1	
7499	705	Germany	Female	46	4	115518.07	1	0	

7500 rows × 10 columns

In [ ]:
```
x_test.reset_index(drop=True,inplace=True)
x_test
```

Out[354]:

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsAc
0	706	France	Female	29	5	112564.62	1	1	
1	554	Germany	Female	31	6	135470.90	1	1	
2	704	Germany	Female	24	7	113034.22	1	1	
3	757	France	Female	71	0	88084.13	2	1	
4	651	France	Male	36	7	0.00	2	1	
...	...	...	...	...	...	...	...	...	
2495	577	Spain	Male	43	8	79757.21	1	1	
2496	608	Germany	Male	26	1	106648.98	1	0	
2497	697	France	Female	25	1	0.00	2	0	
2498	634	France	Male	26	8	0.00	1	1	
2499	437	France	Female	39	0	102721.49	1	0	

2500 rows × 10 columns

In [ ]:
```
from sklearn.preprocessing import OneHotEncoder
```

In [ ]:
```
ohe = OneHotEncoder(drop='first',sparse=False,handle_unknown='ignore')
```

In [ ]:
```
ohe.fit(x_train[['Gender','Geography']])
```

Out[357]:
```
OneHotEncoder(drop='first', handle_unknown='ignore', sparse=False)
```

In [ ]:
```
ohe.get_feature_names(['Gender','Geography'])
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureW
arning: Function get_feature_names is deprecated; get_feature_names is deprecat
ed in 1.0 and will be removed in 1.2. Please use get_feature_names_out instead.
  warnings.warn(msg, category=FutureWarning)
```

Out[358]:
```
array(['Gender_Male', 'Geography_Germany', 'Geography_Spain'],
      dtype=object)
```

In [ ]:
```
x_train_encoded = pd.DataFrame(ohe.transform(x_train[['Gender','Geography']]),co
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureW
arning: Function get_feature_names is deprecated; get_feature_names is deprecat
ed in 1.0 and will be removed in 1.2. Please use get_feature_names_out instead.
  warnings.warn(msg, category=FutureWarning)
```

In [ ]:
```
x_train_encoded
```

Out[360]:

	Gender_Male	Geography_Germany	Geography_Spain
0	1.0	0.0	1.0
1	1.0	0.0	1.0
2	0.0	0.0	1.0
3	1.0	0.0	0.0
4	0.0	1.0	0.0
...	...	...	...
7495	0.0	0.0	0.0
7496	1.0	0.0	1.0
7497	1.0	1.0	0.0
7498	1.0	0.0	1.0
7499	0.0	1.0	0.0

7500 rows × 3 columns

In [ ]:
```
x_train_new = pd.concat([x_train,x_train_encoded],axis=1)
x_train_new
```

Out[361]:

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsAc
0	686	Spain	Male	41	7	102749.72	1	0	
1	749	Spain	Male	42	9	222267.63	1	0	
2	777	Spain	Female	35	3	0.00	2	1	
3	650	France	Male	60	8	0.00	2	1	
4	696	Germany	Female	27	2	96129.32	2	1	
...	...	...	...	...	...	...	...	...	
7495	656	France	Female	46	5	113402.14	2	1	
7496	526	Spain	Male	49	2	0.00	1	1	
7497	780	Germany	Male	51	4	126725.25	1	1	
7498	850	Spain	Male	48	2	0.00	1	1	
7499	705	Germany	Female	46	4	115518.07	1	0	

7500 rows × 13 columns

In [ ]:
```
x_train_new.drop(['Gender','Geography'],axis=1,inplace=True)
x_train_new
```

Out[362]:

	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	Estimatec
0	686	41	7	102749.72	1	0	1	19
1	749	42	9	222267.63	1	0	0	10
2	777	35	3	0.00	2	1	1	1
3	650	60	8	0.00	2	1	1	10
4	696	27	2	96129.32	2	1	1	
...	...	...	...	...	...	...	...	
7495	656	46	5	113402.14	2	1	1	13
7496	526	49	2	0.00	1	1	0	11
7497	780	51	4	126725.25	1	1	0	19
7498	850	48	2	0.00	1	1	0	16
7499	705	46	4	115518.07	1	0	0	7

