Grp_B3_Churn_Modelling

October 19, 2023

0.0.1 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

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.

[134]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

[135]:
```
import warnings
```

[136]:
```
warnings.filterwarnings("ignore")
```

[137]:
```
df = pd.read_csv('/Users/shreyas/Downloads/RAW/Mini-Pro/Churn_Modelling.csv')
```

[138]:
```
df.head(3)
```

[138]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1
2	3	15619304	Onio	502	France	Female	42					

```
2       8  159660.80              3          1               0

   EstimatedSalary  Exited
0        101348.88       1
1        112542.58       0
2        113931.57       1
```

[139]:
```
df.shape
```

[139]:
```
(10000, 14)
```

[140]:
```
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
```

[141]:
```
df['RowNumber'].value_counts()
```

[141]:
```
RowNumber
1        1
6671     1
6664     1
6665     1
6666     1
        ..
3334     1
3335     1
3336     1
3337     1
```

```
10000    1
Name: count, Length: 10000, dtype: int64
```

[142]:
```
df['RowNumber'].nunique()
```

[142]:
```
10000
```

[143]:
```
df['CustomerId'].nunique()
```

[143]:
```
10000
```

[144]:
```
df.drop(['RowNumber','CustomerId','Surname'],axis=1,inplace=True)
```

[145]:
```
df.shape
```

[145]:
```
(10000, 11)
```

[146]:
```
df.duplicated().sum()
```

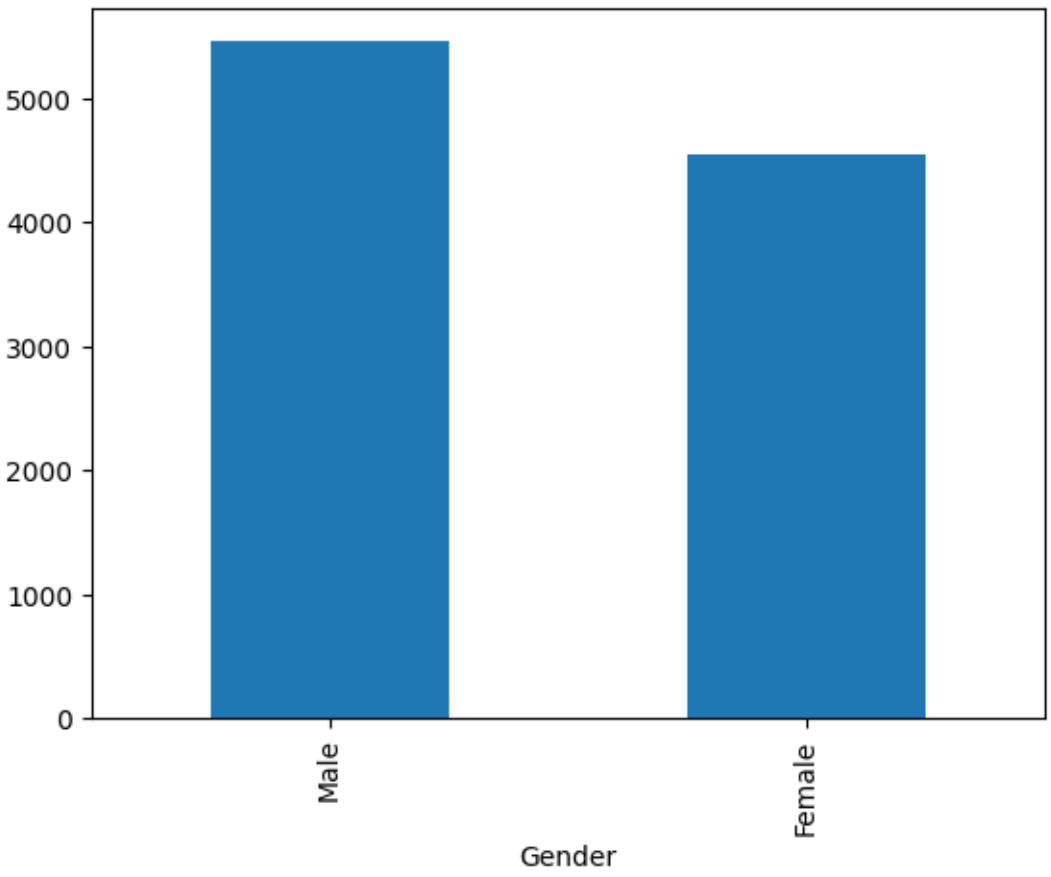
[146]:
```
0
```

[147]:
```
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
```

[148]:
```
df['Gender'].value_counts().plot(kind='bar')
```

[148]:
```
<Axes: xlabel='Gender'>
```

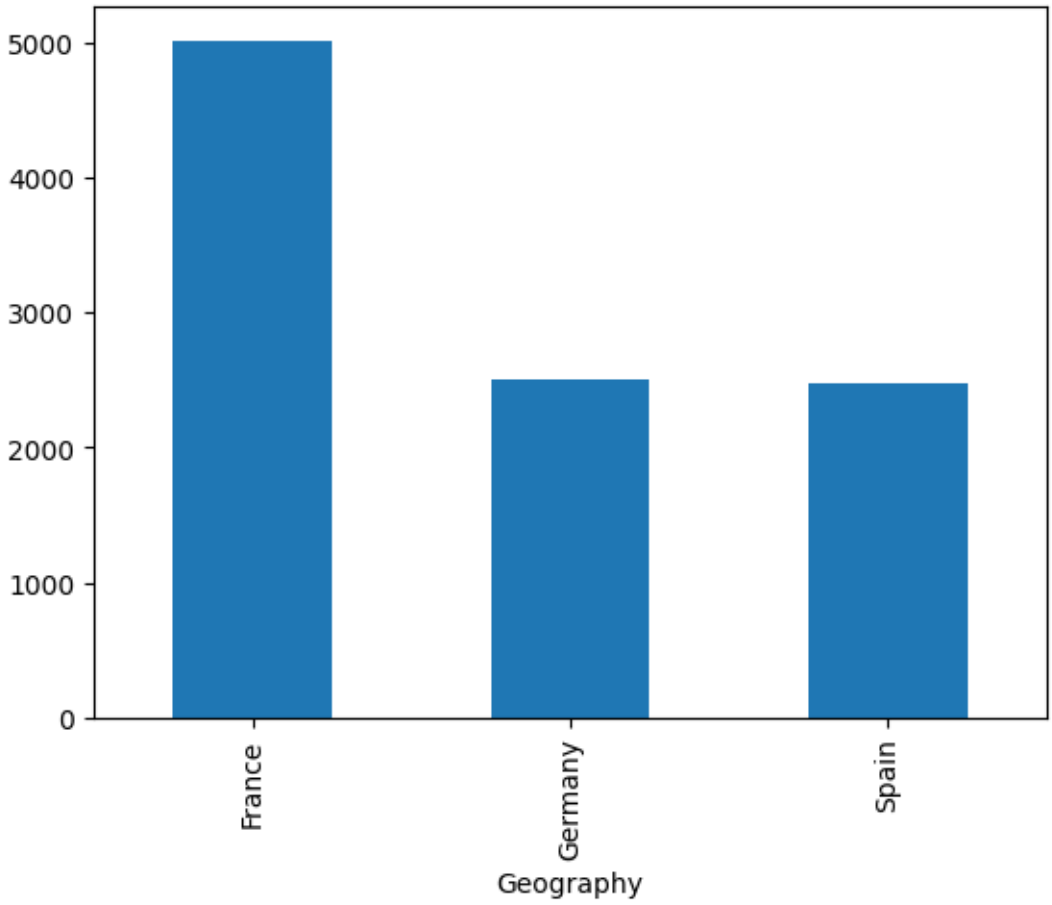


[149]:
```
grp = df.groupby('Gender')['Exited'].value_counts()
grp
```

[149]:
```
Gender  Exited
Female  0         3404
        1         1139
Male    0         4559
        1          898
Name: count, dtype: int64
```

[150]:
```
df['Geography'].value_counts().plot(kind='bar')
```

[150]:
```
<Axes: xlabel='Geography'>
```



[151]:
```
df.groupby('Geography')['Exited'].value_counts()
```

[151]:
```
Geography  Exited
France     0         4204
           1          810
Germany    0         1695
           1          814
Spain      0         2064
           1          413
Name: count, dtype: int64
```

[152]:
```
df.Exited.value_counts()
```

[152]:
```
Exited
0    7963
1    2037
Name: count, dtype: int64
```

[153]:
```
df.to_csv('analytical_base_table.csv', index=None)
```

[154]:
```
pip install keras_tuner
```

```
Requirement already satisfied: keras_tuner in /opt/homebrew/lib/python3.11/site-
packages (1.4.5)
Requirement already satisfied: keras-core in /opt/homebrew/lib/python3.11/site-
packages (from keras_tuner) (0.1.7)
Requirement already satisfied: packaging in /opt/homebrew/lib/python3.11/site-
packages (from keras_tuner) (23.2)
Requirement already satisfied: requests in /opt/homebrew/lib/python3.11/site-
packages (from keras_tuner) (2.31.0)
Requirement already satisfied: kt-legacy in /opt/homebrew/lib/python3.11/site-
packages (from keras_tuner) (1.0.5)
Requirement already satisfied: absl-py in /opt/homebrew/lib/python3.11/site-
packages (from keras-core->keras_tuner) (2.0.0)
Requirement already satisfied: numpy in /opt/homebrew/lib/python3.11/site-
packages (from keras-core->keras_tuner) (1.26.1)
Requirement already satisfied: rich in /opt/homebrew/lib/python3.11/site-
packages (from keras-core->keras_tuner) (13.6.0)
Requirement already satisfied: namex in /opt/homebrew/lib/python3.11/site-
packages (from keras-core->keras_tuner) (0.0.7)
Requirement already satisfied: h5py in /opt/homebrew/lib/python3.11/site-
packages (from keras-core->keras_tuner) (3.10.0)
Requirement already satisfied: dm-tree in /opt/homebrew/lib/python3.11/site-
packages (from keras-core->keras_tuner) (0.1.8)
Requirement already satisfied: charset-normalizer<4,>=2 in
/opt/homebrew/lib/python3.11/site-packages (from requests->keras_tuner) (3.3.0)
Requirement already satisfied: idna<4,>=2.5 in
/opt/homebrew/lib/python3.11/site-packages (from requests->keras_tuner) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/opt/homebrew/lib/python3.11/site-packages (from requests->keras_tuner) (2.0.6)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/homebrew/lib/python3.11/site-packages (from requests->keras_tuner)
(2023.7.22)
Requirement already satisfied: markdown-it-py>=2.2.0 in
/opt/homebrew/lib/python3.11/site-packages (from rich->keras-core->keras_tuner)
(3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
/opt/homebrew/lib/python3.11/site-packages (from rich->keras-core->keras_tuner)
(2.16.1)
Requirement already satisfied: mdurl~=0.1 in /opt/homebrew/lib/python3.11/site-
packages (from markdown-it-py>=2.2.0->rich->keras-core->keras_tuner) (0.1.2)
Note: you may need to restart the kernel to use updated packages.
```

