ANN Model

April 8, 2024

0.1 Modeling

[ ]:
```
# !pip uninstall scikit-learn --yes
# !pip uninstall imblearn --yes
```

```
WARNING: Skipping scikit-learn as it is not installed.

WARNING: Skipping imblearn as it is not installed.
```

[ ]:
```
# !pip install scikit-learn==1.2.2
# !pip install imblearn
```

```
Collecting scikit-learn==1.2.2
  Obtaining dependency information for scikit-learn==1.2.2 from https://files.py
thonhosted.org/packages/2f/fd/9fcbe7fe94150e72d87120cbc462bde1971c3674e726b81f4a
4c4fdfa8e1/scikit_learn-1.2.2-cp311-cp311-macosx_12_0_arm64.whl.metadata
  Downloading scikit_learn-1.2.2-cp311-cp311-macosx_12_0_arm64.whl.metadata (11
kB)
Requirement already satisfied: numpy>=1.17.3 in
/Users/aadityakasbekar/anaconda3/lib/python3.11/site-packages (from scikit-
learn==1.2.2) (1.24.3)
Requirement already satisfied: scipy>=1.3.2 in
/Users/aadityakasbekar/anaconda3/lib/python3.11/site-packages (from scikit-
learn==1.2.2) (1.11.1)
Requirement already satisfied: joblib>=1.1.1 in
/Users/aadityakasbekar/anaconda3/lib/python3.11/site-packages (from scikit-
learn==1.2.2) (1.2.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/Users/aadityakasbekar/anaconda3/lib/python3.11/site-packages (from scikit-
learn==1.2.2) (2.2.0)
Downloading scikit_learn-1.2.2-cp311-cp311-macosx_12_0_arm64.whl (8.4 MB)
                        8.4/8.4 MB
3.9 MB/s eta 0:00:0000:0100:01
Installing collected packages: scikit-learn
Successfully installed scikit-learn-1.2.2
Collecting imblearn
  Obtaining dependency information for imblearn from https://files.pythonhosted.
org/packages/81/a7/4179e6ebfd654bd0eac0b9c06125b8b4c96a9d0a8ff9e9507eb2a26d2d7e/
```

```
imblearn-0.0-py2.py3-none-any.whl.metadata
  Downloading imblearn-0.0-py2.py3-none-any.whl.metadata (355 bytes)
Requirement already satisfied: imbalanced-learn in
/Users/aadityakasbekar/anaconda3/lib/python3.11/site-packages (from imblearn)
(0.10.1)
Requirement already satisfied: numpy>=1.17.3 in
/Users/aadityakasbekar/anaconda3/lib/python3.11/site-packages (from imbalanced-
learn->imblearn) (1.24.3)
Requirement already satisfied: scipy>=1.3.2 in
/Users/aadityakasbekar/anaconda3/lib/python3.11/site-packages (from imbalanced-
learn->imblearn) (1.11.1)
Requirement already satisfied: scikit-learn>=1.0.2 in
/Users/aadityakasbekar/anaconda3/lib/python3.11/site-packages (from imbalanced-
learn->imblearn) (1.2.2)
Requirement already satisfied: joblib>=1.1.1 in
/Users/aadityakasbekar/anaconda3/lib/python3.11/site-packages (from imbalanced-
learn->imblearn) (1.2.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/Users/aadityakasbekar/anaconda3/lib/python3.11/site-packages (from imbalanced-
learn->imblearn) (2.2.0)
Downloading imblearn-0.0-py2.py3-none-any.whl (1.9 kB)
Installing collected packages: imblearn
Successfully installed imblearn-0.0
```

[ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import gc
pd.options.mode.chained_assignment = None

root = './data/'

from sklearn.model_selection import train_test_split, GridSearchCV,␣
 ↪cross_val_score
from sklearn.metrics import confusion_matrix, accuracy_score,␣
 ↪classification_report
from sklearn.metrics import roc_auc_score, roc_curve, precision_score,␣
 ↪recall_score, f1_score
from imblearn.over_sampling import SMOTE
```

[ ]:
```
df = pd.read_pickle(root + 'Finaldata.pkl')
df.head()
```

[ ]:

	user_id	product_id	total_product_orders_by_user	\
0	1	196	10.0	
1	1	10258	9.0	

```
2        1       10326                          1.0
3        1       12427                         10.0
4        1       13032                          3.0

   total_product_reorders_by_user  user_product_reorder_percentage  \
0                             9.0                         0.900000
1                             8.0                         0.888889
2                             0.0                         0.000000
3                             9.0                         0.900000
4                             2.0                         0.666667

   avg_add_to_cart_by_user  avg_days_since_last_bought  last_ordered_in  \
0                 1.400000                   17.600000             10.0
1                 3.333333                   19.555555             10.0
2                 5.000000                   28.000000              5.0
3                 3.300000                   17.600000             10.0
4                 6.333333                   21.666666             10.0

   is_reorder_3  is_reorder_2  …  total_reorders_by_user  \
0           1.0           1.0  …                      41
1           1.0           1.0  …                      41
2           0.0           0.0  …                      41
3           1.0           1.0  …                      41
4           1.0           0.0  …                      41

   reorder_propotion_by_user  average_order_size  reorder_in_order  orders_3  \
0                   0.694915                 5.9          0.705833         6
1                   0.694915                 5.9          0.705833         6
2                   0.694915                 5.9          0.705833         6
3                   0.694915                 5.9          0.705833         6
4                   0.694915                 5.9          0.705833         6

   orders_2  orders_1  reorder_3  reorder_2  reorder_1
0         6         9   0.666667        1.0   0.666667
1         6         9   0.666667        1.0   0.666667
2         6         9   0.666667        1.0   0.666667
3         6         9   0.666667        1.0   0.666667
4         6         9   0.666667        1.0   0.666667

[5 rows x 69 columns]
```

[ ]:
```python
def reduce_memory(df):

    """
    This function reduce the dataframe memory usage by converting it's type for␣
 ↪easier handling.
```

```
    Parameters: Dataframe
    Return: Dataframe
    """

    start_mem_usg = df.memory_usage().sum() / 1024**2
    print("Memory usage of properties dataframe is :",start_mem_usg," MB")

    for col in df.columns:
        if df[col].dtypes in ["int64", "int32", "int16"]:

            cmin = df[col].min()
            cmax = df[col].max()

            if cmin > np.iinfo(np.int8).min and cmax < np.iinfo(np.int8).max:
                df[col] = df[col].astype(np.int8)

            elif cmin > np.iinfo(np.int16).min and cmax < np.iinfo(np.int16).
↪max:
                df[col] = df[col].astype(np.int16)

            elif cmin > np.iinfo(np.int32).min and cmax < np.iinfo(np.int32).
↪max:
                df[col] = df[col].astype(np.int32)

        if df[col].dtypes in ["float64", "float32"]:

            cmin = df[col].min()
            cmax = df[col].max()

            if cmin > np.finfo(np.float16).min and cmax < np.finfo(np.float16).
↪max:
                df[col] = df[col].astype(np.float16)

            elif cmin > np.finfo(np.float32).min and cmax < np.finfo(np.
↪float32).max:
                df[col] = df[col].astype(np.float32)

    print("")
    print("___MEMORY USAGE AFTER COMPLETION:___")
    mem_usg = df.memory_usage().sum() / 1024**2
    print("Memory usage is: ",mem_usg," MB")
    print("This is ",100*mem_usg/start_mem_usg,"% of the initial size")

    return df
```

[ ]:
```
df = reduce_memory(df)
```

```
Memory usage of properties dataframe is : 4315.823656082153  MB

___MEMORY USAGE AFTER COMPLETION:___
Memory usage is:  1163.8177070617676  MB
This is  26.96629426509668 % of the initial size
```

[ ]:
```
df['order_diff'] = df.order_number - df.last_ordered_in
df.drop(['user_id', 'product_id'], axis = 1, inplace = True)
```

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

[ ]:
```
   total_product_orders_by_user  total_product_reorders_by_user  \
0                          10.0                             9.0
1                           9.0                             8.0
2                           1.0                             0.0
3                          10.0                             9.0
4                           3.0                             2.0

   user_product_reorder_percentage  avg_add_to_cart_by_user  \
0                         0.899902                 1.400391
1                         0.888672                 3.333984
2                         0.000000                 5.000000
3                         0.899902                 3.300781
4                         0.666504                 6.332031

   avg_days_since_last_bought  last_ordered_in  is_reorder_3  is_reorder_2  \
0                   17.593750            10.0           1.0           1.0
1                   19.562500            10.0           1.0           1.0
2                   28.000000             5.0           0.0           0.0
3                   17.593750            10.0           1.0           1.0
4                   21.671875            10.0           1.0           0.0

   is_reorder_1  order_number  …  reorder_propotion_by_user  \
0           1.0          11.0  …                   0.694824
1           1.0          11.0  …                   0.694824
2           0.0          11.0  …                   0.694824
3           1.0          11.0  …                   0.694824
4           0.0          11.0  …                   0.694824

   average_order_size  reorder_in_order  orders_3  orders_2  orders_1  \
0            5.898438          0.706055         6         6         9
1            5.898438          0.706055         6         6         9
2            5.898438          0.706055         6         6         9
3            5.898438          0.706055         6         6         9
4            5.898438          0.706055         6         6         9

   reorder_3  reorder_2  reorder_1  order_diff
```

```
0   0.666504        1.0   0.666504        1.0
1   0.666504        1.0   0.666504        1.0
2   0.666504        1.0   0.666504        6.0
3   0.666504        1.0   0.666504        1.0
4   0.666504        1.0   0.666504        1.0

[5 rows x 68 columns]
```

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

[ ]:
```
(8474661, 68)
```

[ ]:
```
label = 'reordered'
x_cols = df.columns.drop('reordered')
```

[ ]:
```
X = df[x_cols]
y = df[label]
```

[ ]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, stratify = y,␣
 ↪test_size = 0.25)

print(X_train.shape, y_train.shape)
print(X_test.shape, y_test.shape)
```

```
(6355995, 67) (6355995,)
(2118666, 67) (2118666,)
```

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

[ ]:
```
reordered
0.0    5734377
1.0     621618
Name: count, dtype: int64
```

[ ]:
```
np.ceil(y_train.value_counts()[0]/y_train.value_counts()[1])
```

[ ]:
```
10.0
```

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

[ ]:
```
reordered
0.0    1911460
1.0     207206
Name: count, dtype: int64
```

[ ]:
```
# freeing memory
del df, X, y
gc.collect()
```

[ ]: 0

0.1.1 Neural Network model

[ ]:
```
# !pip install keras
```

```
Requirement already satisfied: keras in
/Users/aadityakasbekar/anaconda3/lib/python3.11/site-packages (2.15.0)
```

[ ]:
```
import keras
from keras.models import Sequential
from keras.layers import Dense, Dropout
from keras.regularizers import l2
from keras.callbacks import History
from keras import backend as K
from sklearn.preprocessing import MinMaxScaler
```

[ ]:
```
sc = MinMaxScaler()
X_train_sc = sc.fit_transform(X_train)
X_test_sc = sc.transform(X_test)
```

[ ]:
```
input_dim =  X_train_sc.shape[1]
input_dim
```

[ ]: 67

[ ]:
```
def recall_m(y_true, y_pred):
    true_positives = K.sum(K.round(K.clip(y_true * y_pred, 0, 1)))
    possible_positives = K.sum(K.round(K.clip(y_true, 0, 1)))
    recall = true_positives / (possible_positives + K.epsilon())
    return recall

def precision_m(y_true, y_pred):
    true_positives = K.sum(K.round(K.clip(y_true * y_pred, 0, 1)))
    predicted_positives = K.sum(K.round(K.clip(y_pred, 0, 1)))
    precision = true_positives / (predicted_positives + K.epsilon())
    return precision

def f1_m(y_true, y_pred):
    precision = precision_m(y_true, y_pred)
    recall = recall_m(y_true, y_pred)
    return 2*((precision*recall)/(precision+recall+K.epsilon()))
```

[ ]:
```
history = History()

classifier = Sequential()

classifier.add(Dense(units = 64, activation = 'relu', input_dim = input_dim))
```

```
classifier.add(Dense(units = 15, activation = 'relu'))
classifier.add(Dense(units = 4, activation = 'relu'))
classifier.add(Dense(units = 1, activation ='sigmoid'))

classifier.compile(optimizer = "adam", loss = 'binary_crossentropy', metrics =␣
 ↪['accuracy', f1_m, precision_m, recall_m])

classifier.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 64)                4352

 dense_1 (Dense)             (None, 15)                975

 dense_2 (Dense)             (None, 4)                 64

 dense_3 (Dense)             (None, 1)                 5

=================================================================
Total params: 5396 (21.08 KB)
Trainable params: 5396 (21.08 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

[ ]:
```
%%time
# fit the model
classifier.fit(X_train_sc, y_train, epochs=50, batch_size=512,␣
 ↪validation_split=0.15, verbose=1,class_weight= {0:1, 1:10},
          callbacks = [history, keras.callbacks.
 ↪EarlyStopping(monitor='val_loss',
                                                        min_delta=0,␣
 ↪patience=10, verbose=0, mode='auto')])
```

```
Epoch 1/50
10552/10552 [==============================] - 8s 740us/step - loss: 0.9739 -
accuracy: 0.7186 - f1_m: 0.3518 - precision_m: 0.2300 - recall_m: 0.7708 -
val_loss: 0.5506 - val_accuracy: 0.7093 - val_f1_m: 0.3474 - val_precision_m:
0.2229 - val_recall_m: 0.7970
Epoch 2/50
10552/10552 [==============================] - 7s 706us/step - loss: 0.9571 -
accuracy: 0.7319 - f1_m: 0.3597 - precision_m: 0.2361 - recall_m: 0.7699 -
val_loss: 0.5407 - val_accuracy: 0.7195 - val_f1_m: 0.3534 - val_precision_m:
0.2285 - val_recall_m: 0.7895
Epoch 3/50
10552/10552 [==============================] - 7s 703us/step - loss: 0.9543 -
```

```
accuracy: 0.7358 - f1_m: 0.3621 - precision_m: 0.2382 - recall_m: 0.7678 -
val_loss: 0.5559 - val_accuracy: 0.7061 - val_f1_m: 0.3470 - val_precision_m:
0.2220 - val_recall_m: 0.8042
Epoch 4/50
10552/10552 [==============================] - 7s 708us/step - loss: 0.9525 -
accuracy: 0.7368 - f1_m: 0.3629 - precision_m: 0.2388 - recall_m: 0.7683 -
val_loss: 0.5675 - val_accuracy: 0.7126 - val_f1_m: 0.3506 - val_precision_m:
0.2254 - val_recall_m: 0.7992
Epoch 5/50
10552/10552 [==============================] - 7s 691us/step - loss: 0.9516 -
accuracy: 0.7374 - f1_m: 0.3635 - precision_m: 0.2393 - recall_m: 0.7682 -
val_loss: 0.5320 - val_accuracy: 0.7380 - val_f1_m: 0.3639 - val_precision_m:
0.2389 - val_recall_m: 0.7723
Epoch 6/50
10552/10552 [==============================] - 8s 712us/step - loss: 0.9503 -
accuracy: 0.7381 - f1_m: 0.3641 - precision_m: 0.2397 - recall_m: 0.7687 -
val_loss: 0.4964 - val_accuracy: 0.7508 - val_f1_m: 0.3708 - val_precision_m:
0.2465 - val_recall_m: 0.7567
Epoch 7/50
10552/10552 [==============================] - 8s 713us/step - loss: 0.9492 -
accuracy: 0.7378 - f1_m: 0.3641 - precision_m: 0.2396 - recall_m: 0.7698 -
val_loss: 0.5348 - val_accuracy: 0.7374 - val_f1_m: 0.3640 - val_precision_m:
0.2389 - val_recall_m: 0.7743
Epoch 8/50
10552/10552 [==============================] - 7s 711us/step - loss: 0.9483 -
accuracy: 0.7387 - f1_m: 0.3648 - precision_m: 0.2403 - recall_m: 0.7694 -
val_loss: 0.5187 - val_accuracy: 0.7422 - val_f1_m: 0.3670 - val_precision_m:
0.2418 - val_recall_m: 0.7700
Epoch 9/50
10552/10552 [==============================] - 8s 711us/step - loss: 0.9474 -
accuracy: 0.7392 - f1_m: 0.3654 - precision_m: 0.2407 - recall_m: 0.7699 -
val_loss: 0.5962 - val_accuracy: 0.6945 - val_f1_m: 0.3419 - val_precision_m:
0.2169 - val_recall_m: 0.8171
Epoch 10/50
10552/10552 [==============================] - 8s 726us/step - loss: 0.9466 -
accuracy: 0.7401 - f1_m: 0.3659 - precision_m: 0.2412 - recall_m: 0.7692 -
val_loss: 0.5462 - val_accuracy: 0.7319 - val_f1_m: 0.3617 - val_precision_m:
0.2361 - val_recall_m: 0.7826
Epoch 11/50
10552/10552 [==============================] - 8s 725us/step - loss: 0.9461 -
accuracy: 0.7403 - f1_m: 0.3661 - precision_m: 0.2414 - recall_m: 0.7691 -
val_loss: 0.5199 - val_accuracy: 0.7501 - val_f1_m: 0.3711 - val_precision_m:
0.2465 - val_recall_m: 0.7601
Epoch 12/50
10552/10552 [==============================] - 7s 710us/step - loss: 0.9455 -
accuracy: 0.7415 - f1_m: 0.3669 - precision_m: 0.2421 - recall_m: 0.7682 -
val_loss: 0.5182 - val_accuracy: 0.7511 - val_f1_m: 0.3720 - val_precision_m:
0.2473 - val_recall_m: 0.7599
```

```
Epoch 13/50
10552/10552 [==============================] - 8s 721us/step - loss: 0.9451 -
accuracy: 0.7417 - f1_m: 0.3671 - precision_m: 0.2423 - recall_m: 0.7682 -
val_loss: 0.4894 - val_accuracy: 0.7659 - val_f1_m: 0.3806 - val_precision_m:
0.2571 - val_recall_m: 0.7417
Epoch 14/50
10552/10552 [==============================] - 8s 716us/step - loss: 0.9446 -
accuracy: 0.7418 - f1_m: 0.3671 - precision_m: 0.2423 - recall_m: 0.7680 -
val_loss: 0.5330 - val_accuracy: 0.7304 - val_f1_m: 0.3610 - val_precision_m:
0.2353 - val_recall_m: 0.7846
Epoch 15/50
10552/10552 [==============================] - 8s 714us/step - loss: 0.9441 -
accuracy: 0.7417 - f1_m: 0.3671 - precision_m: 0.2423 - recall_m: 0.7683 -
val_loss: 0.5394 - val_accuracy: 0.7308 - val_f1_m: 0.3613 - val_precision_m:
0.2356 - val_recall_m: 0.7848
Epoch 16/50
10552/10552 [==============================] - 8s 715us/step - loss: 0.9438 -
accuracy: 0.7417 - f1_m: 0.3673 - precision_m: 0.2424 - recall_m: 0.7689 -
val_loss: 0.5201 - val_accuracy: 0.7494 - val_f1_m: 0.3712 - val_precision_m:
0.2463 - val_recall_m: 0.7624
Epoch 17/50
10552/10552 [==============================] - 8s 715us/step - loss: 0.9436 -
accuracy: 0.7416 - f1_m: 0.3672 - precision_m: 0.2423 - recall_m: 0.7690 -
val_loss: 0.5267 - val_accuracy: 0.7451 - val_f1_m: 0.3690 - val_precision_m:
0.2438 - val_recall_m: 0.7680
Epoch 18/50
10552/10552 [==============================] - 8s 720us/step - loss: 0.9433 -
accuracy: 0.7417 - f1_m: 0.3673 - precision_m: 0.2424 - recall_m: 0.7689 -
val_loss: 0.5488 - val_accuracy: 0.7258 - val_f1_m: 0.3585 - val_precision_m:
0.2328 - val_recall_m: 0.7892
Epoch 19/50
10552/10552 [==============================] - 8s 714us/step - loss: 0.9432 -
accuracy: 0.7418 - f1_m: 0.3674 - precision_m: 0.2424 - recall_m: 0.7691 -
val_loss: 0.5217 - val_accuracy: 0.7465 - val_f1_m: 0.3700 - val_precision_m:
0.2447 - val_recall_m: 0.7673
Epoch 20/50
10552/10552 [==============================] - 8s 717us/step - loss: 0.9430 -
accuracy: 0.7418 - f1_m: 0.3674 - precision_m: 0.2425 - recall_m: 0.7691 -
val_loss: 0.5202 - val_accuracy: 0.7442 - val_f1_m: 0.3684 - val_precision_m:
0.2432 - val_recall_m: 0.7688
Epoch 21/50
10552/10552 [==============================] - 8s 718us/step - loss: 0.9427 -
accuracy: 0.7421 - f1_m: 0.3677 - precision_m: 0.2427 - recall_m: 0.7693 -
val_loss: 0.4916 - val_accuracy: 0.7630 - val_f1_m: 0.3791 - val_precision_m:
0.2552 - val_recall_m: 0.7456
Epoch 22/50
10552/10552 [==============================] - 8s 716us/step - loss: 0.9426 -
accuracy: 0.7423 - f1_m: 0.3677 - precision_m: 0.2427 - recall_m: 0.7688 -
```

```
val_loss: 0.5052 - val_accuracy: 0.7529 - val_f1_m: 0.3735 - val_precision_m:
0.2486 - val_recall_m: 0.7591
Epoch 23/50
10552/10552 [==============================] - 8s 718us/step - loss: 0.9425 -
accuracy: 0.7422 - f1_m: 0.3677 - precision_m: 0.2427 - recall_m: 0.7690 -
val_loss: 0.5075 - val_accuracy: 0.7524 - val_f1_m: 0.3731 - val_precision_m:
0.2483 - val_recall_m: 0.7594
CPU times: user 3min 54s, sys: 55.7 s, total: 4min 49s
Wall time: 2min 56s
```

[ ]:
```
<keras.src.callbacks.History at 0x42d2b65d0>
```

[ ]:
```
eval_model=classifier.evaluate(X_train_sc, y_train)
print('loss: ', eval_model[0], 'and Accuracy: ', eval_model[1])
```

```
198625/198625 [==============================] - 57s 287us/step - loss: 0.5076 -
accuracy: 0.7521 - f1_m: 0.3546 - precision_m: 0.2485 - recall_m: 0.7298
loss:  0.5076255798339844 and Accuracy:  0.7521149516105652
```

[ ]:
```
print(history.history.keys())
```

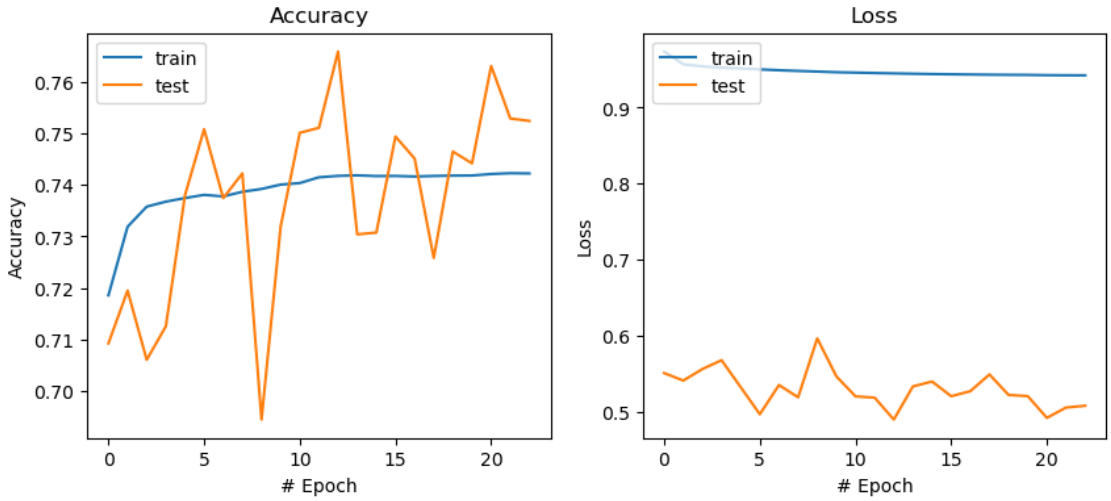
```
dict_keys(['loss', 'accuracy', 'f1_m', 'precision_m', 'recall_m', 'val_loss',
'val_accuracy', 'val_f1_m', 'val_precision_m', 'val_recall_m'])
```

[ ]:
```
fig, ax = plt.subplots(nrows = 1, ncols = 2, figsize = (10, 4))

# Plot accuracy
ax[0].plot(history.history['accuracy'])
ax[0].plot(history.history['val_accuracy'])
ax[0].set_ylabel('Accuracy')
ax[0].set_xlabel('# Epoch')
ax[0].legend(['train', 'test'], loc='upper left')
ax[0].set_title('Accuracy')

# Plot loss
ax[1].plot(history.history['loss'])
ax[1].plot(history.history['val_loss'])
ax[1].set_ylabel('Loss')
ax[1].set_xlabel('# Epoch')
ax[1].legend(['train', 'test'], loc='upper left')
ax[1].set_title('Loss')
```

[ ]:
```
Text(0.5, 1.0, 'Loss')
```



[ ]:
```
for key, value in classifier.__dict__.items():
    print(key, ":", value)
    print()
```

```
_self_setattr_tracking : True

_obj_reference_counts_dict : ObjectIdentityDictionary({<_ObjectIdentityWrapper
wrapping True>: 3, <_ObjectIdentityWrapper wrapping
<keras.src.saving.serialization_lib.Config object at 0x42d2b6b90>>: 1,
<_ObjectIdentityWrapper wrapping <keras.src.saving.serialization_lib.Config
object at 0x42d421910>>: 1, <_ObjectIdentityWrapper wrapping
<keras.src.optimizers.legacy.adam.Adam object at 0x42d2c0150>>: 1,
<_ObjectIdentityWrapper wrapping <keras.src.engine.compile_utils.LossesContainer
object at 0x137efcf50>>: 1, <_ObjectIdentityWrapper wrapping
<keras.src.engine.compile_utils.MetricsContainer object at 0x42910ca50>>: 1,
<_ObjectIdentityWrapper wrapping 0>: 1, <_ObjectIdentityWrapper wrapping
'binary_crossentropy'>: 1, <_ObjectIdentityWrapper wrapping
<tensorflow.python.eager.polymorphic_function.polymorphic_function.Function
object at 0x42d2e1fd0>>: 2, <_ObjectIdentityWrapper wrapping
<tensorflow.python.eager.polymorphic_function.polymorphic_function.Function
object at 0x42dfd2610>>: 1, <_ObjectIdentityWrapper wrapping
<keras.src.callbacks.History object at 0x15e934f10>>: 1, <_ObjectIdentityWrapper
wrapping
<tensorflow.python.eager.polymorphic_function.polymorphic_function.Function
object at 0x42d72a650>>: 1})

_auto_get_config : True

_auto_config : <keras.src.saving.serialization_lib.Config object at 0x42d2b6b90>
```

```
_is_model_for_instrumentation : True

_instrumented_keras_api : True

_instrumented_keras_layer_class : False

_instrumented_keras_model_class : True

_trainable : True

_stateful : False

built : True

_input_spec : None

_build_input_shape : (None, 67)

_saved_model_inputs_spec : TensorSpec(shape=(None, 67), dtype=tf.float32,
name='dense_input')

_saved_model_arg_spec : ([TensorSpec(shape=(None, 67), dtype=tf.float32,
name='dense_input')], {})

_supports_masking : True

_name : sequential

_activity_regularizer : None

_trainable_weights : []

_non_trainable_weights : []

_updates : []

_thread_local : <_thread._local object at 0x42d4138d0>

_callable_losses : []

_losses : []

_metrics : []

_metrics_lock : <unlocked _thread.lock object at 0x42d2e25c0>

_dtype_policy : <Policy "float32">
```

```
_compute_dtype_object : <dtype: 'float32'>

_autocast : False

_self_tracked_trackables : [<keras.src.engine.input_layer.InputLayer object at
0x42d2c0a50>, <keras.src.layers.core.dense.Dense object at 0x12f0967d0>,
<keras.src.layers.core.dense.Dense object at 0x12dcfe650>,
<keras.src.layers.core.dense.Dense object at 0x42d422090>,
<keras.src.layers.core.dense.Dense object at 0x42d421d90>]

_inbound_nodes_value : []

_outbound_nodes_value : []

_call_spec : <keras.src.utils.layer_utils.CallFunctionSpec object at
0x42917a090>

_dynamic : False

_initial_weights : None

_auto_track_sub_layers : False

_preserve_input_structure_in_config : False

_name_scope_on_declaration :

_captured_weight_regularizer : []

_is_graph_network : True

inputs : [<KerasTensor: shape=(None, 67) dtype=float32 (created by layer
'dense_input')>]

outputs : [<KerasTensor: shape=(None, 1) dtype=float32 (created by layer
'dense_3')>]

input_names : ['dense_input']

output_names : ['dense_3']

_compute_output_and_mask_jointly : True

_distribution_strategy : None

_distribute_reduction_method : None

_cluster_coordinator : None
```

```
_compiled_trainable_state : <WeakKeyDictionary at 0x42d2e27d0>

_training_state : None

_self_unconditional_checkpoint_dependencies :
[TrackableReference(name=optimizer, ref=<keras.src.optimizers.legacy.adam.Adam
object at 0x42d2c0150>), TrackableReference(name=train_tf_function,
ref=<tensorflow.python.eager.polymorphic_function.polymorphic_function.Function
object at 0x42d2e1fd0>), TrackableReference(name=train_function,
ref=<tensorflow.python.eager.polymorphic_function.polymorphic_function.Function
object at 0x42d2e1fd0>), TrackableReference(name=test_function,
ref=<tensorflow.python.eager.polymorphic_function.polymorphic_function.Function
object at 0x42dfd2610>), TrackableReference(name=predict_function,
ref=<tensorflow.python.eager.polymorphic_function.polymorphic_function.Function
object at 0x42d72a650>)]

_self_unconditional_dependency_names : {'optimizer':
<keras.src.optimizers.legacy.adam.Adam object at 0x42d2c0150>,
'train_tf_function':
<tensorflow.python.eager.polymorphic_function.polymorphic_function.Function
object at 0x42d2e1fd0>, 'train_function':
<tensorflow.python.eager.polymorphic_function.polymorphic_function.Function
object at 0x42d2e1fd0>, 'test_function':
<tensorflow.python.eager.polymorphic_function.polymorphic_function.Function
object at 0x42dfd2610>, 'predict_function':
<tensorflow.python.eager.polymorphic_function.polymorphic_function.Function
object at 0x42d72a650>}

_self_unconditional_deferred_dependencies : {}

_self_update_uid : -1

_self_name_based_restores : set()

_self_saveable_object_factories : {}

_checkpoint : <tensorflow.python.checkpoint.checkpoint.Checkpoint object at
0x137f27fd0>

_steps_per_execution : <tf.Variable 'Variable:0' shape=() dtype=int64, numpy=1>

_steps_per_execution_tuner : None

_autotune_steps_per_execution : False

_layout_map : None
```

```
_train_counter : <tf.Variable 'Variable:0' shape=() dtype=int64, numpy=242696>

_test_counter : <tf.Variable 'Variable:0' shape=() dtype=int64, numpy=198625>

_predict_counter : <tf.Variable 'Variable:0' shape=() dtype=int64, numpy=66209>

_base_model_initialized : True

_inferred_input_shape : None

_has_explicit_input_shape : True

_input_dtype : None

_layer_call_argspecs : {<keras.src.engine.input_layer.InputLayer object at
0x42d2c0a50>: FullArgSpec(args=['self', 'inputs'], varargs='args',
varkw='kwargs', defaults=None, kwonlyargs=[], kwonlydefaults=None,
annotations={}), <keras.src.layers.core.dense.Dense object at 0x12f0967d0>:
FullArgSpec(args=['self', 'inputs'], varargs=None, varkw=None, defaults=None,
kwonlyargs=[], kwonlydefaults=None, annotations={}),
<keras.src.layers.core.dense.Dense object at 0x12dcfe650>:
FullArgSpec(args=['self', 'inputs'], varargs=None, varkw=None, defaults=None,
kwonlyargs=[], kwonlydefaults=None, annotations={}),
<keras.src.layers.core.dense.Dense object at 0x42d422090>:
FullArgSpec(args=['self', 'inputs'], varargs=None, varkw=None, defaults=None,
kwonlyargs=[], kwonlydefaults=None, annotations={}),
<keras.src.layers.core.dense.Dense object at 0x42d421d90>:
FullArgSpec(args=['self', 'inputs'], varargs=None, varkw=None, defaults=None,
kwonlyargs=[], kwonlydefaults=None, annotations={})}

_created_nodes : set()

_graph_initialized : True

_use_legacy_deferred_behavior : False

_nested_inputs : KerasTensor(type_spec=TensorSpec(shape=(None, 67),
dtype=tf.float32, name='dense_input'), name='dense_input', description="created
by layer 'dense_input'")

_nested_outputs : KerasTensor(type_spec=TensorSpec(shape=(None, 1),
dtype=tf.float32, name=None), name='dense_3/Sigmoid:0', description="created by
layer 'dense_3'")

_enable_dict_to_input_mapping : True

_input_layers : [<keras.src.engine.input_layer.InputLayer object at
0x42d2c0a50>]
```

```
_output_layers : [<keras.src.layers.core.dense.Dense object at 0x42d421d90>]

_input_coordinates : [(<keras.src.engine.input_layer.InputLayer object at
0x42d2c0a50>, 0, 0)]

_output_coordinates : [(<keras.src.layers.core.dense.Dense object at
0x42d421d90>, 0, 0)]

_output_mask_cache : {}

_output_tensor_cache : {}

_output_shape_cache : {}

_network_nodes : {'dense_2_ib-0', 'dense_ib-0', 'dense_1_ib-0',
'dense_input_ib-0', 'dense_3_ib-0'}

_nodes_by_depth : defaultdict(<class 'list'>, {0: [<keras.src.engine.node.Node
object at 0x42d2c3850>], 1: [<keras.src.engine.node.Node object at
0x4172fdb50>], 2: [<keras.src.engine.node.Node object at 0x14a4aeb50>], 3:
[<keras.src.engine.node.Node object at 0x12f575f50>], 4:
[<keras.src.engine.node.Node object at 0x137f26ad0>]})

_feed_input_names : ['dense_input']

_feed_inputs : [<KerasTensor: shape=(None, 67) dtype=float32 (created by layer
'dense_input')>]

_feed_input_shapes : [(None, 67)]

_tensor_usage_count : Counter({'5063518288': 1, '17937694672': 1, '17939637584':
1, '17939518608': 1, '17939796496': 1})

_compile_config : <keras.src.saving.serialization_lib.Config object at
0x42d421910>

_run_eagerly : None

optimizer : <keras.src.optimizers.legacy.adam.Adam object at 0x42d2c0150>

compiled_loss : <keras.src.engine.compile_utils.LossesContainer object at
0x137efcf50>

compiled_metrics : <keras.src.engine.compile_utils.MetricsContainer object at
0x42910ca50>

_pss_evaluation_shards : 0
```

```
_is_compiled : True

loss : binary_crossentropy

_jit_compile : None

train_tf_function :
<tensorflow.python.eager.polymorphic_function.polymorphic_function.Function
object at 0x42d2e1fd0>

train_function :
<tensorflow.python.eager.polymorphic_function.polymorphic_function.Function
object at 0x42d2e1fd0>

stop_training : True

test_function :
<tensorflow.python.eager.polymorphic_function.polymorphic_function.Function
object at 0x42dfd2610>

history : <keras.src.callbacks.History object at 0x15e934f10>

predict_function :
<tensorflow.python.eager.polymorphic_function.polymorphic_function.Function
object at 0x42d72a650>
```

[ ]:
```
probabilities = classifier.predict(X_test_sc)
# predictions = classifier.predict_classes(X_test_sc)
predictions = np.argmax(classifier.predict(X_test_sc), axis=-1)

print ("\n Classification report : \n",classification_report(y_test,␣
 ↪predictions))
print ("Accuracy   Score : ",accuracy_score(y_test, predictions))

#confusion matrix
conf_matrix = confusion_matrix(y_test,predictions)
plt.figure(figsize=(12,12))
plt.subplot(221)
sns.heatmap(conf_matrix, fmt = "d",annot=True, cmap='Blues')
b, t = plt.ylim()
plt.ylim(b + 0.5, t - 0.5)
plt.title('Confuion Matrix')
plt.ylabel('True Values')
plt.xlabel('Predicted Values')
```

```python
#f1-score
f1 = f1_score(y_test, predictions)
print("F1 Score: ", f1)

#roc_auc_score
model_roc_auc = roc_auc_score(y_test,probabilities)
print ("Area under curve : ",model_roc_auc,"\n")
fpr,tpr,thresholds = roc_curve(y_test,probabilities)
gmeans = np.sqrt(tpr * (1-fpr))
ix = np.argmax(gmeans)
threshold = np.round(thresholds[ix],3)

plt.subplot(222)
plt.plot(fpr, tpr, color='darkorange', lw=1, label = "Auc : %.3f"␣
↪%model_roc_auc)
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.scatter(fpr[ix], tpr[ix], marker='o', color='black', label='Best Threshold:
↪' + str(threshold))
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic')
plt.legend(loc="lower right")

plt.show()
```

```
66209/66209 [==============================] - 15s 220us/step
66209/66209 [==============================] - 15s 222us/step
```

```
/Users/aadityakasbekar/anaconda3/lib/python3.11/site-
packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning:
Precision and F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/Users/aadityakasbekar/anaconda3/lib/python3.11/site-
packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning:
Precision and F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/Users/aadityakasbekar/anaconda3/lib/python3.11/site-
packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning:
Precision and F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

```
 Classification report :
```

```
              precision    recall  f1-score   support

         0.0       0.90      1.00      0.95   1911460
         1.0       0.00      0.00      0.00    207206

    accuracy                           0.90   2118666
   macro avg       0.45      0.50      0.47   2118666
weighted avg       0.81      0.90      0.86   2118666

Accuracy   Score :  0.902199780427873
F1 Score:  0.0
Area under curve :  0.8345176414666227
```