7500 rows × 11 columns

```
In [ ]: x_test_encoded = pd.DataFrame(ohe.transform(x_test[['Gender','Geography']]),colum
        x_test_encoded
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureW
arning: Function get_feature_names is deprecated; get_feature_names is deprecat
ed in 1.0 and will be removed in 1.2. Please use get_feature_names_out instead.
  warnings.warn(msg, category=FutureWarning)
```

Out[363]:

	Gender_Male	Geography_Germany	Geography_Spain
0	0.0	0.0	0.0
1	0.0	1.0	0.0
2	0.0	1.0	0.0
3	0.0	0.0	0.0
4	1.0	0.0	0.0
...	...	...	...
2495	1.0	0.0	1.0
2496	1.0	1.0	0.0
2497	0.0	0.0	0.0
2498	1.0	0.0	0.0
2499	0.0	0.0	0.0

2500 rows × 3 columns

In [ ]:
```
x_test_new = pd.concat([x_test,x_test_encoded],axis=1)
x_test_new
```

Out[364]:

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsAc
0	706	France	Female	29	5	112564.62	1	1	
1	554	Germany	Female	31	6	135470.90	1	1	
2	704	Germany	Female	24	7	113034.22	1	1	
3	757	France	Female	71	0	88084.13	2	1	
4	651	France	Male	36	7	0.00	2	1	
...	...	...	...	...	...	...	...	...	
2495	577	Spain	Male	43	8	79757.21	1	1	
2496	608	Germany	Male	26	1	106648.98	1	0	
2497	697	France	Female	25	1	0.00	2	0	
2498	634	France	Male	26	8	0.00	1	1	
2499	437	France	Female	39	0	102721.49	1	0	

2500 rows × 13 columns

In [ ]:
```
x_test_new.drop(['Gender','Geography'],axis=1,inplace=True)
x_test_new
```

Out[365]:

	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	Estimate
0	706	29	5	112564.62	1	1	0	4
1	554	31	6	135470.90	1	1	0	10
2	704	24	7	113034.22	1	1	0	16
3	757	71	0	88084.13	2	1	1	15
4	651	36	7	0.00	2	1	0	1
...	...	...	...	...	...	...	...	
2495	577	43	8	79757.21	1	1	0	13
2496	608	26	1	106648.98	1	0	1	
2497	697	25	1	0.00	2	0	0	8
2498	634	26	8	0.00	1	1	0	2
2499	437	39	0	102721.49	1	0	0	2

2500 rows × 11 columns

In [ ]:
```
from sklearn.linear_model import LogisticRegression
```

In [ ]:
```
lr =LogisticRegression()
```

In [ ]:
```
lr.fit(x_train_new,y_train)
```

Out[368]:
```
LogisticRegression()
```

In [ ]:
```
y_pred=lr.predict(x_test_new)
y_pred
```

Out[369]:
```
array([0, 0, 0, ..., 0, 0, 0])
```

In [ ]:
```
from sklearn.metrics import confusion_matrix,classification_report,precision_scor
```

In [ ]:
```
confusion_matrix(y_test,y_pred)
```

Out[371]:
```
array([[1957,   34],
       [ 483,   26]])
```

In [ ]:
```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.80      0.98      0.88      1991
           1       0.43      0.05      0.09       509

    accuracy                           0.79      2500
   macro avg       0.62      0.52      0.49      2500
weighted avg       0.73      0.79      0.72      2500
```

In [ ]:
```
pre = precision_score(y_test,y_pred)
```

In [ ]:
```
re = recall_score(y_test,y_pred)
```

In [ ]:
```
acc = accuracy_score(y_test,y_pred)
```

In [ ]:
```
fbeta = fbeta_score(y_test,y_pred,beta=2)
```

In [ ]:
```
result = pd.DataFrame(columns=['Accuracy','Precision','Reall','Fbeta Score'])
result
```

Out[377]:

Accuracy	Precision	Reall	Fbeta Score

In [ ]:
```
result.loc['LR'] = [acc,pre,re,fbeta]
result
```

Out[378]:

	Accuracy	Precision	Reall	Fbeta Score
LR	0.7932	0.433333	0.051081	0.062023

In [ ]:
```
from sklearn.preprocessing import MinMaxScaler
```

In [ ]:
```
scaler = MinMaxScaler()
scaler.fit(x_train_new)
x_train_new_scaled=scaler.transform(x_train_new)
x_test_new_scaled=scaler.transform(x_test_new)
```