[155]:
```
pip install tensorflow
```

```
Requirement already satisfied: tensorflow in /opt/homebrew/lib/python3.11/site-
```

```
packages (2.14.0)
Requirement already satisfied: tensorflow-macos==2.14.0 in
/opt/homebrew/lib/python3.11/site-packages (from tensorflow) (2.14.0)
Requirement already satisfied: absl-py>=1.0.0 in
/opt/homebrew/lib/python3.11/site-packages (from tensorflow-
macos==2.14.0->tensorflow) (2.0.0)
Requirement already satisfied: astunparse>=1.6.0 in
/opt/homebrew/lib/python3.11/site-packages (from tensorflow-
macos==2.14.0->tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=23.5.26 in
/opt/homebrew/lib/python3.11/site-packages (from tensorflow-
macos==2.14.0->tensorflow) (23.5.26)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in
/opt/homebrew/lib/python3.11/site-packages (from tensorflow-
macos==2.14.0->tensorflow) (0.5.4)
Requirement already satisfied: google-pasta>=0.1.1 in
/opt/homebrew/lib/python3.11/site-packages (from tensorflow-
macos==2.14.0->tensorflow) (0.2.0)
Requirement already satisfied: h5py>=2.9.0 in /opt/homebrew/lib/python3.11/site-
packages (from tensorflow-macos==2.14.0->tensorflow) (3.10.0)
Requirement already satisfied: libclang>=13.0.0 in
/opt/homebrew/lib/python3.11/site-packages (from tensorflow-
macos==2.14.0->tensorflow) (16.0.6)
Requirement already satisfied: ml-dtypes==0.2.0 in
/opt/homebrew/lib/python3.11/site-packages (from tensorflow-
macos==2.14.0->tensorflow) (0.2.0)
Requirement already satisfied: numpy>=1.23.5 in
/opt/homebrew/lib/python3.11/site-packages (from tensorflow-
macos==2.14.0->tensorflow) (1.26.1)
Requirement already satisfied: opt-einsum>=2.3.2 in
/opt/homebrew/lib/python3.11/site-packages (from tensorflow-
macos==2.14.0->tensorflow) (3.3.0)
Requirement already satisfied: packaging in /opt/homebrew/lib/python3.11/site-
packages (from tensorflow-macos==2.14.0->tensorflow) (23.2)
Requirement already satisfied:
protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3
in /opt/homebrew/lib/python3.11/site-packages (from tensorflow-
macos==2.14.0->tensorflow) (4.24.4)
Requirement already satisfied: setuptools in /opt/homebrew/lib/python3.11/site-
packages (from tensorflow-macos==2.14.0->tensorflow) (68.2.2)
Requirement already satisfied: six>=1.12.0 in /opt/homebrew/lib/python3.11/site-
packages (from tensorflow-macos==2.14.0->tensorflow) (1.16.0)
Requirement already satisfied: termcolor>=1.1.0 in
/opt/homebrew/lib/python3.11/site-packages (from tensorflow-
macos==2.14.0->tensorflow) (2.3.0)
Requirement already satisfied: typing-extensions>=3.6.6 in
/opt/homebrew/lib/python3.11/site-packages (from tensorflow-
macos==2.14.0->tensorflow) (4.8.0)
```

```
Requirement already satisfied: wrapt<1.15,>=1.11.0 in
/opt/homebrew/lib/python3.11/site-packages (from tensorflow-
macos==2.14.0->tensorflow) (1.14.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
/opt/homebrew/lib/python3.11/site-packages (from tensorflow-
macos==2.14.0->tensorflow) (0.34.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
/opt/homebrew/lib/python3.11/site-packages (from tensorflow-
macos==2.14.0->tensorflow) (1.59.0)
Requirement already satisfied: tensorboard<2.15,>=2.14 in
/opt/homebrew/lib/python3.11/site-packages (from tensorflow-
macos==2.14.0->tensorflow) (2.14.1)
Requirement already satisfied: tensorflow-estimator<2.15,>=2.14.0 in
/opt/homebrew/lib/python3.11/site-packages (from tensorflow-
macos==2.14.0->tensorflow) (2.14.0)
Requirement already satisfied: keras<2.15,>=2.14.0 in
/opt/homebrew/lib/python3.11/site-packages (from tensorflow-
macos==2.14.0->tensorflow) (2.14.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
/opt/homebrew/lib/python3.11/site-packages (from astunparse>=1.6.0->tensorflow-
macos==2.14.0->tensorflow) (0.41.2)
Requirement already satisfied: google-auth<3,>=1.6.3 in
/opt/homebrew/lib/python3.11/site-packages (from
tensorboard<2.15,>=2.14->tensorflow-macos==2.14.0->tensorflow) (2.23.3)
Requirement already satisfied: google-auth-oauthlib<1.1,>=0.5 in
/opt/homebrew/lib/python3.11/site-packages (from
tensorboard<2.15,>=2.14->tensorflow-macos==2.14.0->tensorflow) (1.0.0)
Requirement already satisfied: markdown>=2.6.8 in
/opt/homebrew/lib/python3.11/site-packages (from
tensorboard<2.15,>=2.14->tensorflow-macos==2.14.0->tensorflow) (3.5)
Requirement already satisfied: requests<3,>=2.21.0 in
/opt/homebrew/lib/python3.11/site-packages (from
tensorboard<2.15,>=2.14->tensorflow-macos==2.14.0->tensorflow) (2.31.0)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in
/opt/homebrew/lib/python3.11/site-packages (from
tensorboard<2.15,>=2.14->tensorflow-macos==2.14.0->tensorflow) (0.7.1)
Requirement already satisfied: werkzeug>=1.0.1 in
/opt/homebrew/lib/python3.11/site-packages (from
tensorboard<2.15,>=2.14->tensorflow-macos==2.14.0->tensorflow) (3.0.0)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in
/opt/homebrew/lib/python3.11/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.15,>=2.14->tensorflow-macos==2.14.0->tensorflow)
(5.3.1)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/opt/homebrew/lib/python3.11/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.15,>=2.14->tensorflow-macos==2.14.0->tensorflow)
(0.3.0)
Requirement already satisfied: rsa<5,>=3.1.4 in
```

```
/opt/homebrew/lib/python3.11/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.15,>=2.14->tensorflow-macos==2.14.0->tensorflow)
(4.9)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
/opt/homebrew/lib/python3.11/site-packages (from google-auth-
oauthlib<1.1,>=0.5->tensorboard<2.15,>=2.14->tensorflow-
macos==2.14.0->tensorflow) (1.3.1)
Requirement already satisfied: charset-normalizer<4,>=2 in
/opt/homebrew/lib/python3.11/site-packages (from
requests<3,>=2.21.0->tensorboard<2.15,>=2.14->tensorflow-
macos==2.14.0->tensorflow) (3.3.0)
Requirement already satisfied: idna<4,>=2.5 in
/opt/homebrew/lib/python3.11/site-packages (from
requests<3,>=2.21.0->tensorboard<2.15,>=2.14->tensorflow-
macos==2.14.0->tensorflow) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/opt/homebrew/lib/python3.11/site-packages (from
requests<3,>=2.21.0->tensorboard<2.15,>=2.14->tensorflow-
macos==2.14.0->tensorflow) (2.0.6)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/homebrew/lib/python3.11/site-packages (from
requests<3,>=2.21.0->tensorboard<2.15,>=2.14->tensorflow-
macos==2.14.0->tensorflow) (2023.7.22)
Requirement already satisfied: MarkupSafe>=2.1.1 in
/opt/homebrew/lib/python3.11/site-packages (from
werkzeug>=1.0.1->tensorboard<2.15,>=2.14->tensorflow-macos==2.14.0->tensorflow)
(2.1.3)
Requirement already satisfied: pyasn1<0.6.0,>=0.4.6 in
/opt/homebrew/lib/python3.11/site-packages (from pyasn1-modules>=0.2.1->google-
auth<3,>=1.6.3->tensorboard<2.15,>=2.14->tensorflow-macos==2.14.0->tensorflow)
(0.5.0)
Requirement already satisfied: oauthlib>=3.0.0 in
/opt/homebrew/lib/python3.11/site-packages (from requests-
oauthlib>=0.7.0->google-auth-
oauthlib<1.1,>=0.5->tensorboard<2.15,>=2.14->tensorflow-
macos==2.14.0->tensorflow) (3.2.2)
Note: you may need to restart the kernel to use updated packages.
```

[156]:
```
import tensorflow as tf
# from keras_tuner.tuners import RandomSearch
```

[157]:
```
df = pd.read_csv('analytical_base_table.csv')
```

[158]:
```
df.head(2)
```

[158]:

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	\
0	619	France	Female	42	2	0.00	1	
1	608	Spain	Female	41	1	83807.86	1	

```
   HasCrCard  IsActiveMember  EstimatedSalary  Exited
0          1               1        101348.88       1
1          0               1        112542.58       0
```

[159]:
```
x=df.drop(['Exited'],axis=1)
x.shape
```

[159]:
```
(10000, 10)
```

[160]:
```
y=df['Exited']
y.shape
```

[160]:
```
(10000,)
```

[161]:
```
from sklearn.model_selection import train_test_split
```

[162]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.
↪2,random_state=42,stratify=y)
```

[163]:
```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

[163]:
```
((8000, 10), (2000, 10), (8000,), (2000,))
```

[164]:
```
x_train.head(2)
```

[164]:
```
      CreditScore Geography Gender  Age  Tenure    Balance  NumOfProducts  \
2151          753    France   Male   57       7       0.00              1
8392          739   Germany   Male   32       3  102128.27              1

      HasCrCard  IsActiveMember  EstimatedSalary
2151          1               0        159475.08
8392          1               0         63981.37
```

[165]:
```
x_train.reset_index(drop=True,inplace=True)
```

[166]:
```
x_train.head(2)
```

[166]:
```
   CreditScore Geography Gender  Age  Tenure    Balance  NumOfProducts  \
0          753    France   Male   57       7       0.00              1
1          739   Germany   Male   32       3  102128.27              1

   HasCrCard  IsActiveMember  EstimatedSalary
0          1               0        159475.08
1          1               0         63981.37
```

[167]:
```
from sklearn.preprocessing import OneHotEncoder
```

[168]:
```
ohe = OneHotEncoder(drop='first',sparse=False,handle_unknown='ignore')
```

[169]:
```
ohe.fit(x_train[['Gender','Geography']])
```

[169]:
```
OneHotEncoder(drop='first', handle_unknown='ignore', sparse=False,
              sparse_output=False)
```

[170]:
```
x_train_encoded = ohe.transform(x_train[['Gender','Geography']])
```

[171]:
```
x_train_encoded
```

[171]:
```
array([[1., 0., 0.],
       [1., 1., 0.],
       [0., 1., 0.],
       …,
       [0., 0., 0.],
       [1., 0., 1.],
       [1., 0., 1.]])
```

[172]:
```
x_train_new = pd.DataFrame(x_train_encoded, columns = ohe.
 ↪get_feature_names_out(['Gender', 'Geography']))
x_train_new
```

[172]:

	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 x 3 columns]

[173]:
```
x_train1 = pd.concat([x_train,x_train_new],axis=1)
x_train1.head(2)
```

[173]:

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	\
0	753	France	Male	57	7	0.00	1	
1	739	Germany	Male	32	3	102128.27	1	

	HasCrCard	IsActiveMember	EstimatedSalary	Gender_Male	Geography_Germany	\
0	1	0	159475.08	1.0	0.0	
1	1	0	63981.37	1.0	1.0	

```
   Geography_Spain
0              0.0
1              0.0
```

[174]:
```
x_train1.drop(['Geography','Gender'],axis=1,inplace=True)
x_train1.shape
```

[174]:
```
(8000, 11)
```

[175]:
```
x_test.head(2)
```

[175]:
```
      CreditScore Geography Gender  Age  Tenure    Balance  NumOfProducts  \
5702          585    France   Male   36       7       0.00              2
3667          525   Germany   Male   33       4  131023.76              2

      HasCrCard  IsActiveMember  EstimatedSalary
5702          1               0         94283.09
3667          0               0         55072.93
```

[176]:
```
x_test.reset_index(drop=True,inplace=True)
x_test.head(2)
```

[176]:
```
   CreditScore Geography Gender  Age  Tenure    Balance  NumOfProducts  \
0          585    France   Male   36       7       0.00              2
1          525   Germany   Male   33       4  131023.76              2

   HasCrCard  IsActiveMember  EstimatedSalary
0          1               0         94283.09
1          0               0         55072.93
```

[177]:
```
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_out(['Gender','Geography']))
x_test_new
```

[177]:
```
      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 x 3 columns]
```

[178]:
```
x_test1 = pd.concat([x_test,x_test_new],axis=1)
x_test1.head(2)
```

[178]:
```
   CreditScore Geography Gender  Age  Tenure    Balance  NumOfProducts  \
0          585    France   Male   36       7       0.00              2
1          525   Germany   Male   33       4  131023.76              2

   HasCrCard  IsActiveMember  EstimatedSalary  Gender_Male  Geography_Germany  \
0          1               0         94283.09          1.0                0.0
1          0               0         55072.93          1.0                1.0

   Geography_Spain
0              0.0
1              0.0
```

[179]:
```
x_test1.drop(['Geography','Gender'],axis=1,inplace=True)
x_test1.head()
```

[179]:
```
   CreditScore  Age  Tenure    Balance  NumOfProducts  HasCrCard  \
0          585   36       7       0.00              2          1
1          525   33       4  131023.76              2          0
2          557   40       4       0.00              2          0
3          639   34       5  139393.19              2          0
4          640   34       3   77826.80              1          1

   IsActiveMember  EstimatedSalary  Gender_Male  Geography_Germany  \
0               0         94283.09          1.0                0.0
1               0         55072.93          1.0                1.0
2               1        105433.53          0.0                0.0
3               0         33950.08          1.0                0.0
4               1        168544.85          0.0                0.0

   Geography_Spain
0              0.0
1              0.0
2              1.0
3              1.0
4              1.0
```

[180]:
```
from sklearn.preprocessing import MinMaxScaler
```

[181]:
```
sc = MinMaxScaler()
sc.fit(x_train1)
```

```
x_train1_sc = sc.transform(x_train1)
x_test1_sc = sc.transform(x_test1)
```

[182]:
```
x_train1_sc.shape
```

[182]:
```
(8000, 11)
```

[183]:
```
type(x_train1_sc)
```

[183]:
```
numpy.ndarray
```

[184]:
```
import tensorflow.keras as tk
```

[185]:
```
# instantiate the model
model = tk.Sequential()
```

[186]:
```
# 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_uniform'))
# Adding the second hidden layer
model.add(tk.layers.
 ↪Dense(units=6,activation='relu',kernel_initializer='he_uniform'))
# Adding the output layer
model.add(tk.layers.
 ↪Dense(units=1,activation='sigmoid',kernel_initializer='glorot_uniform'))
```

[187]:
```
# Compiling the model
model.
 ↪compile(optimizer='Adam',loss='binary_crossentropy',metrics=['Precision','accuracy'])
```

[188]:
```
model.summary()
```

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

 dense_1 (Dense)             (None, 6)                 42

 dense_2 (Dense)             (None, 1)                 7

=================================================================
Total params: 121 (484.00 Byte)
Trainable params: 121 (484.00 Byte)
```

```
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

[189]:
```
model.summary()
```

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

 dense_1 (Dense)             (None, 6)                 42

 dense_2 (Dense)             (None, 1)                 7

=================================================================
Total params: 121 (484.00 Byte)
Trainable params: 121 (484.00 Byte)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

[190]:
```
x_train1_sc.shape,x_test1_sc.shape,y_train.shape,y_test.shape
```

[190]:
```
((8000, 11), (2000, 11), (8000,), (2000,))
```

[191]:
```
import time
```

[192]:
```
# Training the model
start = time.time()
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 [==============================] - 0s 828us/step - loss: 0.6650 -
precision: 0.1294 - accuracy: 0.6474 - val_loss: 0.5698 - val_precision:
0.0000e+00 - val_accuracy: 0.7965
Epoch 2/100
250/250 [==============================] - 0s 483us/step - loss: 0.5300 -
precision: 0.0000e+00 - accuracy: 0.7962 - val_loss: 0.5069 - val_precision:
0.0000e+00 - val_accuracy: 0.7965
Epoch 3/100
250/250 [==============================] - 0s 508us/step - loss: 0.4936 -
precision: 0.0000e+00 - accuracy: 0.7962 - val_loss: 0.4866 - val_precision:
0.0000e+00 - val_accuracy: 0.7965
```

```
Epoch 4/100
250/250 [==============================] - 0s 508us/step - loss: 0.4703 -
precision: 0.0000e+00 - accuracy: 0.7962 - val_loss: 0.4684 - val_precision:
0.0000e+00 - val_accuracy: 0.7965
Epoch 5/100
250/250 [==============================] - 0s 498us/step - loss: 0.4576 -
precision: 0.0000e+00 - accuracy: 0.7962 - val_loss: 0.4593 - val_precision:
0.0000e+00 - val_accuracy: 0.7965
Epoch 6/100
250/250 [==============================] - 0s 501us/step - loss: 0.4510 -
precision: 0.0000e+00 - accuracy: 0.7962 - val_loss: 0.4521 - val_precision:
0.0000e+00 - val_accuracy: 0.7965
Epoch 7/100
250/250 [==============================] - 0s 498us/step - loss: 0.4459 -
precision: 0.7143 - accuracy: 0.8011 - val_loss: 0.4483 - val_precision: 0.6234
- val_accuracy: 0.8060
Epoch 8/100
250/250 [==============================] - 0s 511us/step - loss: 0.4417 -
precision: 0.6479 - accuracy: 0.8067 - val_loss: 0.4438 - val_precision: 0.6203
- val_accuracy: 0.8060
Epoch 9/100
250/250 [==============================] - 0s 501us/step - loss: 0.4389 -
precision: 0.6399 - accuracy: 0.8098 - val_loss: 0.4406 - val_precision: 0.6308
- val_accuracy: 0.8135
Epoch 10/100
250/250 [==============================] - 0s 494us/step - loss: 0.4360 -
precision: 0.6347 - accuracy: 0.8131 - val_loss: 0.4385 - val_precision: 0.5723
- val_accuracy: 0.8080
Epoch 11/100
250/250 [==============================] - 0s 480us/step - loss: 0.4331 -
precision: 0.6115 - accuracy: 0.8123 - val_loss: 0.4350 - val_precision: 0.6357
- val_accuracy: 0.8155
Epoch 12/100
250/250 [==============================] - 0s 483us/step - loss: 0.4309 -
precision: 0.6190 - accuracy: 0.8138 - val_loss: 0.4322 - val_precision: 0.6333
- val_accuracy: 0.8165
Epoch 13/100
250/250 [==============================] - 0s 481us/step - loss: 0.4292 -
precision: 0.6163 - accuracy: 0.8138 - val_loss: 0.4294 - val_precision: 0.6449
- val_accuracy: 0.8165
Epoch 14/100
250/250 [==============================] - 0s 485us/step - loss: 0.4273 -
precision: 0.6191 - accuracy: 0.8146 - val_loss: 0.4281 - val_precision: 0.6369
- val_accuracy: 0.8180
Epoch 15/100
250/250 [==============================] - 0s 498us/step - loss: 0.4266 -
precision: 0.6219 - accuracy: 0.8158 - val_loss: 0.4268 - val_precision: 0.6589
- val_accuracy: 0.8170
```

```
Epoch 16/100
250/250 [==============================] - 0s 501us/step - loss: 0.4252 -
precision: 0.6244 - accuracy: 0.8159 - val_loss: 0.4261 - val_precision: 0.6185
- val_accuracy: 0.8170
Epoch 17/100
250/250 [==============================] - 0s 509us/step - loss: 0.4243 -
precision: 0.6200 - accuracy: 0.8164 - val_loss: 0.4244 - val_precision: 0.6387
- val_accuracy: 0.8180
Epoch 18/100
250/250 [==============================] - 0s 511us/step - loss: 0.4235 -
precision: 0.6350 - accuracy: 0.8183 - val_loss: 0.4237 - val_precision: 0.6258
- val_accuracy: 0.8160
Epoch 19/100
250/250 [==============================] - 0s 501us/step - loss: 0.4233 -
precision: 0.6304 - accuracy: 0.8181 - val_loss: 0.4228 - val_precision: 0.6352
- val_accuracy: 0.8180
Epoch 20/100
250/250 [==============================] - 0s 499us/step - loss: 0.4221 -
precision: 0.6374 - accuracy: 0.8194 - val_loss: 0.4229 - val_precision: 0.6301
- val_accuracy: 0.8190
Epoch 21/100
250/250 [==============================] - 0s 494us/step - loss: 0.4218 -
precision: 0.6272 - accuracy: 0.8180 - val_loss: 0.4216 - val_precision: 0.6375
- val_accuracy: 0.8185
Epoch 22/100
250/250 [==============================] - 0s 489us/step - loss: 0.4208 -
precision: 0.6365 - accuracy: 0.8196 - val_loss: 0.4217 - val_precision: 0.6617
- val_accuracy: 0.8180
Epoch 23/100
250/250 [==============================] - 0s 483us/step - loss: 0.4209 -
precision: 0.6480 - accuracy: 0.8206 - val_loss: 0.4202 - val_precision: 0.6474
- val_accuracy: 0.8195
Epoch 24/100
250/250 [==============================] - 0s 487us/step - loss: 0.4198 -
precision: 0.6421 - accuracy: 0.8217 - val_loss: 0.4206 - val_precision: 0.6393
- val_accuracy: 0.8220
Epoch 25/100
250/250 [==============================] - 0s 493us/step - loss: 0.4196 -
precision: 0.6406 - accuracy: 0.8210 - val_loss: 0.4191 - val_precision: 0.6453
- val_accuracy: 0.8215
Epoch 26/100
250/250 [==============================] - 0s 504us/step - loss: 0.4185 -
precision: 0.6373 - accuracy: 0.8204 - val_loss: 0.4203 - val_precision: 0.6557
- val_accuracy: 0.8155
Epoch 27/100
250/250 [==============================] - 0s 498us/step - loss: 0.4185 -
precision: 0.6428 - accuracy: 0.8211 - val_loss: 0.4173 - val_precision: 0.6625
- val_accuracy: 0.8225
```

```
Epoch 28/100
250/250 [==============================] - 0s 499us/step - loss: 0.4177 -
precision: 0.6486 - accuracy: 0.8223 - val_loss: 0.4173 - val_precision: 0.6409
- val_accuracy: 0.8220
Epoch 29/100
250/250 [==============================] - 0s 495us/step - loss: 0.4170 -
precision: 0.6593 - accuracy: 0.8251 - val_loss: 0.4156 - val_precision: 0.6477
- val_accuracy: 0.8225
Epoch 30/100
250/250 [==============================] - 0s 501us/step - loss: 0.4168 -
precision: 0.6429 - accuracy: 0.8227 - val_loss: 0.4153 - val_precision: 0.6543
- val_accuracy: 0.8215
Epoch 31/100
250/250 [==============================] - 0s 494us/step - loss: 0.4158 -
precision: 0.6526 - accuracy: 0.8234 - val_loss: 0.4147 - val_precision: 0.6828
- val_accuracy: 0.8230
Epoch 32/100
250/250 [==============================] - 0s 489us/step - loss: 0.4149 -
precision: 0.6560 - accuracy: 0.8242 - val_loss: 0.4141 - val_precision: 0.6765
- val_accuracy: 0.8265
Epoch 33/100
250/250 [==============================] - 0s 484us/step - loss: 0.4144 -
precision: 0.6648 - accuracy: 0.8259 - val_loss: 0.4127 - val_precision: 0.6871
- val_accuracy: 0.8270
Epoch 34/100
250/250 [==============================] - 0s 505us/step - loss: 0.4140 -
precision: 0.6765 - accuracy: 0.8278 - val_loss: 0.4121 - val_precision: 0.7006
- val_accuracy: 0.8280
Epoch 35/100
250/250 [==============================] - 0s 507us/step - loss: 0.4135 -
precision: 0.6815 - accuracy: 0.8269 - val_loss: 0.4125 - val_precision: 0.6742
- val_accuracy: 0.8275
Epoch 36/100
250/250 [==============================] - 0s 499us/step - loss: 0.4136 -
precision: 0.6869 - accuracy: 0.8279 - val_loss: 0.4108 - val_precision: 0.7379
- val_accuracy: 0.8310
Epoch 37/100
250/250 [==============================] - 0s 508us/step - loss: 0.4132 -
precision: 0.7035 - accuracy: 0.8307 - val_loss: 0.4099 - val_precision: 0.7160
- val_accuracy: 0.8315
Epoch 38/100
250/250 [==============================] - 0s 502us/step - loss: 0.4119 -
precision: 0.7046 - accuracy: 0.8309 - val_loss: 0.4094 - val_precision: 0.7160
- val_accuracy: 0.8330
Epoch 39/100
250/250 [==============================] - 0s 497us/step - loss: 0.4115 -
precision: 0.7062 - accuracy: 0.8298 - val_loss: 0.4100 - val_precision: 0.6771
- val_accuracy: 0.8305
```

```
Epoch 40/100
250/250 [==============================] - 0s 493us/step - loss: 0.4113 -
precision: 0.7019 - accuracy: 0.8306 - val_loss: 0.4093 - val_precision: 0.7256
- val_accuracy: 0.8335
Epoch 41/100
250/250 [==============================] - 0s 474us/step - loss: 0.4103 -
precision: 0.7088 - accuracy: 0.8317 - val_loss: 0.4093 - val_precision: 0.7088
- val_accuracy: 0.8345
Epoch 42/100
250/250 [==============================] - 0s 479us/step - loss: 0.4100 -
precision: 0.7080 - accuracy: 0.8303 - val_loss: 0.4077 - val_precision: 0.7423
- val_accuracy: 0.8360
Epoch 43/100
250/250 [==============================] - 0s 474us/step - loss: 0.4100 -
precision: 0.7103 - accuracy: 0.8316 - val_loss: 0.4082 - val_precision: 0.7184
- val_accuracy: 0.8345
Epoch 44/100
250/250 [==============================] - 0s 475us/step - loss: 0.4099 -
precision: 0.7077 - accuracy: 0.8313 - val_loss: 0.4076 - val_precision: 0.7533
- val_accuracy: 0.8345
Epoch 45/100
250/250 [==============================] - 0s 484us/step - loss: 0.4095 -
precision: 0.7165 - accuracy: 0.8317 - val_loss: 0.4070 - val_precision: 0.7595
- val_accuracy: 0.8375
Epoch 46/100
250/250 [==============================] - 0s 485us/step - loss: 0.4093 -
precision: 0.7175 - accuracy: 0.8326 - val_loss: 0.4074 - val_precision: 0.7159
- val_accuracy: 0.8345
Epoch 47/100
250/250 [==============================] - 0s 492us/step - loss: 0.4083 -
precision: 0.7233 - accuracy: 0.8334 - val_loss: 0.4082 - val_precision: 0.7652
- val_accuracy: 0.8315
Epoch 48/100
250/250 [==============================] - 0s 486us/step - loss: 0.4087 -
precision: 0.7231 - accuracy: 0.8341 - val_loss: 0.4067 - val_precision: 0.7391
- val_accuracy: 0.8350
Epoch 49/100
250/250 [==============================] - 0s 498us/step - loss: 0.4081 -
precision: 0.7282 - accuracy: 0.8336 - val_loss: 0.4065 - val_precision: 0.7687
- val_accuracy: 0.8360
Epoch 50/100
250/250 [==============================] - 0s 489us/step - loss: 0.4077 -
precision: 0.7385 - accuracy: 0.8353 - val_loss: 0.4063 - val_precision: 0.7299
- val_accuracy: 0.8365
Epoch 51/100
250/250 [==============================] - 0s 496us/step - loss: 0.4076 -
precision: 0.7308 - accuracy: 0.8338 - val_loss: 0.4070 - val_precision: 0.6939
- val_accuracy: 0.8345
```

```
Epoch 52/100
250/250 [==============================] - 0s 488us/step - loss: 0.4076 -
precision: 0.7240 - accuracy: 0.8334 - val_loss: 0.4051 - val_precision: 0.7671
- val_accuracy: 0.8355
Epoch 53/100
250/250 [==============================] - 0s 481us/step - loss: 0.4063 -
precision: 0.7316 - accuracy: 0.8349 - val_loss: 0.4059 - val_precision: 0.7000
- val_accuracy: 0.8345
Epoch 54/100
250/250 [==============================] - 0s 494us/step - loss: 0.4069 -
precision: 0.7196 - accuracy: 0.8332 - val_loss: 0.4047 - val_precision: 0.7548
- val_accuracy: 0.8360
Epoch 55/100
250/250 [==============================] - 0s 490us/step - loss: 0.4075 -
precision: 0.7323 - accuracy: 0.8340 - val_loss: 0.4063 - val_precision: 0.7135
- val_accuracy: 0.8360
Epoch 56/100
250/250 [==============================] - 0s 493us/step - loss: 0.4068 -
precision: 0.7240 - accuracy: 0.8340 - val_loss: 0.4041 - val_precision: 0.7425
- val_accuracy: 0.8370
Epoch 57/100
250/250 [==============================] - 0s 500us/step - loss: 0.4066 -
precision: 0.7285 - accuracy: 0.8335 - val_loss: 0.4050 - val_precision: 0.7817
- val_accuracy: 0.8365
Epoch 58/100
250/250 [==============================] - 0s 498us/step - loss: 0.4062 -
precision: 0.7374 - accuracy: 0.8351 - val_loss: 0.4047 - val_precision: 0.7341
- val_accuracy: 0.8370
Epoch 59/100
250/250 [==============================] - 0s 490us/step - loss: 0.4056 -
precision: 0.7342 - accuracy: 0.8353 - val_loss: 0.4069 - val_precision: 0.7969
- val_accuracy: 0.8345
Epoch 60/100
250/250 [==============================] - 0s 491us/step - loss: 0.4055 -
precision: 0.7406 - accuracy: 0.8347 - val_loss: 0.4044 - val_precision: 0.7651
- val_accuracy: 0.8360
Epoch 61/100
250/250 [==============================] - 0s 494us/step - loss: 0.4067 -
precision: 0.7337 - accuracy: 0.8349 - val_loss: 0.4049 - val_precision: 0.7755
- val_accuracy: 0.8370
Epoch 62/100
250/250 [==============================] - 0s 491us/step - loss: 0.4055 -
precision: 0.7419 - accuracy: 0.8354 - val_loss: 0.4051 - val_precision: 0.7832
- val_accuracy: 0.8370
Epoch 63/100
250/250 [==============================] - 0s 491us/step - loss: 0.4053 -
precision: 0.7500 - accuracy: 0.8367 - val_loss: 0.4050 - val_precision: 0.7793
- val_accuracy: 0.8370
```

```
Epoch 64/100
250/250 [==============================] - 0s 490us/step - loss: 0.4053 -
precision: 0.7288 - accuracy: 0.8346 - val_loss: 0.4043 - val_precision: 0.7742
- val_accuracy: 0.8390
Epoch 65/100
250/250 [==============================] - 0s 496us/step - loss: 0.4050 -
precision: 0.7320 - accuracy: 0.8350 - val_loss: 0.4042 - val_precision: 0.7688
- val_accuracy: 0.8395
Epoch 66/100
250/250 [==============================] - 0s 490us/step - loss: 0.4043 -
precision: 0.7423 - accuracy: 0.8357 - val_loss: 0.4038 - val_precision: 0.7712
- val_accuracy: 0.8380
Epoch 67/100
250/250 [==============================] - 0s 501us/step - loss: 0.4050 -
precision: 0.7410 - accuracy: 0.8365 - val_loss: 0.4047 - val_precision: 0.7872
- val_accuracy: 0.8370
Epoch 68/100
250/250 [==============================] - 0s 491us/step - loss: 0.4046 -
precision: 0.7419 - accuracy: 0.8354 - val_loss: 0.4046 - val_precision: 0.6750
- val_accuracy: 0.8315
Epoch 69/100
250/250 [==============================] - 0s 491us/step - loss: 0.4039 -
precision: 0.7419 - accuracy: 0.8372 - val_loss: 0.4040 - val_precision: 0.7662
- val_accuracy: 0.8375
Epoch 70/100
250/250 [==============================] - 0s 492us/step - loss: 0.4035 -
precision: 0.7394 - accuracy: 0.8357 - val_loss: 0.4032 - val_precision: 0.7410
- val_accuracy: 0.8365
Epoch 71/100
250/250 [==============================] - 0s 500us/step - loss: 0.4034 -
precision: 0.7485 - accuracy: 0.8365 - val_loss: 0.4027 - val_precision: 0.7857
- val_accuracy: 0.8405
Epoch 72/100
250/250 [==============================] - 0s 491us/step - loss: 0.4028 -
precision: 0.7293 - accuracy: 0.8356 - val_loss: 0.4026 - val_precision: 0.7440
- val_accuracy: 0.8375
Epoch 73/100
250/250 [==============================] - 0s 490us/step - loss: 0.4025 -
precision: 0.7428 - accuracy: 0.8364 - val_loss: 0.4015 - val_precision: 0.7673
- val_accuracy: 0.8390
Epoch 74/100
250/250 [==============================] - 0s 488us/step - loss: 0.4020 -
precision: 0.7358 - accuracy: 0.8357 - val_loss: 0.4023 - val_precision: 0.6984
- val_accuracy: 0.8340
Epoch 75/100
250/250 [==============================] - 0s 488us/step - loss: 0.4012 -
precision: 0.7388 - accuracy: 0.8363 - val_loss: 0.4031 - val_precision: 0.6471
- val_accuracy: 0.8290
```

```
Epoch 76/100
250/250 [==============================] - 0s 491us/step - loss: 0.4010 -
precision: 0.7312 - accuracy: 0.8363 - val_loss: 0.4002 - val_precision: 0.7821
- val_accuracy: 0.8405
Epoch 77/100
250/250 [==============================] - 0s 488us/step - loss: 0.4005 -
precision: 0.7430 - accuracy: 0.8374 - val_loss: 0.3995 - val_precision: 0.7862
- val_accuracy: 0.8420
Epoch 78/100
250/250 [==============================] - 0s 491us/step - loss: 0.3994 -
precision: 0.7314 - accuracy: 0.8367 - val_loss: 0.3983 - val_precision: 0.7651
- val_accuracy: 0.8405
Epoch 79/100
250/250 [==============================] - 0s 487us/step - loss: 0.3981 -
precision: 0.7378 - accuracy: 0.8375 - val_loss: 0.4001 - val_precision: 0.8239
- val_accuracy: 0.8425
Epoch 80/100
250/250 [==============================] - 0s 493us/step - loss: 0.3977 -
precision: 0.7315 - accuracy: 0.8370 - val_loss: 0.3974 - val_precision: 0.7875
- val_accuracy: 0.8425
Epoch 81/100
250/250 [==============================] - 0s 489us/step - loss: 0.3967 -
precision: 0.7438 - accuracy: 0.8381 - val_loss: 0.3966 - val_precision: 0.7670
- val_accuracy: 0.8435
Epoch 82/100
250/250 [==============================] - 0s 493us/step - loss: 0.3962 -
precision: 0.7338 - accuracy: 0.8378 - val_loss: 0.3964 - val_precision: 0.7005
- val_accuracy: 0.8380
Epoch 83/100
250/250 [==============================] - 0s 490us/step - loss: 0.3953 -
precision: 0.7376 - accuracy: 0.8395 - val_loss: 0.3974 - val_precision: 0.8079
- val_accuracy: 0.8430
Epoch 84/100
250/250 [==============================] - 0s 492us/step - loss: 0.3942 -
precision: 0.7411 - accuracy: 0.8386 - val_loss: 0.3946 - val_precision: 0.7950
- val_accuracy: 0.8440
Epoch 85/100
250/250 [==============================] - 0s 490us/step - loss: 0.3933 -
precision: 0.7419 - accuracy: 0.8391 - val_loss: 0.3926 - val_precision: 0.7317
- val_accuracy: 0.8440
Epoch 86/100
250/250 [==============================] - 0s 488us/step - loss: 0.3925 -
precision: 0.7473 - accuracy: 0.8415 - val_loss: 0.3909 - val_precision: 0.7709
- val_accuracy: 0.8450
Epoch 87/100
250/250 [==============================] - 0s 500us/step - loss: 0.3924 -
precision: 0.7448 - accuracy: 0.8404 - val_loss: 0.3916 - val_precision: 0.7952
- val_accuracy: 0.8455
```

```
Epoch 88/100
250/250 [==============================] - 0s 489us/step - loss: 0.3907 -
precision: 0.7483 - accuracy: 0.8409 - val_loss: 0.3903 - val_precision: 0.7907
- val_accuracy: 0.8465
Epoch 89/100
250/250 [==============================] - 0s 480us/step - loss: 0.3899 -
precision: 0.7456 - accuracy: 0.8419 - val_loss: 0.3890 - val_precision: 0.7898
- val_accuracy: 0.8475
Epoch 90/100
250/250 [==============================] - 0s 481us/step - loss: 0.3889 -
precision: 0.7409 - accuracy: 0.8411 - val_loss: 0.3885 - val_precision: 0.8193
- val_accuracy: 0.8495
Epoch 91/100
250/250 [==============================] - 0s 478us/step - loss: 0.3881 -
precision: 0.7436 - accuracy: 0.8414 - val_loss: 0.3872 - val_precision: 0.7914
- val_accuracy: 0.8510
Epoch 92/100
250/250 [==============================] - 0s 479us/step - loss: 0.3867 -
precision: 0.7524 - accuracy: 0.8429 - val_loss: 0.3853 - val_precision: 0.7454
- val_accuracy: 0.8495
Epoch 93/100
250/250 [==============================] - 0s 489us/step - loss: 0.3844 -
precision: 0.7451 - accuracy: 0.8436 - val_loss: 0.3830 - val_precision: 0.7789
- val_accuracy: 0.8520
Epoch 94/100
250/250 [==============================] - 0s 489us/step - loss: 0.3825 -
precision: 0.7552 - accuracy: 0.8450 - val_loss: 0.3832 - val_precision: 0.8415
- val_accuracy: 0.8525
Epoch 95/100
250/250 [==============================] - 0s 489us/step - loss: 0.3808 -
precision: 0.7581 - accuracy: 0.8461 - val_loss: 0.3806 - val_precision: 0.7772
- val_accuracy: 0.8525
Epoch 96/100
250/250 [==============================] - 0s 490us/step - loss: 0.3762 -
precision: 0.7607 - accuracy: 0.8482 - val_loss: 0.3771 - val_precision: 0.7545
- val_accuracy: 0.8535
Epoch 97/100
250/250 [==============================] - 0s 494us/step - loss: 0.3726 -
precision: 0.7620 - accuracy: 0.8494 - val_loss: 0.3738 - val_precision: 0.7850
- val_accuracy: 0.8575
Epoch 98/100
250/250 [==============================] - 0s 490us/step - loss: 0.3685 -
precision: 0.7662 - accuracy: 0.8506 - val_loss: 0.3715 - val_precision: 0.8324
- val_accuracy: 0.8560
Epoch 99/100
250/250 [==============================] - 0s 488us/step - loss: 0.3660 -
precision: 0.7677 - accuracy: 0.8510 - val_loss: 0.3676 - val_precision: 0.7706
- val_accuracy: 0.8555
```

```
Epoch 100/100
250/250 [==============================] - 0s 489us/step - loss: 0.3633 -
precision: 0.7628 - accuracy: 0.8516 - val_loss: 0.3652 - val_precision: 0.8065
- val_accuracy: 0.8535
12.77235221862793
```

[ ]: