```
!pip install numpy
!pip install tensorflow --upgrade
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (1.22.4)
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: tensorflow in /usr/local/lib/python3.10/dist-packages (2.12.0)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=2.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (23.3.3)
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.4.0)
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.54.0)
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (3.8.0)
Requirement already satisfied: jax>=0.3.15 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.4.8)
Requirement already satisfied: keras<2.13,>=2.12.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.12.0)
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (16.0.0)
Requirement already satisfied: numpy<1.24,>=1.22 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.22.4)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (3.3.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from tensorflow) (23.1)
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 /usr/local/lib/pythor
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from tensorflow) (67.7.2)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.16.0)
Requirement already satisfied: tensorboard<2.13,>=2.12 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.12.2)
Requirement already satisfied: tensorflow-estimator<2.13,>=2.12.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.12.0)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.3.0)
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (4.5.0)
Requirement already satisfied: wrapt<1.15,>=1.11.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.14.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.32.(
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.10/dist-packages (from astunparse>=1.6.0->tensorflow) (0.40.
Requirement already satisfied: ml-dtypes>=0.0.3 in /usr/local/lib/python3.10/dist-packages (from jax>=0.3.15->tensorflow) (0.1.0)
Requirement already satisfied: scipy>=1.7 in /usr/local/lib/python3.10/dist-packages (from jax>=0.3.15->tensorflow) (1.10.1)
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.13,>=2.12->tensorflc
Requirement already satisfied: google-auth-oauthlib<1.1,>=0.5 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.13,>=2.12->
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.13,>=2.12->tensorflow) (3.
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.13,>=2.12->tensorflow)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.13,>
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.13,>=2.12->t
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.13,>=2.12->tensorflow) (2.
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3->tensorboar
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3->tensorboarc
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3->tensorboard<2.13,>=
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.10/dist-packages (from google-auth-oauthlib<1.1,>=0.5-
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorboard<2
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorboard<2.13
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorboa
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorboard<2.13,>=2.1
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.10/dist-packages (from werkzeug>=1.0.1->tensorboard<2.13,>=2.
Requirement already satisfied: pyasn1<0.6.0,>=0.4.6 in /usr/local/lib/python3.10/dist-packages (from pyasn1-modules>=0.2.1->google-auth<
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.10/dist-packages (from requests-oauthlib>=0.7.0->google-auth-oa
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.model_selection import train_test_split
import tensorflow as tf


from zipfile import ZipFile



with ZipFile("//content/voice.csv .zip") as zipobj:
  zipobj.extractall("/content/sample_data/abc")


data = pd.read_csv('/content/voice.csv .zip')


data
```

	meanfreq	sd	median	Q25	Q75	IQR	skew	kurt	sp.ent	sfm	...	centroid	mea
0	0.059781	0.064241	0.032027	0.015071	0.090193	0.075122	12.863462	274.402906	0.893369	0.491918	...	0.059781	0.08
1	0.066009	0.067310	0.040229	0.019414	0.092666	0.073252	22.423285	634.613855	0.892193	0.513724	...	0.066009	0.10
2	0.077316	0.083829	0.036718	0.008701	0.131908	0.123207	30.757155	1024.927705	0.846389	0.478905	...	0.077316	0.09
3	0.151228	0.072111	0.158011	0.096582	0.207955	0.111374	1.232831	4.177296	0.963322	0.727232	...	0.151228	0.08
4	0.135120	0.079146	0.124656	0.078720	0.206045	0.127325	1.101174	4.333713	0.971955	0.783568	...	0.135120	0.10
...	...	...	...	...	...	...	...	...	...	...	...	...	
3163	0.131884	0.084734	0.153707	0.049285	0.201144	0.151859	1.762129	6.630383	0.962934	0.763182	...	0.131884	0.18
3164	0.116221	0.089221	0.076758	0.042718	0.204911	0.162193	0.693730	2.503954	0.960716	0.709570	...	0.116221	0.18
3165	0.142056	0.095798	0.183731	0.033424	0.224360	0.190936	1.876502	6.604509	0.946854	0.654196	...	0.142056	0.20
3166	0.143659	0.090628	0.184976	0.043508	0.219943	0.176435	1.591065	5.388298	0.950436	0.675470	...	0.143659	0.17

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3168 entries, 0 to 3167
Data columns (total 21 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   meanfreq  3168 non-null   float64
 1   sd        3168 non-null   float64
 2   median    3168 non-null   float64
 3   Q25       3168 non-null   float64
 4   Q75       3168 non-null   float64
 5   IQR       3168 non-null   float64
 6   skew      3168 non-null   float64
 7   kurt      3168 non-null   float64
 8   sp.ent    3168 non-null   float64
 9   sfm       3168 non-null   float64
 10  mode      3168 non-null   float64
 11  centroid  3168 non-null   float64
 12  meanfun   3168 non-null   float64
 13  minfun    3168 non-null   float64
 14  maxfun    3168 non-null   float64
 15  meandom   3168 non-null   float64
 16  mindom    3168 non-null   float64
 17  maxdom    3168 non-null   float64
 18  dfrange   3168 non-null   float64
 19  modindx   3168 non-null   float64
 20  label     3168 non-null   object 
dtypes: float64(20), object(1)
memory usage: 519.9+ KB
```

```
label_encoder = LabelEncoder()
```

```
data['label'] = label_encoder.fit_transform(data['label'])
```

```
dict(enumerate(label_encoder.classes_))
```

```
{0: 'female', 1: 'male'}
```

```
data
```

	meanfreq	sd	median	Q25	Q75	IQR	skew	kurt	sp.ent	sfm	...	centroid	mea
0	0.059781	0.064241	0.032027	0.015071	0.090193	0.075122	12.863462	274.402906	0.893369	0.491918	...	0.059781	0.08
1	0.066009	0.067310	0.040229	0.019414	0.092666	0.073252	22.423285	634.613855	0.892193	0.513724	...	0.066009	0.10
2	0.077316	0.083829	0.036718	0.008701	0.131908	0.123207	30.757155	1024.927705	0.846389	0.478905	...	0.077316	0.09
3	0.151228	0.072111	0.158011	0.096582	0.207955	0.111374	1.232831	4.177296	0.963322	0.727232		0.151228	0.08

```
y = data['label'].copy()
X = data.drop('label', axis=1).copy()
```

...	...	...	...	...	...	...	...	...	...	...	...	...

```
scaler = StandardScaler()

X = scaler.fit_transform(X)
```

3165	0.142056	0.095798	0.183731	0.033424	0.224360	0.190936	1.876502	6.604509	0.946854	0.654196	...	0.142056	0.20

```
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7, random_state=42)


X.shape
```

```
(3168, 20)
```

```
inputs = tf.keras.Input(shape=(X.shape[1],))

x = tf.keras.layers.Dense(64, activation='relu')(inputs)
x = tf.keras.layers.Dense(64, activation='relu')(x)

outputs = tf.keras.layers.Dense(1, activation='sigmoid')(x)

model = tf.keras.Model(inputs, outputs)


model.summary()
```

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 20)]              0

 dense (Dense)               (None, 64)                1344

 dense_1 (Dense)             (None, 64)                4160

 dense_2 (Dense)             (None, 1)                 65

=================================================================
Total params: 5,569
Trainable params: 5,569
Non-trainable params: 0
_________________________________________________________________
```

```
model.compile(
    optimizer='adam',
    loss='binary_crossentropy',
    metrics=[
        'accuracy',
        tf.keras.metrics.AUC(name='auc')
    ]
)

history = model.fit(
    X_train,
    y_train,
    validation_split=0.2,
    batch_size=32,
    epochs=100,
    callbacks=[
        tf.keras.callbacks.EarlyStopping(
            monitor='val_loss',
            patience=3,
            restore_best_weights=True
        )
    ]
)
```

```
Epoch 1/100
56/56 [==============================] - 1s 6ms/step - loss: 0.4351 - accuracy: 0.8156 - auc: 0.9013 - val_loss: 0.2285 - val_accuracy:
Epoch 2/100
56/56 [==============================] - 0s 2ms/step - loss: 0.1685 - accuracy: 0.9487 - auc: 0.9889 - val_loss: 0.1049 - val_accuracy:
Epoch 3/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0959 - accuracy: 0.9695 - auc: 0.9954 - val_loss: 0.0685 - val_accuracy:
Epoch 4/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0752 - accuracy: 0.9752 - auc: 0.9962 - val_loss: 0.0628 - val_accuracy:
Epoch 5/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0674 - accuracy: 0.9769 - auc: 0.9971 - val_loss: 0.0569 - val_accuracy:
Epoch 6/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0609 - accuracy: 0.9769 - auc: 0.9974 - val_loss: 0.0512 - val_accuracy:
Epoch 7/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0557 - accuracy: 0.9803 - auc: 0.9979 - val_loss: 0.0517 - val_accuracy:
Epoch 8/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0502 - accuracy: 0.9842 - auc: 0.9983 - val_loss: 0.0452 - val_accuracy:
Epoch 9/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0497 - accuracy: 0.9825 - auc: 0.9982 - val_loss: 0.0423 - val_accuracy:
Epoch 10/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0465 - accuracy: 0.9848 - auc: 0.9985 - val_loss: 0.0478 - val_accuracy:
Epoch 11/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0420 - accuracy: 0.9876 - auc: 0.9987 - val_loss: 0.0405 - val_accuracy:
Epoch 12/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0418 - accuracy: 0.9853 - auc: 0.9988 - val_loss: 0.0455 - val_accuracy:
Epoch 13/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0392 - accuracy: 0.9876 - auc: 0.9989 - val_loss: 0.0502 - val_accuracy:
Epoch 14/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0367 - accuracy: 0.9870 - auc: 0.9990 - val_loss: 0.0485 - val_accuracy:
```

```
model.evaluate(X_test, y_test)
```

```
30/30 [==============================] - 0s 1ms/step - loss: 0.0619 - accuracy: 0.9811 - auc: 0.9975
[0.061910390853881836, 0.9810725450515747, 0.997472882270813]
```

```
X = tf.keras.preprocessing.sequence.pad_sequences(X, dtype=float, maxlen=25, padding='post')

X = X.reshape(-1, 5, 5)
X = np.expand_dims(X, axis=3)
```

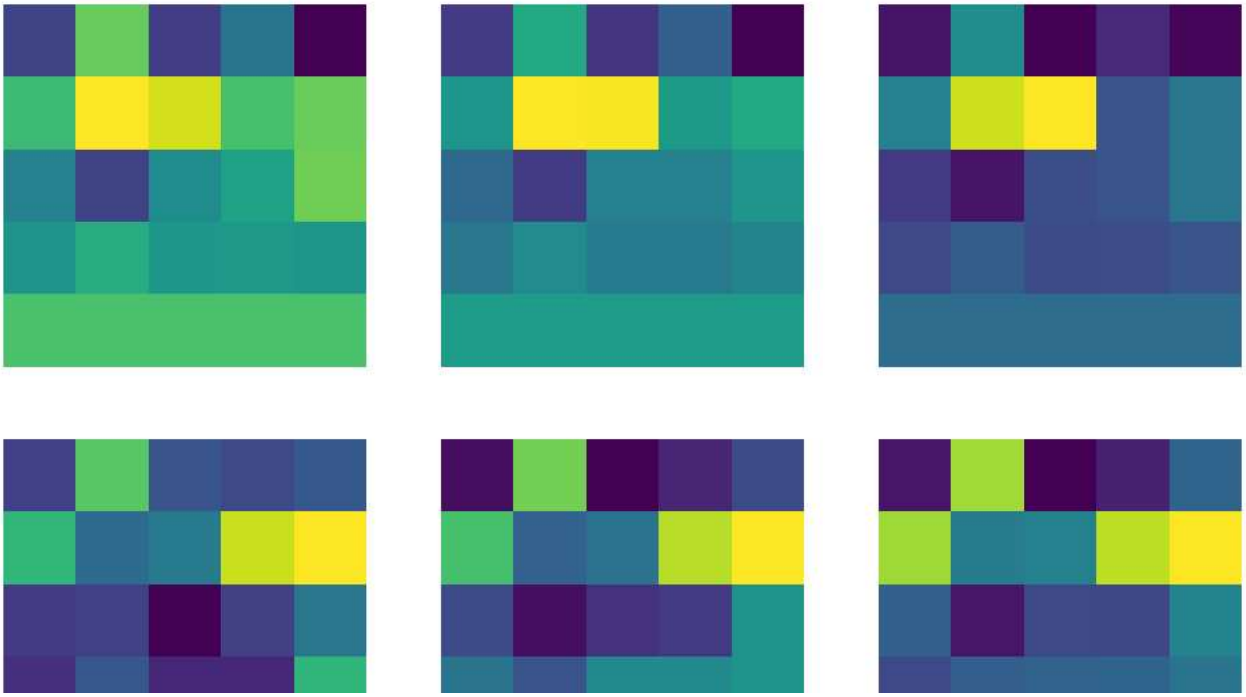
```
X.shape
```

```
(3168, 5, 5, 1)
```

```
plt.figure(figsize=(12, 12))

for i in range(9):
    plt.subplot(3, 3, i + 1)
    plt.imshow(np.squeeze(X[i]))
    plt.axis('off')

plt.show()
```



```
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7, random_state=42)
```

```
inputs = tf.keras.Input(shape=(X.shape[1], X.shape[2], X.shape[3]))

x = tf.keras.layers.Conv2D(16, 2, activation='relu')(inputs)
x = tf.keras.layers.MaxPooling2D()(x)

x = tf.keras.layers.Conv2D(32, 1, activation='relu')(x)
x = tf.keras.layers.MaxPooling2D()(x)

x = tf.keras.layers.Flatten()(x)

x = tf.keras.layers.Dense(64, activation='relu')(x)

outputs = tf.keras.layers.Dense(1, activation='sigmoid')(x)

model = tf.keras.Model(inputs, outputs)
```

```
model.summary()
```

```
Model: "model_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 5, 5, 1)]         0

 conv2d (Conv2D)             (None, 4, 4, 16)          80

 max_pooling2d (MaxPooling2D  (None, 2, 2, 16)         0
 )

 conv2d_1 (Conv2D)           (None, 2, 2, 32)          544

 max_pooling2d_1 (MaxPooling  (None, 1, 1, 32)         0
 2D)

 flatten (Flatten)           (None, 32)                0

 dense_3 (Dense)             (None, 64)                2112

 dense_4 (Dense)             (None, 1)                 65

=================================================================
Total params: 2,801
Trainable params: 2,801
Non-trainable params: 0
_________________________________________________________________
```

```
model.compile(
```

```
    optimizer='adam',
    loss='binary_crossentropy',
    metrics=[
        'accuracy',
        tf.keras.metrics.AUC(name='auc')
    ]
)

history = model.fit(
    X_train,
    y_train,
    validation_split=0.2,
    batch_size=32,
    epochs=100
)
```

```
56/56 [==============================] - 0s 2ms/step - loss: 0.0425 - accuracy: 0.9865 - auc: 0.9990 - val_loss: 0.0827 - val_accura
Epoch 41/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0411 - accuracy: 0.9882 - auc: 0.9991 - val_loss: 0.0955 - val_accura
Epoch 42/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0446 - accuracy: 0.9865 - auc: 0.9989 - val_loss: 0.0994 - val_accura
Epoch 43/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0403 - accuracy: 0.9876 - auc: 0.9991 - val_loss: 0.0871 - val_accura
Epoch 44/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0386 - accuracy: 0.9870 - auc: 0.9992 - val_loss: 0.0932 - val_accura
Epoch 45/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0390 - accuracy: 0.9870 - auc: 0.9992 - val_loss: 0.0872 - val_accura
Epoch 46/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0358 - accuracy: 0.9887 - auc: 0.9994 - val_loss: 0.0875 - val_accura
Epoch 47/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0349 - accuracy: 0.9893 - auc: 0.9994 - val_loss: 0.0943 - val_accura
Epoch 48/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0357 - accuracy: 0.9898 - auc: 0.9993 - val_loss: 0.0907 - val_accura
Epoch 49/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0317 - accuracy: 0.9904 - auc: 0.9995 - val_loss: 0.0932 - val_accura
Epoch 50/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0292 - accuracy: 0.9927 - auc: 0.9997 - val_loss: 0.0924 - val_accura
Epoch 51/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0301 - accuracy: 0.9893 - auc: 0.9996 - val_loss: 0.0964 - val_accura
Epoch 52/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0308 - accuracy: 0.9910 - auc: 0.9996 - val_loss: 0.0993 - val_accura
Epoch 53/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0309 - accuracy: 0.9898 - auc: 0.9995 - val_loss: 0.0991 - val_accura
Epoch 54/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0301 - accuracy: 0.9921 - auc: 0.9995 - val_loss: 0.1197 - val_accura
Epoch 55/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0331 - accuracy: 0.9876 - auc: 0.9994 - val_loss: 0.0930 - val_accura
Epoch 56/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0260 - accuracy: 0.9915 - auc: 0.9997 - val_loss: 0.1126 - val_accura
Epoch 57/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0323 - accuracy: 0.9882 - auc: 0.9994 - val_loss: 0.0941 - val_accura
Epoch 58/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0310 - accuracy: 0.9882 - auc: 0.9995 - val_loss: 0.1115 - val_accura
Epoch 59/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0261 - accuracy: 0.9904 - auc: 0.9997 - val_loss: 0.1014 - val_accura
Epoch 60/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0229 - accuracy: 0.9932 - auc: 0.9998 - val_loss: 0.1006 - val_accura
Epoch 61/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0289 - accuracy: 0.9904 - auc: 0.9996 - val_loss: 0.1009 - val_accura
Epoch 62/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0497 - accuracy: 0.9791 - auc: 0.9985 - val_loss: 0.1020 - val_accura
Epoch 63/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0279 - accuracy: 0.9904 - auc: 0.9996 - val_loss: 0.1125 - val_accura
Epoch 64/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0212 - accuracy: 0.9921 - auc: 0.9998 - val_loss: 0.1097 - val_accura
Epoch 65/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0202 - accuracy: 0.9932 - auc: 0.9998 - val_loss: 0.1103 - val_accura
Epoch 66/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0198 - accuracy: 0.9938 - auc: 0.9999 - val_loss: 0.1070 - val_accura
Epoch 67/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0181 - accuracy: 0.9966 - auc: 0.9999 - val_loss: 0.1088 - val_accura
Epoch 68/100
56/56 [==============================] - 0s 2ms/step - loss: 0.0173 - accuracy: 0.9961 - auc: 0.9999 - val_loss: 0.1037 - val_accura
```

```
model.evaluate(X_test, y_test)
```

```
30/30 [==============================] - 0s 1ms/step - loss: 0.1843 - accuracy: 0.9558 - auc: 0.9841
[0.1842736154794693, 0.9558359384536743, 0.9841187000274658]
```

✓ 0s completed at 6:47 AM