In [ ]:
```
lr.fit(x_train_new_scaled,y_train)
y_pred=lr.predict(x_test_new_scaled)
y_pred
```

Out[381]:
```
array([0, 0, 0, ..., 0, 0, 0])
```

In [ ]:
```
confusion_matrix(y_test,y_pred)
```

Out[382]:
```
array([[1918,   73],
       [ 415,   94]])
```

In [ ]:
```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.82      0.96      0.89      1991
           1       0.56      0.18      0.28       509

    accuracy                           0.80      2500
   macro avg       0.69      0.57      0.58      2500
weighted avg       0.77      0.80      0.76      2500
```

In [ ]:
```
acc = accuracy_score(y_test,y_pred)
```

In [ ]:
```
re = recall_score(y_test,y_pred)
```

In [ ]:
```
pre = precision_score(y_test,y_pred)
```

In [ ]:
```
fbeta = fbeta_score(y_test,y_pred,beta=2)
```

In [ ]:
```
result.loc['LR_Scaling'] = [acc,pre,re,fbeta]
result
```

Out[388]:

	Accuracy	Precision	Reall	Fbeta Score
LR	0.7932	0.433333	0.051081	0.062023
LR_Scaling	0.8048	0.562874	0.184676	0.213345

In [ ]:
```
# Keras
# from tensorflow.keras.wrappers.scikit_learn import KerasClassifier
# from tensorflow.keras.layers import Dense, Input, Dropout
# from tensorflow.keras.models import Sequential
```

In [ ]:
```
pip install keras_tuner
```

```
Looking in indexes: https://pypi.org/simple, (https://pypi.org/simple,) http
s://us-python.pkg.dev/colab-wheels/public/simple/ (https://us-python.pkg.dev/
colab-wheels/public/simple/)
Requirement already satisfied: keras_tuner in /usr/local/lib/python3.7/dist-p
ackages (1.1.3)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-package
s (from keras_tuner) (1.21.6)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-pack
ages (from keras_tuner) (2.23.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-pac
kages (from keras_tuner) (21.3)
Requirement already satisfied: tensorboard in /usr/local/lib/python3.7/dist-p
ackages (from keras_tuner) (2.8.0)
Requirement already satisfied: kt-legacy in /usr/local/lib/python3.7/dist-pac
kages (from keras_tuner) (1.0.4)
Requirement already satisfied: ipython in /usr/local/lib/python3.7/dist-packa
ges (from keras_tuner) (7.9.0)
Requirement already satisfied: pexpect in /usr/local/lib/python3.7/dist-packa
ges (from ipython->keras_tuner) (4.8.0)
Requirement already satisfied: pygments in /usr/local/lib/python3.7/dist-pack
ages (from ipython->keras_tuner) (2.6.1)
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.7/d
ist-packages (from ipython->keras_tuner) (57.4.0)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.7/dist-p
ackages (from ipython->keras_tuner) (0.7.5)
Requirement already satisfied: backcall in /usr/local/lib/python3.7/dist-pack
ages (from ipython->keras_tuner) (0.2.0)
Requirement already satisfied: jedi>=0.10 in /usr/local/lib/python3.7/dist-pa
ckages (from ipython->keras_tuner) (0.18.1)
Requirement already satisfied: prompt-toolkit<2.1.0,>=2.0.0 in /usr/local/li
b/python3.7/dist-packages (from ipython->keras_tuner) (2.0.10)
Requirement already satisfied: decorator in /usr/local/lib/python3.7/dist-pac
kages (from ipython->keras_tuner) (4.4.2)
Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.7/dis
t-packages (from ipython->keras_tuner) (5.1.1)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in /usr/local/lib/python3.
7/dist-packages (from jedi>=0.10->ipython->keras_tuner) (0.8.3)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.7/dist-packa
ges (from prompt-toolkit<2.1.0,>=2.0.0->ipython->keras_tuner) (0.2.5)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.7/dist-pa
ckages (from prompt-toolkit<2.1.0,>=2.0.0->ipython->keras_tuner) (1.15.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/pyt
hon3.7/dist-packages (from packaging->keras_tuner) (3.0.9)
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.7/di
st-packages (from pexpect->ipython->keras_tuner) (0.7.0)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-
packages (from requests->keras_tuner) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.
7/dist-packages (from requests->keras_tuner) (2022.6.15)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /us
r/local/lib/python3.7/dist-packages (from requests->keras_tuner) (1.24.3)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/
dist-packages (from requests->keras_tuner) (3.0.4)
Requirement already satisfied: absl-py>=0.4 in /usr/local/lib/python3.7/dist-
```

```
packages (from tensorboard->keras_tuner) (1.2.0)
Requirement already satisfied: werkzeug>=0.11.15 in /usr/local/lib/python3.7/
dist-packages (from tensorboard->keras_tuner) (1.0.1)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.7/di
st-packages (from tensorboard->keras_tuner) (3.4.1)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /usr/
local/lib/python3.7/dist-packages (from tensorboard->keras_tuner) (0.6.1)
Requirement already satisfied: grpcio>=1.24.3 in /usr/local/lib/python3.7/dis
t-packages (from tensorboard->keras_tuner) (1.48.1)
Requirement already satisfied: wheel>=0.26 in /usr/local/lib/python3.7/dist-p
ackages (from tensorboard->keras_tuner) (0.37.1)
Requirement already satisfied: protobuf>=3.6.0 in /usr/local/lib/python3.7/di
st-packages (from tensorboard->keras_tuner) (3.17.3)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/loca
l/lib/python3.7/dist-packages (from tensorboard->keras_tuner) (0.4.6)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local/li
b/python3.7/dist-packages (from tensorboard->keras_tuner) (1.8.1)
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python
3.7/dist-packages (from tensorboard->keras_tuner) (1.35.0)
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.7/dist
-packages (from google-auth<3,>=1.6.3->tensorboard->keras_tuner) (4.9)
Requirement already satisfied: cachetools<5.0,>=2.0.0 in /usr/local/lib/pytho
n3.7/dist-packages (from google-auth<3,>=1.6.3->tensorboard->keras_tuner) (4.
2.4)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python
3.7/dist-packages (from google-auth<3,>=1.6.3->tensorboard->keras_tuner) (0.
2.8)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/pyt
hon3.7/dist-packages (from google-auth-oauthlib<0.5,>=0.4.1->tensorboard->ker
as_tuner) (1.3.1)
Requirement already satisfied: importlib-metadata>=4.4 in /usr/local/lib/pyth
on3.7/dist-packages (from markdown>=2.6.8->tensorboard->keras_tuner) (4.12.0)
Requirement already satisfied: typing-extensions>=3.6.4 in /usr/local/lib/pyt
hon3.7/dist-packages (from importlib-metadata>=4.4->markdown>=2.6.8->tensorbo
ard->keras_tuner) (4.1.1)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-pac
kages (from importlib-metadata>=4.4->markdown>=2.6.8->tensorboard->keras_tune
r) (3.8.1)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/python
3.7/dist-packages (from pyasn1-modules>=0.2.1->google-auth<3,>=1.6.3->tensorb
oard->keras_tuner) (0.4.8)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/di
st-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1-
>tensorboard->keras_tuner) (3.2.0)
```

In [ ]:
```
import tensorflow as tf
from keras_tuner.tuners import RandomSearch
```

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

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

Out[393]:

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActive
0	619	France	Female	42	2	0.00	1	1	
1	608	Spain	Female	41	1	83807.86	1	0	
2	502	France	Female	42	8	159660.80	3	1	
3	699	France	Female	39	1	0.00	2	0	
4	850	Spain	Female	43	2	125510.82	1	1	

```
In [ ]: x=df.drop(['Exited'],axis=1)
        x.shape
```

```
Out[394]: (10000, 10)
```

```
In [ ]: y=df['Exited']
        y.shape
```

```
Out[395]: (10000,)
```

```
In [ ]: x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=4
```

```
In [ ]: x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

```
Out[397]: ((8000, 10), (2000, 10), (8000,), (2000,))
```

```
In [ ]: x_train.head()
```

Out[398]:

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsAc
2151	753	France	Male	57	7	0.00	1	1	
8392	739	Germany	Male	32	3	102128.27	1	1	
5006	755	Germany	Female	37	0	113865.23	2	1	
4117	561	France	Male	37	5	0.00	2	1	
7182	692	Germany	Male	49	6	110540.43	2	0	

```
In [ ]: x_train.reset_index(drop=True,inplace=True)
```

In [ ]:
```
x_train.head()
```

Out[400]:

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActive
0	753	France	Male	57	7	0.00	1	1	
1	739	Germany	Male	32	3	102128.27	1	1	
2	755	Germany	Female	37	0	113865.23	2	1	
3	561	France	Male	37	5	0.00	2	1	
4	692	Germany	Male	49	6	110540.43	2	0	

In [ ]:
```
ohe = OneHotEncoder(drop='first',sparse=False,handle_unknown='ignore')
```

In [ ]:
```
ohe.fit(x_train[['Gender','Geography']])
```

Out[402]:
```
OneHotEncoder(drop='first', handle_unknown='ignore', sparse=False)
```

In [ ]:
```
x_train_encoded = ohe.transform(x_train[['Gender','Geography']])
```

In [ ]:
```
x_train_encoded
```

Out[404]:
```
array([[1., 0., 0.],
       [1., 1., 0.],
       [0., 1., 0.],
       ...,
       [0., 0., 0.],
       [1., 0., 1.],
       [1., 0., 1.]])
```

In [ ]:
```
x_train_new = pd.DataFrame(x_train_encoded,columns=ohe.get_feature_names(['Gender
x_train_new
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureW
arning: Function get_feature_names is deprecated; get_feature_names is deprecat
ed in 1.0 and will be removed in 1.2. Please use get_feature_names_out instead.
  warnings.warn(msg, category=FutureWarning)
```

Out[405]:

	Gender_Male	Geography_Germany	Geography_Spain
0	1.0	0.0	0.0
1	1.0	1.0	0.0
2	0.0	1.0	0.0
3	1.0	0.0	0.0
4	1.0	1.0	0.0
...	...	...	...
7995	0.0	0.0	1.0
7996	1.0	0.0	1.0
7997	0.0	0.0	0.0
7998	1.0	0.0	1.0
7999	1.0	0.0	1.0

8000 rows × 3 columns

In [ ]:
```
x_train1 = pd.concat([x_train,x_train_new],axis=1)
x_train1.head()
```

Out[406]:

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActive
0	753	France	Male	57	7	0.00	1	1	
1	739	Germany	Male	32	3	102128.27	1	1	
2	755	Germany	Female	37	0	113865.23	2	1	
3	561	France	Male	37	5	0.00	2	1	
4	692	Germany	Male	49	6	110540.43	2	0	

In [ ]:
```
x_train1.drop(['Geography','Gender'],axis=1,inplace=True)
x_train1.shape
```

Out[407]:
```
(8000, 11)
```

In [ ]:
```
x_test.head()
```

Out[408]:

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsAc
5702	585	France	Male	36	7	0.00	2	1	
3667	525	Germany	Male	33	4	131023.76	2	0	
1617	557	Spain	Female	40	4	0.00	2	0	
5673	639	Spain	Male	34	5	139393.19	2	0	
4272	640	Spain	Female	34	3	77826.80	1	1	

In [ ]:
```
x_test.reset_index(drop=True,inplace=True)
x_test.head()
```

Out[409]:

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActive
0	585	France	Male	36	7	0.00	2	1	
1	525	Germany	Male	33	4	131023.76	2	0	
2	557	Spain	Female	40	4	0.00	2	0	
3	639	Spain	Male	34	5	139393.19	2	0	
4	640	Spain	Female	34	3	77826.80	1	1	

In [ ]:
```
ohe.fit(x_test[['Gender','Geography']])
x_test_encoded = ohe.transform(x_test[['Gender','Geography']])
x_test_new = pd.DataFrame(x_test_encoded,columns=ohe.get_feature_names(['Gender',
x_test_new
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureW
arning: Function get_feature_names is deprecated; get_feature_names is deprecat
ed in 1.0 and will be removed in 1.2. Please use get_feature_names_out instead.
  warnings.warn(msg, category=FutureWarning)
```

Out[410]:

	Gender_Male	Geography_Germany	Geography_Spain
0	1.0	0.0	0.0
1	1.0	1.0	0.0
2	0.0	0.0	1.0
3	1.0	0.0	1.0
4	0.0	0.0	1.0
...	...	...	...
1995	1.0	0.0	0.0
1996	1.0	1.0	0.0
1997	0.0	1.0	0.0
1998	1.0	0.0	0.0
1999	1.0	0.0	0.0

2000 rows × 3 columns

In [ ]:
```
x_test1 = pd.concat([x_test,x_test_new],axis=1)
x_test1.head()
```

Out[411]:

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActive
0	585	France	Male	36	7	0.00	2	1	
1	525	Germany	Male	33	4	131023.76	2	0	
2	557	Spain	Female	40	4	0.00	2	0	
3	639	Spain	Male	34	5	139393.19	2	0	
4	640	Spain	Female	34	3	77826.80	1	1	

In [ ]:
```
x_test1.drop(['Geography','Gender'],axis=1,inplace=True)
x_test1.head()
```

Out[412]:

	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSa
0	585	36	7	0.00	2	1	0	9428
1	525	33	4	131023.76	2	0	0	5507
2	557	40	4	0.00	2	0	1	10543
3	639	34	5	139393.19	2	0	0	3395
4	640	34	3	77826.80	1	1	1	16854

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

In [ ]:
```
x_train1.columns
```

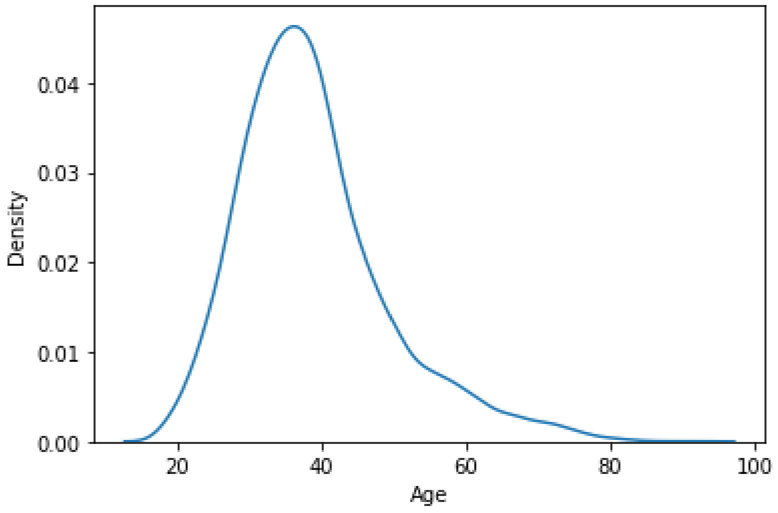
Out[414]:
```
Index(['CreditScore', 'Age', 'Tenure', 'Balance', 'NumOfProducts', 'HasCrCard',
       'IsActiveMember', 'EstimatedSalary', 'Gender_Male', 'Geography_Germany',
       'Geography_Spain'],
      dtype='object')
```

In [ ]:
```
for i in x_train1.columns:
  print(i)
```

```
CreditScore
Age
Tenure
Balance
NumOfProducts
HasCrCard
IsActiveMember
EstimatedSalary
Gender_Male
Geography_Germany
Geography_Spain
```

In [ ]:
```
sns.kdeplot(x_train1['Age'])
```

Out[416]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd674dd40d0>
```



In [ ]:
```
from sklearn.preprocessing import MinMaxScaler
```

In [ ]:
```
sc = MinMaxScaler()
sc.fit(x_train1)
x_train1_sc = sc.transform(x_train1)
x_test1_sc = sc.transform(x_test1)
```

In [ ]:
```
x_train1_sc.shape
```

Out[419]:
```
(8000, 11)
```

In [ ]:
```
type(x_train1_sc)
```

Out[420]:
```
numpy.ndarray
```

In [ ]:
```
import tensorflow.keras as tk
```

In [ ]:
```
# instantiate the model
model = tk.Sequential()
```

In [ ]:
```
# Adding the input layer
model.add(tk.layers.Input(shape=(11,)))
# Adding the first hidden layer
model.add(tk.layers.Dense(units=6,activation='relu',kernel_initializer='he_unifor
# Adding the second hidden layer
model.add(tk.layers.Dense(units=6,activation='relu',kernel_initializer='he_unifor
# Adding the output layer
model.add(tk.layers.Dense(units=1,activation='sigmoid',kernel_initializer='glorot
```

In [ ]:
```
# Compiling the model
model.compile(optimizer='Adam',loss='binary_crossentropy',metrics=['Precision','a
```

In [ ]:
```
model.summary()
```

```
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_12 (Dense)            (None, 6)                 72

 dense_13 (Dense)            (None, 6)                 42

 dense_14 (Dense)            (None, 1)                 7

=================================================================
Total params: 121
Trainable params: 121
Non-trainable params: 0
_________________________________________________________________
```

In [ ]:
```
x_train1_sc.shape,x_test1_sc.shape,y_train.shape,y_test.shape
```

Out[426]:
```
((8000, 11), (2000, 11), (8000,), (2000,))
```

In [ ]:
```
import time
```

In [ ]:
```
# Training the model
start = time.time()
history_object = model.fit(x=x_train1_sc,
                           y=y_train,
                           epochs=100,
                           batch_size=32,
                           validation_data=(x_test1_sc,y_test))
end=time.time()
print(end-start)
```

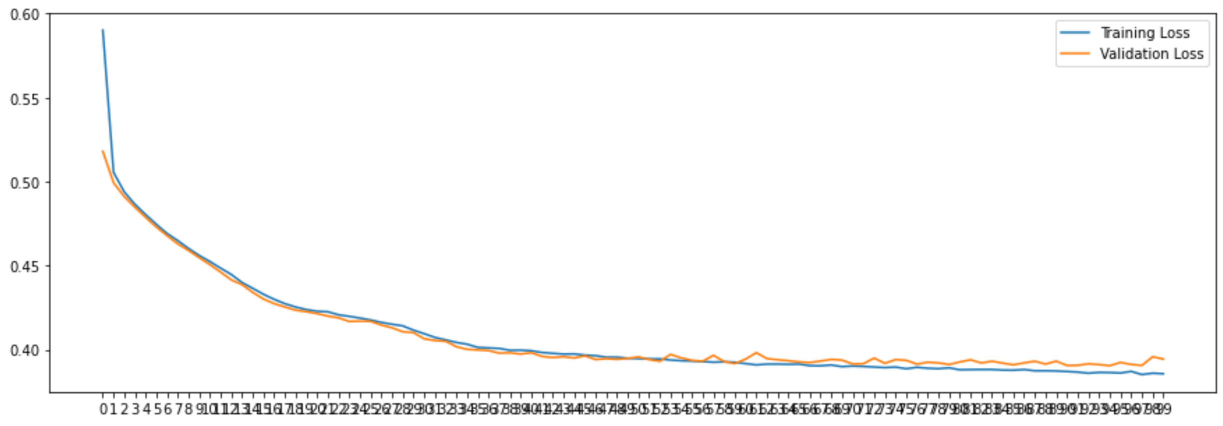
```
Epoch 1/100
250/250 [==============================] - 2s 4ms/step - loss: 0.5900 - preci
sion: 0.1512 - accuracy: 0.7588 - val_loss: 0.5179 - val_precision: 0.0000e+0
0 - val_accuracy: 0.7965
Epoch 2/100
250/250 [==============================] - 0s 2ms/step - loss: 0.5055 - preci
sion: 0.0000e+00 - accuracy: 0.7962 - val_loss: 0.4992 - val_precision: 0.000
0e+00 - val_accuracy: 0.7965
Epoch 3/100
250/250 [==============================] - 1s 2ms/step - loss: 0.4938 - preci
sion: 0.0000e+00 - accuracy: 0.7962 - val_loss: 0.4910 - val_precision: 0.000
0e+00 - val_accuracy: 0.7965
Epoch 4/100
250/250 [==============================] - 1s 2ms/step - loss: 0.4864 - preci
sion: 0.0000e+00 - accuracy: 0.7962 - val_loss: 0.4846 - val_precision: 0.000
0e+00 - val_accuracy: 0.7965
Epoch 5/100
250/250 [==============================] - 1s 2ms/step - loss: 0.4803 - preci
sion: 0.0000e+00 - accuracy: 0.7962 - val_loss: 0.4786 - val_precision: 0.000
```

In [ ]:
```
history_object.history.keys()
```

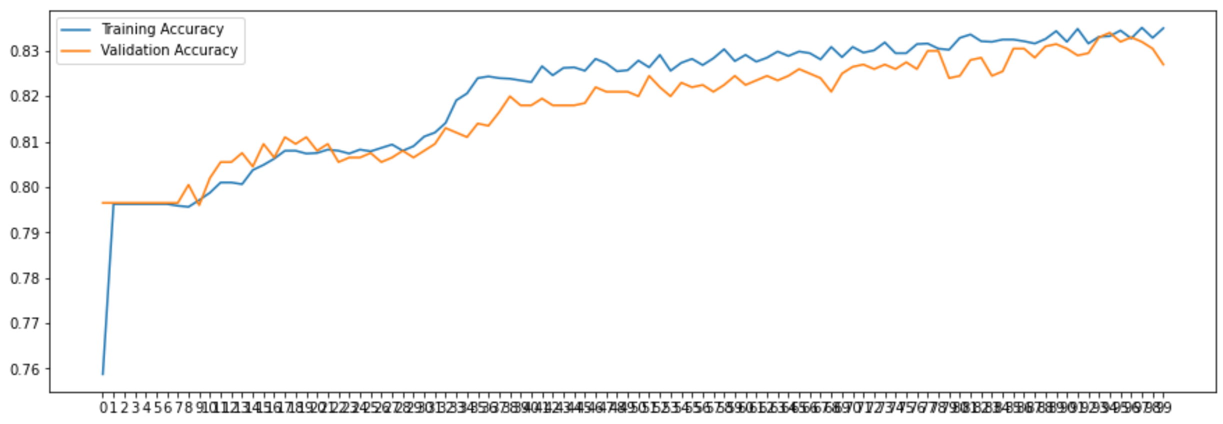
Out[429]:
```
dict_keys(['loss', 'precision', 'accuracy', 'val_loss', 'val_precision', 'val_a
ccuracy'])
```

In [ ]:
```
def lineplotter(history_object, keyword):
  epochs_ = history_object.epoch
  history_data = history_object.history
  tr_key = keyword
  val_key = f'val_{keyword}'

  tr_data = history_data.get(tr_key)
  val_data = history_data.get(val_key)

  plt.figure(figsize=(15,5))
  sns.lineplot(x = epochs_, y = tr_data)
  sns.lineplot(x = epochs_, y = val_data)
  plt.xticks(ticks = epochs_, labels = epochs_)
  plt.legend([f'Training {keyword.title()}',f'Validation {keyword.title()}'])
  plt.show()
```

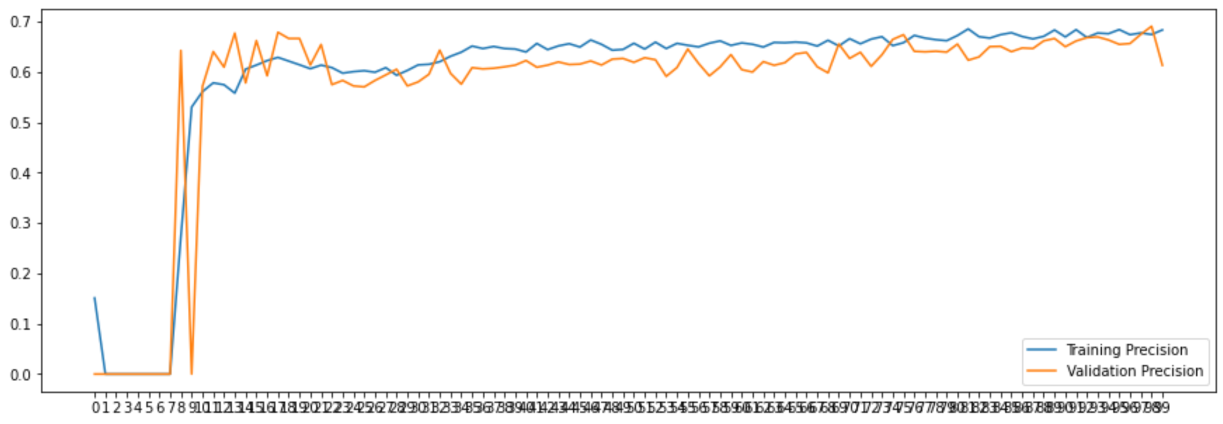
In [ ]:
```
lineplotter(history_object, 'loss')
```



In [ ]:
```
lineplotter(history_object, 'accuracy')
```



In [ ]:
```
lineplotter(history_object, 'precision')
```



In [ ]: