fication-raisindataset-17-11-24-1

November 24, 2024

1 Binary Classification on Raisin Dataset

In this project, we aim to build a binary classification system to predict the target class of a dataset using machine learning (ML) and deep learning (DL) techniques. The classification models will be optimized through random search hyperparameter tuning and evaluated using 10-fold cross-validation to ensure robust performance. The results will include detailed evaluation metrics and visualizations (e.g., ROC curves) for model comparison.

1.0.1 Install All Necessary Packages

[1]:
```
!pip install pandas
!pip install numpy
!pip install matplotlib
!pip install seaborn
!pip install scikit-learn
!pip install tensorflow
!pip install keras-models
!pip install scikeras
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages
(2.2.2)
Requirement already satisfied: numpy>=1.22.4 in /usr/local/lib/python3.10/dist-
packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-
packages (from pandas) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-
packages (from pandas) (2024.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-
packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages
(1.26.4)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-
packages (3.8.0)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (1.3.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-
```

```
packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (4.55.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.7)
Requirement already satisfied: numpy<2,>=1.21 in /usr/local/lib/python3.10/dist-
packages (from matplotlib) (1.26.4)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (24.2)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-
packages (from matplotlib) (11.0.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (3.2.0)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-
packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Requirement already satisfied: seaborn in /usr/local/lib/python3.10/dist-
packages (0.13.2)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in
/usr/local/lib/python3.10/dist-packages (from seaborn) (1.26.4)
Requirement already satisfied: pandas>=1.2 in /usr/local/lib/python3.10/dist-
packages (from seaborn) (2.2.2)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in
/usr/local/lib/python3.10/dist-packages (from seaborn) (3.8.0)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn)
(1.3.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-
packages (from matplotlib!=3.6.1,>=3.4->seaborn) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn)
(4.55.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn)
(1.4.7)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn)
(24.2)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-
packages (from matplotlib!=3.6.1,>=3.4->seaborn) (11.0.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn)
(3.2.0)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn)
(2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-
```

```
packages (from pandas>=1.2->seaborn) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-
packages (from pandas>=1.2->seaborn) (2024.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-
packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.16.0)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-
packages (1.5.2)
Requirement already satisfied: numpy>=1.19.5 in /usr/local/lib/python3.10/dist-
packages (from scikit-learn) (1.26.4)
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.10/dist-
packages (from scikit-learn) (1.13.1)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.10/dist-
packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in
/usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.5.0)
Requirement already satisfied: tensorflow in /usr/local/lib/python3.10/dist-
packages (2.17.1)
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>=24.3.25 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (24.3.25)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.6.0)
Requirement already satisfied: google-pasta>=0.1.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: h5py>=3.10.0 in /usr/local/lib/python3.10/dist-
packages (from tensorflow) (3.12.1)
Requirement already satisfied: libclang>=13.0.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (18.1.1)
Requirement already satisfied: ml-dtypes<0.5.0,>=0.3.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.4.1)
Requirement already satisfied: opt-einsum>=2.3.2 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (3.4.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-
packages (from tensorflow) (24.2)
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 /usr/local/lib/python3.10/dist-packages (from tensorflow) (4.25.5)
Requirement already satisfied: requests<3,>=2.21.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (2.32.3)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-
packages (from tensorflow) (75.1.0)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.10/dist-
packages (from tensorflow) (1.16.0)
Requirement already satisfied: termcolor>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (2.5.0)
```

```
Requirement already satisfied: typing-extensions>=3.6.6 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (4.12.2)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.10/dist-
packages (from tensorflow) (1.16.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.68.0)
Requirement already satisfied: tensorboard<2.18,>=2.17 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (2.17.1)
Requirement already satisfied: keras>=3.2.0 in /usr/local/lib/python3.10/dist-
packages (from tensorflow) (3.5.0)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.37.1)
Requirement already satisfied: numpy<2.0.0,>=1.23.5 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.26.4)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
/usr/local/lib/python3.10/dist-packages (from astunparse>=1.6.0->tensorflow)
(0.45.0)
Requirement already satisfied: rich in /usr/local/lib/python3.10/dist-packages
(from keras>=3.2.0->tensorflow) (13.9.4)
Requirement already satisfied: namex in /usr/local/lib/python3.10/dist-packages
(from keras>=3.2.0->tensorflow) (0.0.8)
Requirement already satisfied: optree in /usr/local/lib/python3.10/dist-packages
(from keras>=3.2.0->tensorflow) (0.13.1)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorflow)
(3.4.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests<3,>=2.21.0->tensorflow) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorflow)
(2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorflow)
(2024.8.30)
Requirement already satisfied: markdown>=2.6.8 in
/usr/local/lib/python3.10/dist-packages (from
tensorboard<2.18,>=2.17->tensorflow) (3.7)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in
/usr/local/lib/python3.10/dist-packages (from
tensorboard<2.18,>=2.17->tensorflow) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from
tensorboard<2.18,>=2.17->tensorflow) (3.1.3)
Requirement already satisfied: MarkupSafe>=2.1.1 in
/usr/local/lib/python3.10/dist-packages (from
werkzeug>=1.0.1->tensorboard<2.18,>=2.17->tensorflow) (3.0.2)
Requirement already satisfied: markdown-it-py>=2.2.0 in
/usr/local/lib/python3.10/dist-packages (from rich->keras>=3.2.0->tensorflow)
```

```
(3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
/usr/local/lib/python3.10/dist-packages (from rich->keras>=3.2.0->tensorflow)
(2.18.0)
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-
packages (from markdown-it-py>=2.2.0->rich->keras>=3.2.0->tensorflow) (0.1.2)
Collecting keras-models
  Downloading keras_models-0.0.7-py3-none-any.whl.metadata (3.4 kB)
Requirement already satisfied: keras in /usr/local/lib/python3.10/dist-packages
(from keras-models) (3.5.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages
(from keras-models) (1.26.4)
Requirement already satisfied: spacy in /usr/local/lib/python3.10/dist-packages
(from keras-models) (3.7.5)
Requirement already satisfied: Pillow in /usr/local/lib/python3.10/dist-packages
(from keras-models) (11.0.0)
Requirement already satisfied: opencv-python in /usr/local/lib/python3.10/dist-
packages (from keras-models) (4.10.0.84)
Requirement already satisfied: pathlib in /usr/local/lib/python3.10/dist-
packages (from keras-models) (1.0.1)
Requirement already satisfied: absl-py in /usr/local/lib/python3.10/dist-
packages (from keras->keras-models) (1.4.0)
Requirement already satisfied: rich in /usr/local/lib/python3.10/dist-packages
(from keras->keras-models) (13.9.4)
Requirement already satisfied: namex in /usr/local/lib/python3.10/dist-packages
(from keras->keras-models) (0.0.8)
Requirement already satisfied: h5py in /usr/local/lib/python3.10/dist-packages
(from keras->keras-models) (3.12.1)
Requirement already satisfied: optree in /usr/local/lib/python3.10/dist-packages
(from keras->keras-models) (0.13.1)
Requirement already satisfied: ml-dtypes in /usr/local/lib/python3.10/dist-
packages (from keras->keras-models) (0.4.1)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-
packages (from keras->keras-models) (24.2)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in
/usr/local/lib/python3.10/dist-packages (from spacy->keras-models) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in
/usr/local/lib/python3.10/dist-packages (from spacy->keras-models) (1.0.5)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in
/usr/local/lib/python3.10/dist-packages (from spacy->keras-models) (1.0.10)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in
/usr/local/lib/python3.10/dist-packages (from spacy->keras-models) (2.0.8)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in
/usr/local/lib/python3.10/dist-packages (from spacy->keras-models) (3.0.9)
Requirement already satisfied: thinc<8.3.0,>=8.2.2 in
/usr/local/lib/python3.10/dist-packages (from spacy->keras-models) (8.2.5)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in
/usr/local/lib/python3.10/dist-packages (from spacy->keras-models) (1.1.3)
```

```
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in
/usr/local/lib/python3.10/dist-packages (from spacy->keras-models) (2.4.8)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in
/usr/local/lib/python3.10/dist-packages (from spacy->keras-models) (2.0.10)
Requirement already satisfied: weasel<0.5.0,>=0.1.0 in
/usr/local/lib/python3.10/dist-packages (from spacy->keras-models) (0.4.1)
Requirement already satisfied: typer<1.0.0,>=0.3.0 in
/usr/local/lib/python3.10/dist-packages (from spacy->keras-models) (0.13.0)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in
/usr/local/lib/python3.10/dist-packages (from spacy->keras-models) (4.66.6)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in
/usr/local/lib/python3.10/dist-packages (from spacy->keras-models) (2.32.3)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in
/usr/local/lib/python3.10/dist-packages (from spacy->keras-models) (2.9.2)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages
(from spacy->keras-models) (3.1.4)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-
packages (from spacy->keras-models) (75.1.0)
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in
/usr/local/lib/python3.10/dist-packages (from spacy->keras-models) (3.4.1)
Requirement already satisfied: language-data>=1.2 in
/usr/local/lib/python3.10/dist-packages (from
langcodes<4.0.0,>=3.2.0->spacy->keras-models) (1.2.0)
Requirement already satisfied: annotated-types>=0.6.0 in
/usr/local/lib/python3.10/dist-packages (from
pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy->keras-models) (0.7.0)
Requirement already satisfied: pydantic-core==2.23.4 in
/usr/local/lib/python3.10/dist-packages (from
pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy->keras-models) (2.23.4)
Requirement already satisfied: typing-extensions>=4.6.1 in
/usr/local/lib/python3.10/dist-packages (from
pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy->keras-models) (4.12.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from
requests<3.0.0,>=2.13.0->spacy->keras-models) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests<3.0.0,>=2.13.0->spacy->keras-models) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from
requests<3.0.0,>=2.13.0->spacy->keras-models) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from
requests<3.0.0,>=2.13.0->spacy->keras-models) (2024.8.30)
Requirement already satisfied: blis<0.8.0,>=0.7.8 in
/usr/local/lib/python3.10/dist-packages (from thinc<8.3.0,>=8.2.2->spacy->keras-
models) (0.7.11)
Requirement already satisfied: confection<1.0.0,>=0.0.1 in
/usr/local/lib/python3.10/dist-packages (from thinc<8.3.0,>=8.2.2->spacy->keras-
```

```
models) (0.1.5)
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.10/dist-
packages (from typer<1.0.0,>=0.3.0->spacy->keras-models) (8.1.7)
Requirement already satisfied: shellingham>=1.3.0 in
/usr/local/lib/python3.10/dist-packages (from typer<1.0.0,>=0.3.0->spacy->keras-
models) (1.5.4)
Requirement already satisfied: markdown-it-py>=2.2.0 in
/usr/local/lib/python3.10/dist-packages (from rich->keras->keras-models) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
/usr/local/lib/python3.10/dist-packages (from rich->keras->keras-models)
(2.18.0)
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in
/usr/local/lib/python3.10/dist-packages (from
weasel<0.5.0,>=0.1.0->spacy->keras-models) (0.20.0)
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in
/usr/local/lib/python3.10/dist-packages (from
weasel<0.5.0,>=0.1.0->spacy->keras-models) (7.0.5)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2->spacy->keras-models)
(3.0.2)
Requirement already satisfied: marisa-trie>=0.7.7 in
/usr/local/lib/python3.10/dist-packages (from language-
data>=1.2->langcodes<4.0.0,>=3.2.0->spacy->keras-models) (1.2.1)
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-
packages (from markdown-it-py>=2.2.0->rich->keras->keras-models) (0.1.2)
Requirement already satisfied: wrapt in /usr/local/lib/python3.10/dist-packages
(from smart-open<8.0.0,>=5.2.1->weasel<0.5.0,>=0.1.0->spacy->keras-models)
(1.16.0)
Downloading keras_models-0.0.7-py3-none-any.whl (18 kB)
Installing collected packages: keras-models
Successfully installed keras-models-0.0.7
Collecting scikeras
  Downloading scikeras-0.13.0-py3-none-any.whl.metadata (3.1 kB)
Requirement already satisfied: keras>=3.2.0 in /usr/local/lib/python3.10/dist-
packages (from scikeras) (3.5.0)
Requirement already satisfied: scikit-learn>=1.4.2 in
/usr/local/lib/python3.10/dist-packages (from scikeras) (1.5.2)
Requirement already satisfied: absl-py in /usr/local/lib/python3.10/dist-
packages (from keras>=3.2.0->scikeras) (1.4.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages
(from keras>=3.2.0->scikeras) (1.26.4)
Requirement already satisfied: rich in /usr/local/lib/python3.10/dist-packages
(from keras>=3.2.0->scikeras) (13.9.4)
Requirement already satisfied: namex in /usr/local/lib/python3.10/dist-packages
(from keras>=3.2.0->scikeras) (0.0.8)
Requirement already satisfied: h5py in /usr/local/lib/python3.10/dist-packages
(from keras>=3.2.0->scikeras) (3.12.1)
Requirement already satisfied: optree in /usr/local/lib/python3.10/dist-packages
```

```
(from keras>=3.2.0->scikeras) (0.13.1)
Requirement already satisfied: ml-dtypes in /usr/local/lib/python3.10/dist-
packages (from keras>=3.2.0->scikeras) (0.4.1)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-
packages (from keras>=3.2.0->scikeras) (24.2)
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.10/dist-
packages (from scikit-learn>=1.4.2->scikeras) (1.13.1)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.10/dist-
packages (from scikit-learn>=1.4.2->scikeras) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in
/usr/local/lib/python3.10/dist-packages (from scikit-learn>=1.4.2->scikeras)
(3.5.0)
Requirement already satisfied: typing-extensions>=4.5.0 in
/usr/local/lib/python3.10/dist-packages (from optree->keras>=3.2.0->scikeras)
(4.12.2)
Requirement already satisfied: markdown-it-py>=2.2.0 in
/usr/local/lib/python3.10/dist-packages (from rich->keras>=3.2.0->scikeras)
(3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
/usr/local/lib/python3.10/dist-packages (from rich->keras>=3.2.0->scikeras)
(2.18.0)
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-
packages (from markdown-it-py>=2.2.0->rich->keras>=3.2.0->scikeras) (0.1.2)
Downloading scikeras-0.13.0-py3-none-any.whl (26 kB)
Installing collected packages: scikeras
Successfully installed scikeras-0.13.0
```

1.0.2 Import all packages & librabries

[2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split, StratifiedKFold,␣
 ↪RandomizedSearchCV
from sklearn.metrics import (
    confusion_matrix, accuracy_score, precision_score, recall_score, f1_score,
    roc_auc_score, log_loss, matthews_corrcoef, roc_curve, auc
)
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import StandardScaler

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, LSTM
from tensorflow.keras.optimizers import Adam
from scikeras.wrappers import KerasClassifier, KerasRegressor
```

```
from tensorflow.keras.utils import to_categorical
```

1.0.3 Load Dataset

Dataset link - https://www.kaggle.com/datasets/nimapourmoradi/raisin-binary-classification

[3]:
```
# Load Dataset
data = pd.read_csv("raisin_dataset.csv")
data.head()
```

[3]:

	Area	MajorAxisLength	MinorAxisLength	Eccentricity	ConvexArea	\
0	87524	442.246011	253.291155	0.819738	90546	
1	75166	406.690687	243.032436	0.801805	78789	
2	90856	442.267048	266.328318	0.798354	93717	
3	45928	286.540559	208.760042	0.684989	47336	
4	79408	352.190770	290.827533	0.564011	81463	

	Extent	Perimeter	Class
0	0.758651	1184.040	Kecimen
1	0.684130	1121.786	Kecimen
2	0.637613	1208.575	Kecimen
3	0.699599	844.162	Kecimen
4	0.792772	1073.251	Kecimen

Basic Information of Dataset

[4]:
```
# Display Shape and Info
print("Dataset Shape:", data.shape)
print("\nDataset Info:")
print(data.info())
```

```
Dataset Shape: (900, 8)

Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 900 entries, 0 to 899
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Area             900 non-null    int64
 1   MajorAxisLength  900 non-null    float64
 2   MinorAxisLength  900 non-null    float64
 3   Eccentricity     900 non-null    float64
 4   ConvexArea       900 non-null    int64
 5   Extent           900 non-null    float64
 6   Perimeter        900 non-null    float64
 7   Class            900 non-null    object
dtypes: float64(5), int64(2), object(1)
```

```
memory usage: 56.4+ KB
None
```

1.0.4 EDA

Remove null values & duplicate values

[5]:
```
# Basic EDA
print("\nNull Values:\n", data.isnull().sum())
data.drop_duplicates(inplace=True)
print("\nAfter Removing Duplicates - Shape:", data.shape)
```

```
Null Values:
 Area               0
MajorAxisLength    0
MinorAxisLength    0
Eccentricity       0
ConvexArea         0
Extent             0
Perimeter          0
Class              0
dtype: int64

After Removing Duplicates - Shape: (900, 8)
```

Describe Dataset

[6]:
```
# Describe Dataset
print("\nDataset Description:")
data.describe()
```

```
Dataset Description:
```

[6]:

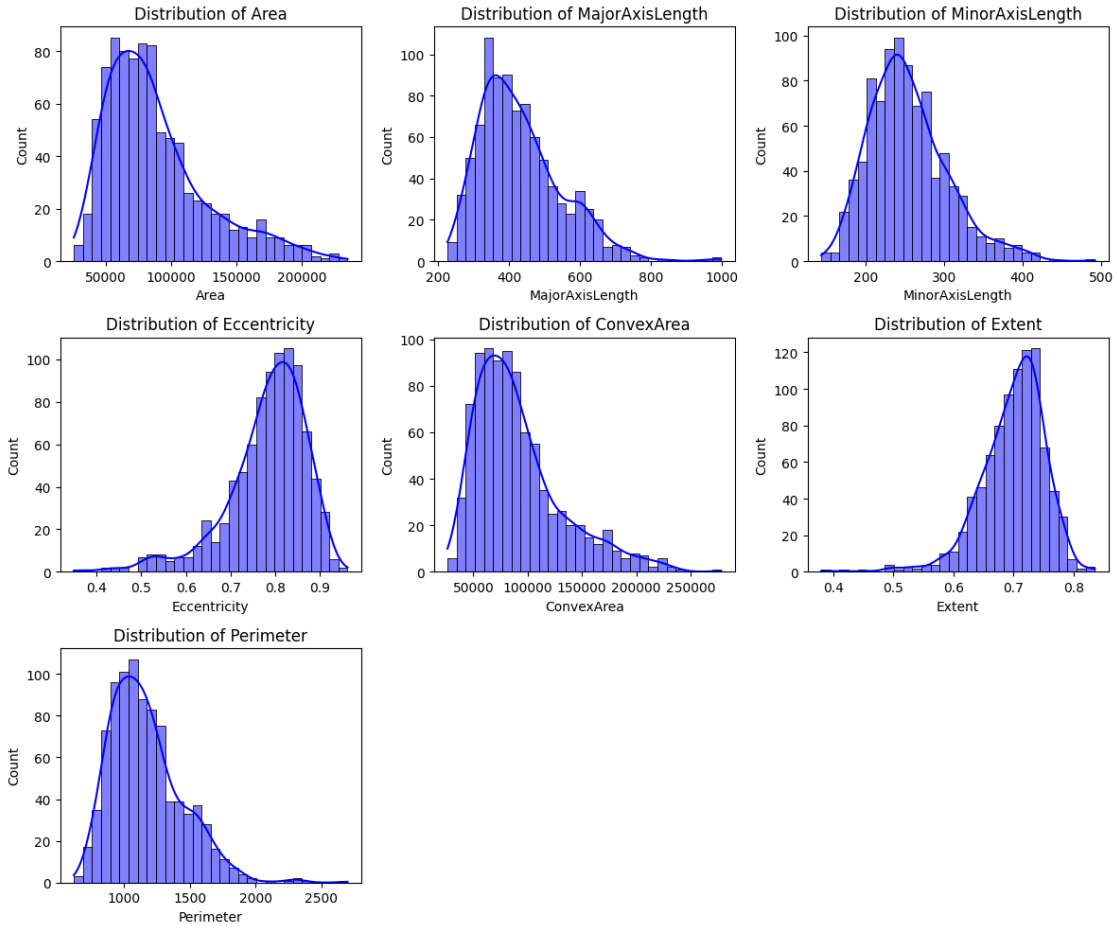
	Area	MajorAxisLength	MinorAxisLength	Eccentricity	\
count	900.000000	900.000000	900.000000	900.000000	
mean	87804.127778	430.929950	254.488133	0.781542	
std	39002.111390	116.035121	49.988902	0.090318	
min	25387.000000	225.629541	143.710872	0.348730	
25%	59348.000000	345.442898	219.111126	0.741766	
50%	78902.000000	407.803951	247.848409	0.798846	
75%	105028.250000	494.187014	279.888575	0.842571	
max	235047.000000	997.291941	492.275279	0.962124	

	ConvexArea	Extent	Perimeter
count	900.000000	900.000000	900.000000
mean	91186.090000	0.699508	1165.906636
std	40769.290132	0.053468	273.764315

min	26139.000000	0.379856	619.074000
25%	61513.250000	0.670869	966.410750
50%	81651.000000	0.707367	1119.509000
75%	108375.750000	0.734991	1308.389750
max	278217.000000	0.835455	2697.753000

Histogram Shows the distribution of each feature across the dataset.

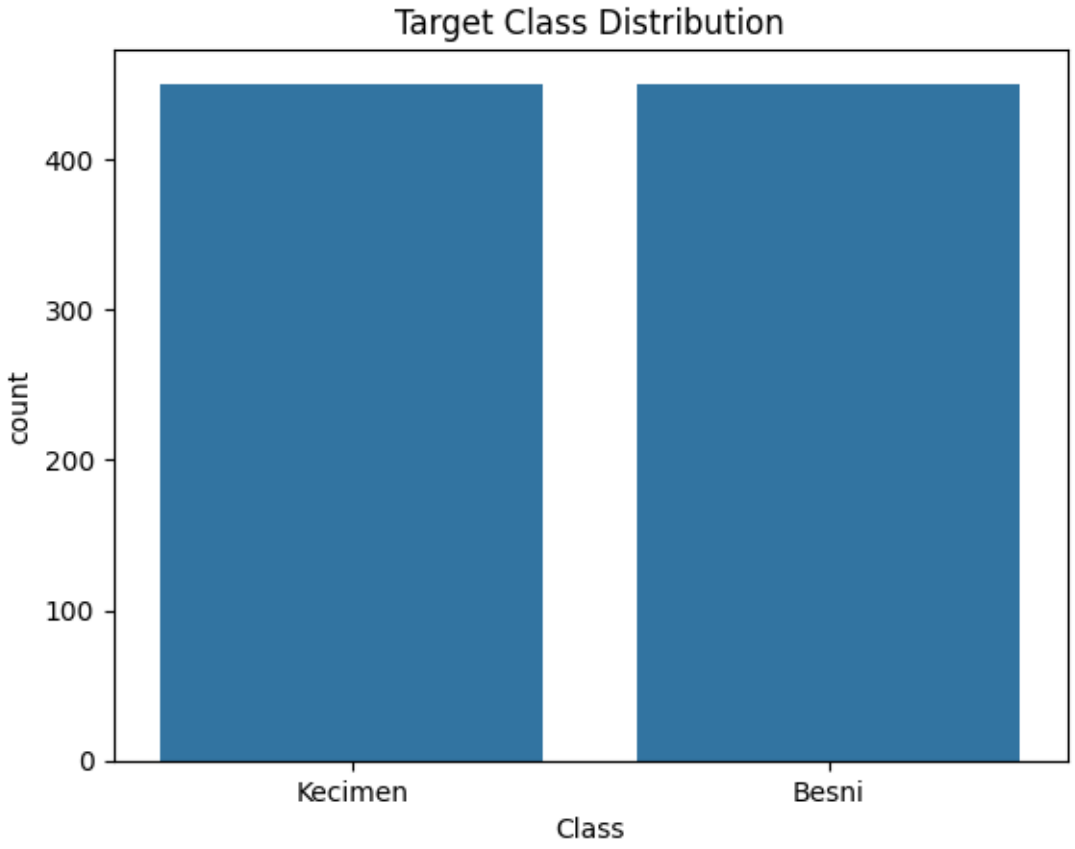
[7]:
```
# Histograms for Feature Distribution
plt.figure(figsize=(12, 10))
for i, column in enumerate(data.columns[:-1], start=1):  # Exclude Outcome
    plt.subplot(3, 3, i)
    sns.histplot(data[column], kde=True, color='blue', bins=30)
    plt.title(f'Distribution of {column}')
plt.tight_layout()
plt.show()
```



Count Plot - Target Class Distribution Displays the frequency distribution of each class, providing insight into class imbalance.

[8]:
```
# Check class distribution
sns.countplot(x='Class', data=data)
plt.title("Target Class Distribution")
plt.show()

# Encode the target column ('Class')
data['Class'] = data['Class'].map({'Kecimen': 0, 'Besni': 1})
```

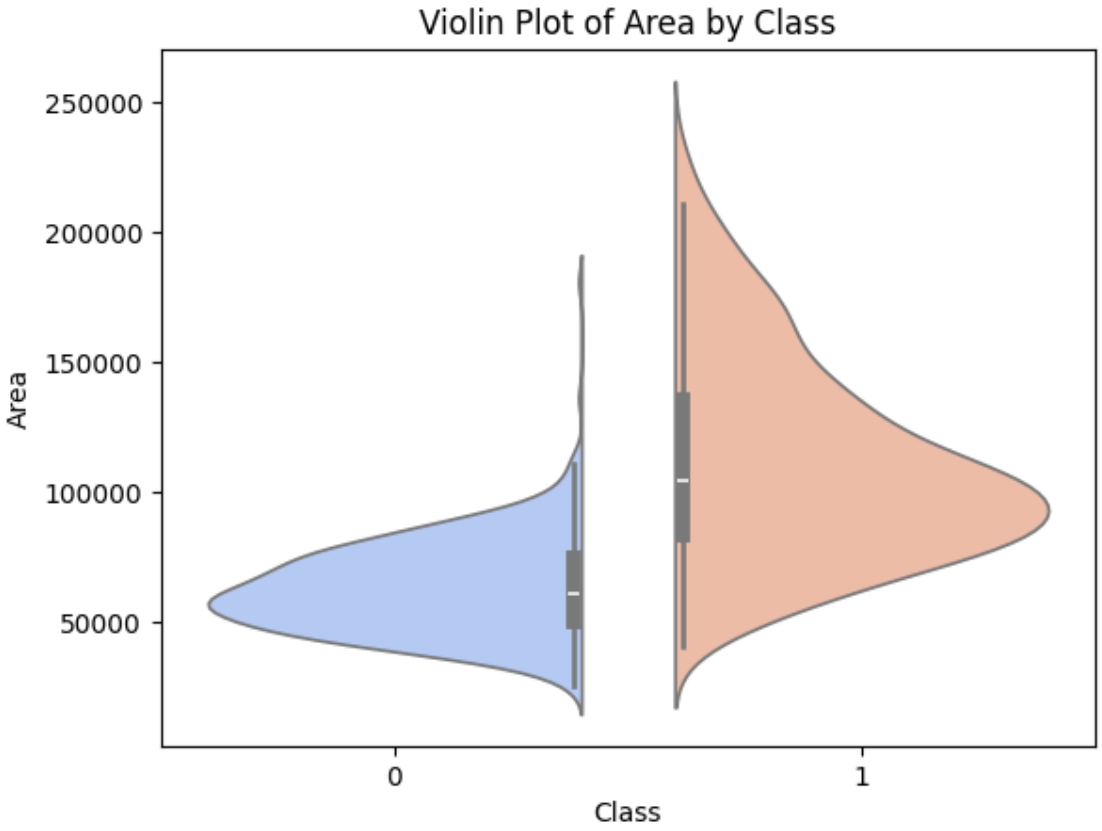


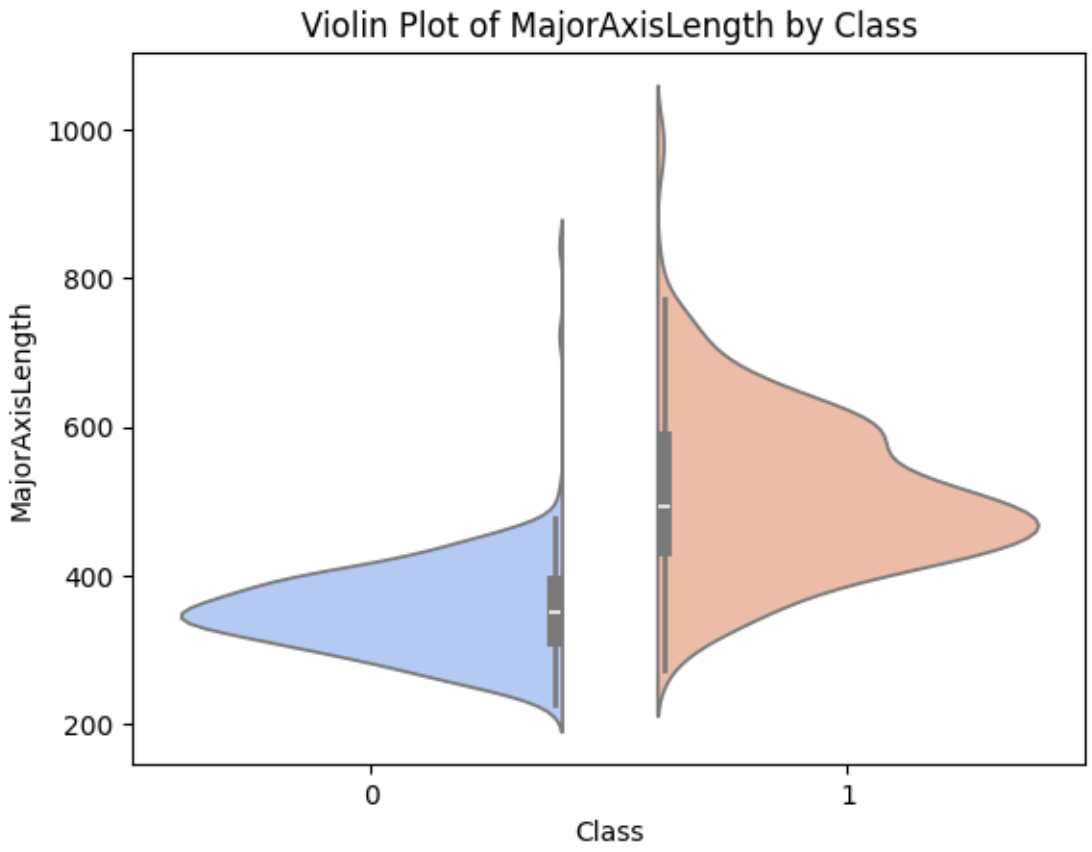
Violin Plot Combines box plots with kernel density estimation, providing more insight into the data distribution.

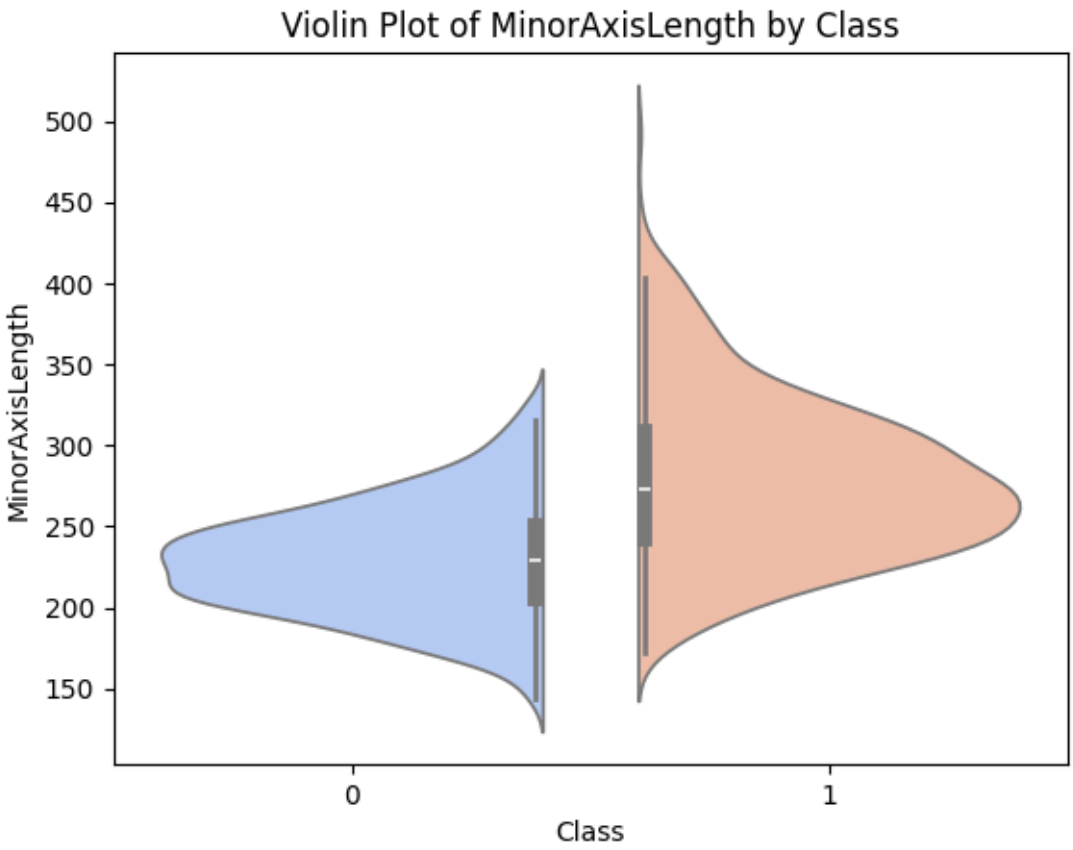
[9]:
```
import warnings
warnings.filterwarnings('ignore')
```

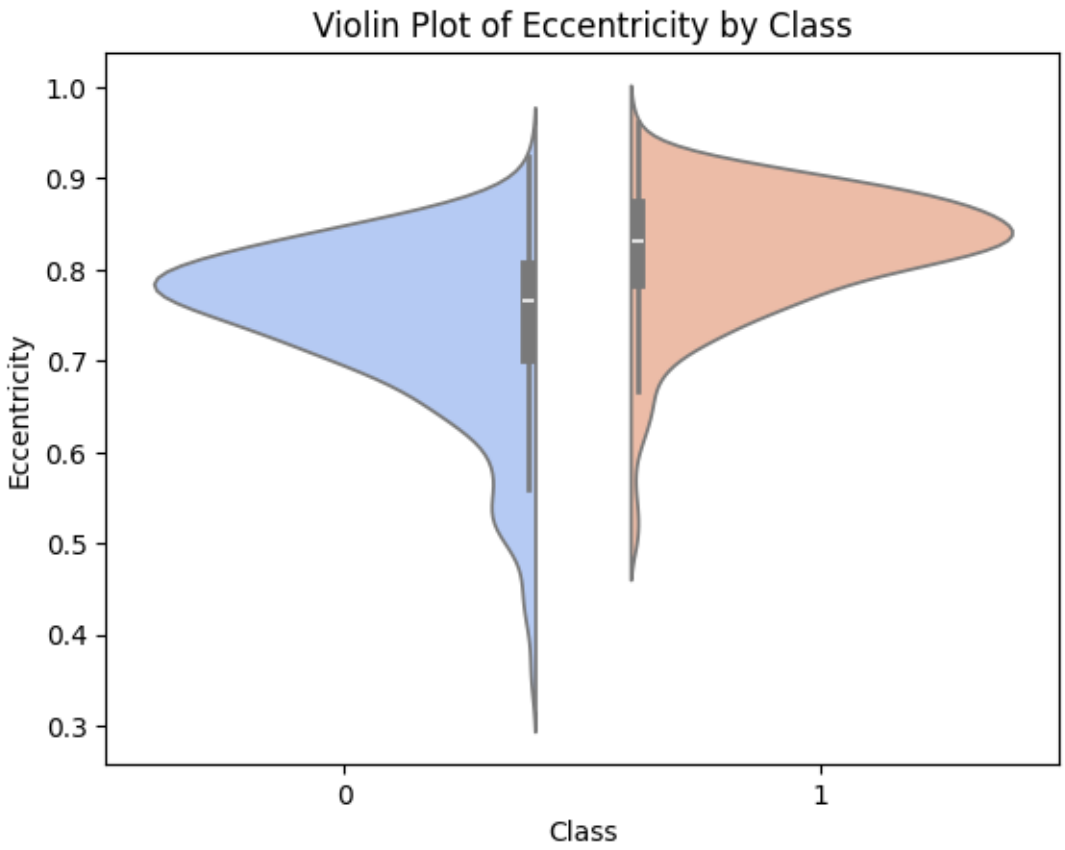
[10]:
```
for column in data.columns[:-1]:
    sns.violinplot(data=data, x="Class", y=column, palette="coolwarm",␣
 ↪split=True)
```

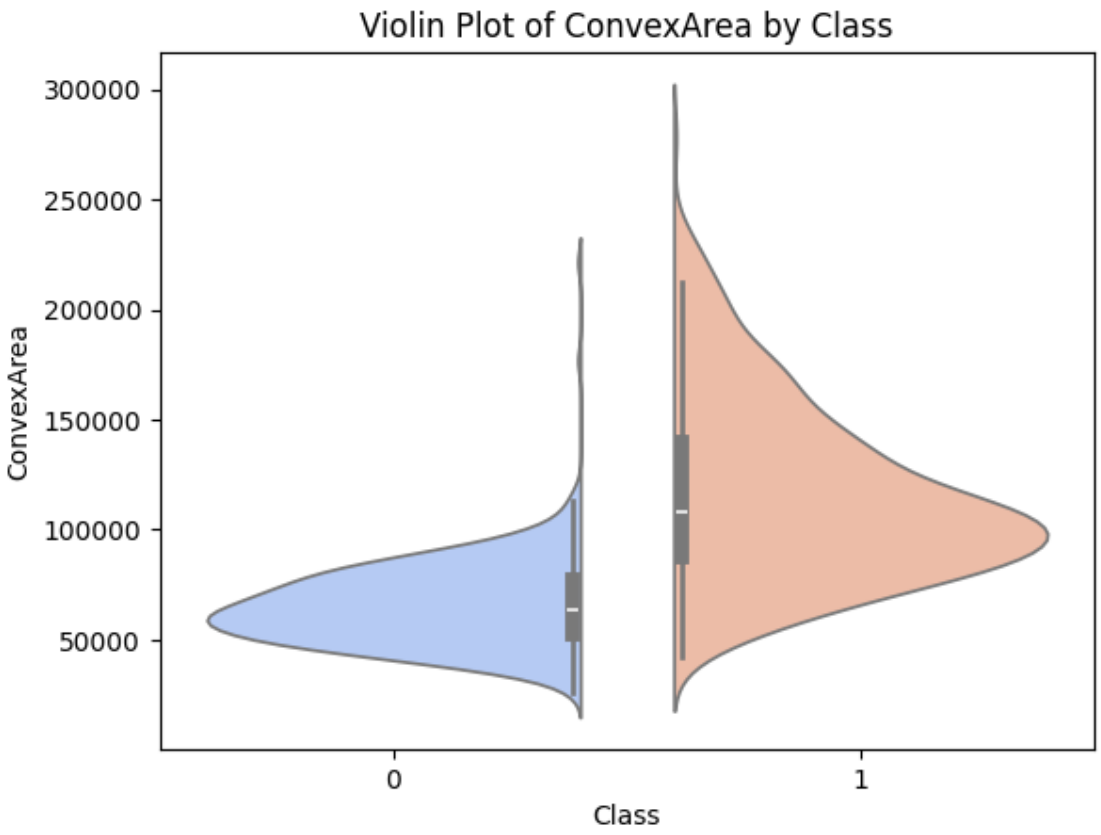
```
    plt.title(f"Violin Plot of {column} by Class")
    plt.show()
```



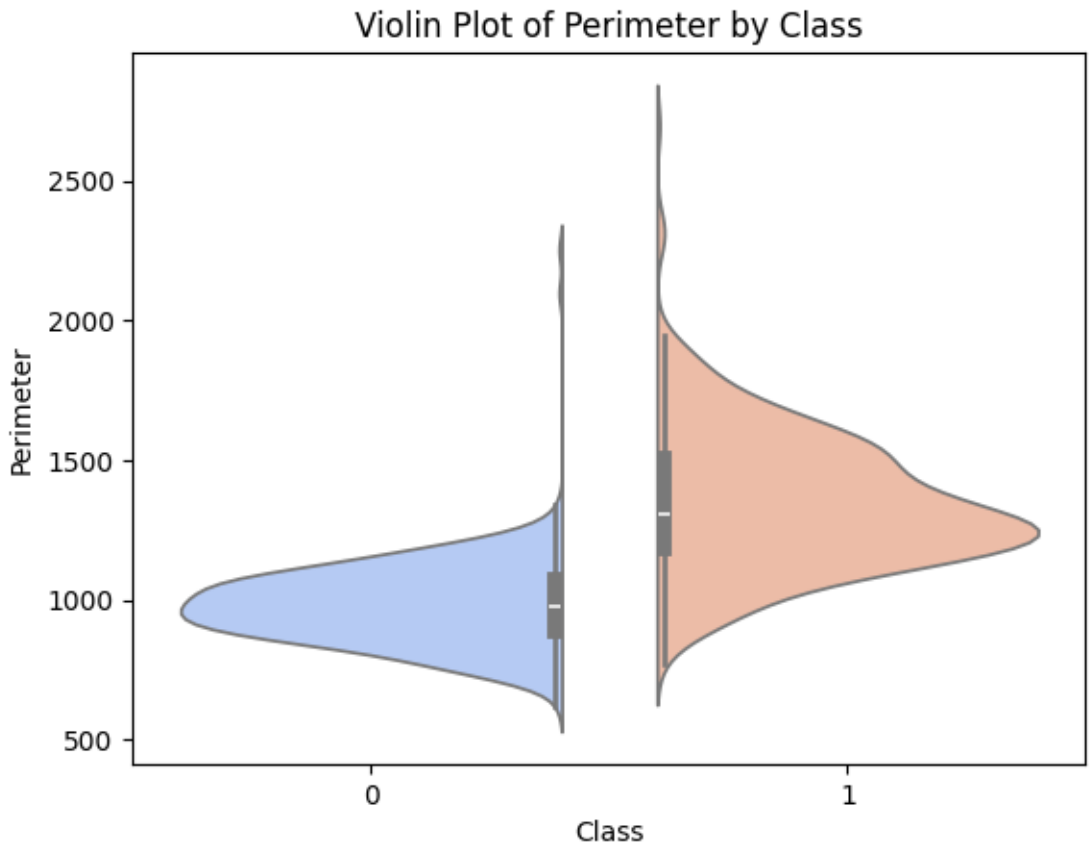

Violin Plot of MajorAxisLength by Class
MajorAxisLength
1000
800
600
400
200
0
1
Class


Violin Plot of MinorAxisLength by Class
MinorAxisLength
500
450
400
350
300
250
200
150
0
1
Class


Violin Plot of Eccentricity by Class
Eccentricity
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
1
Class

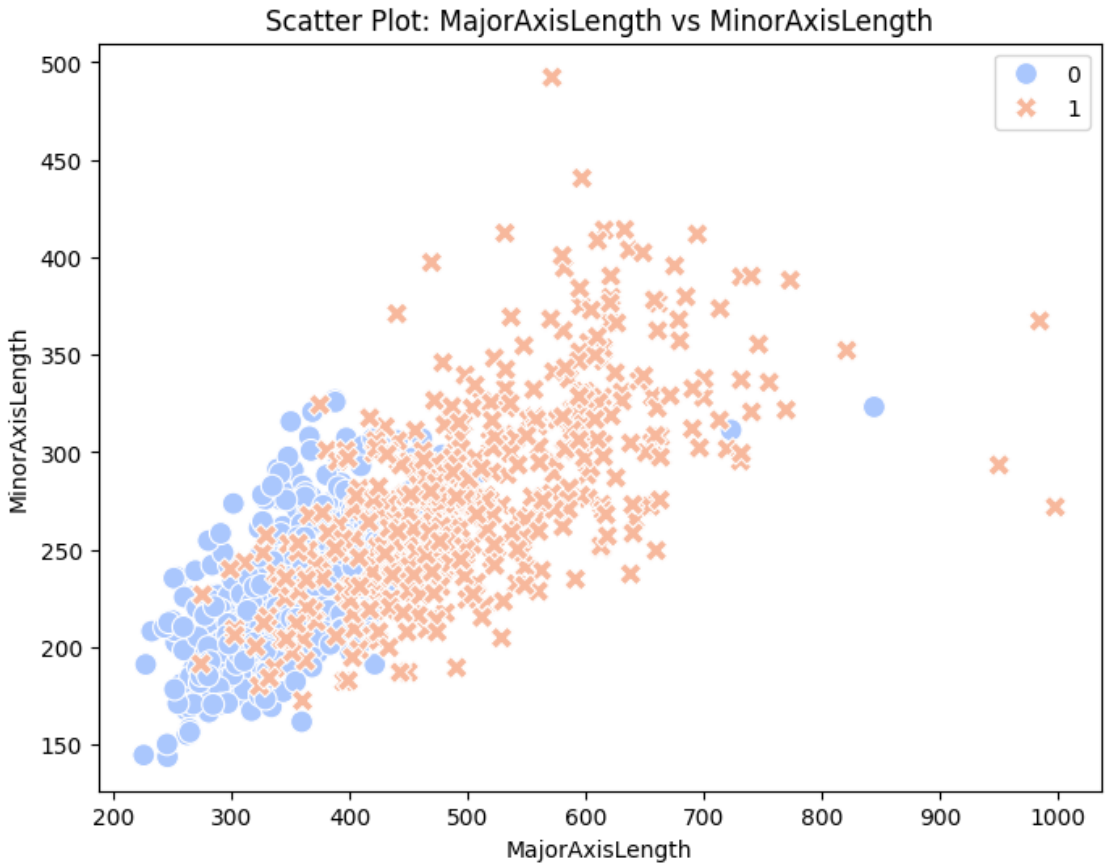

Violin Plot of ConvexArea by Class
ConvexArea
300000
250000
200000
150000
100000
50000
0
1
Class


Violin Plot of Extent by Class
Extent
0.8
0.7
0.6
0.5
0.4
0
1
Class



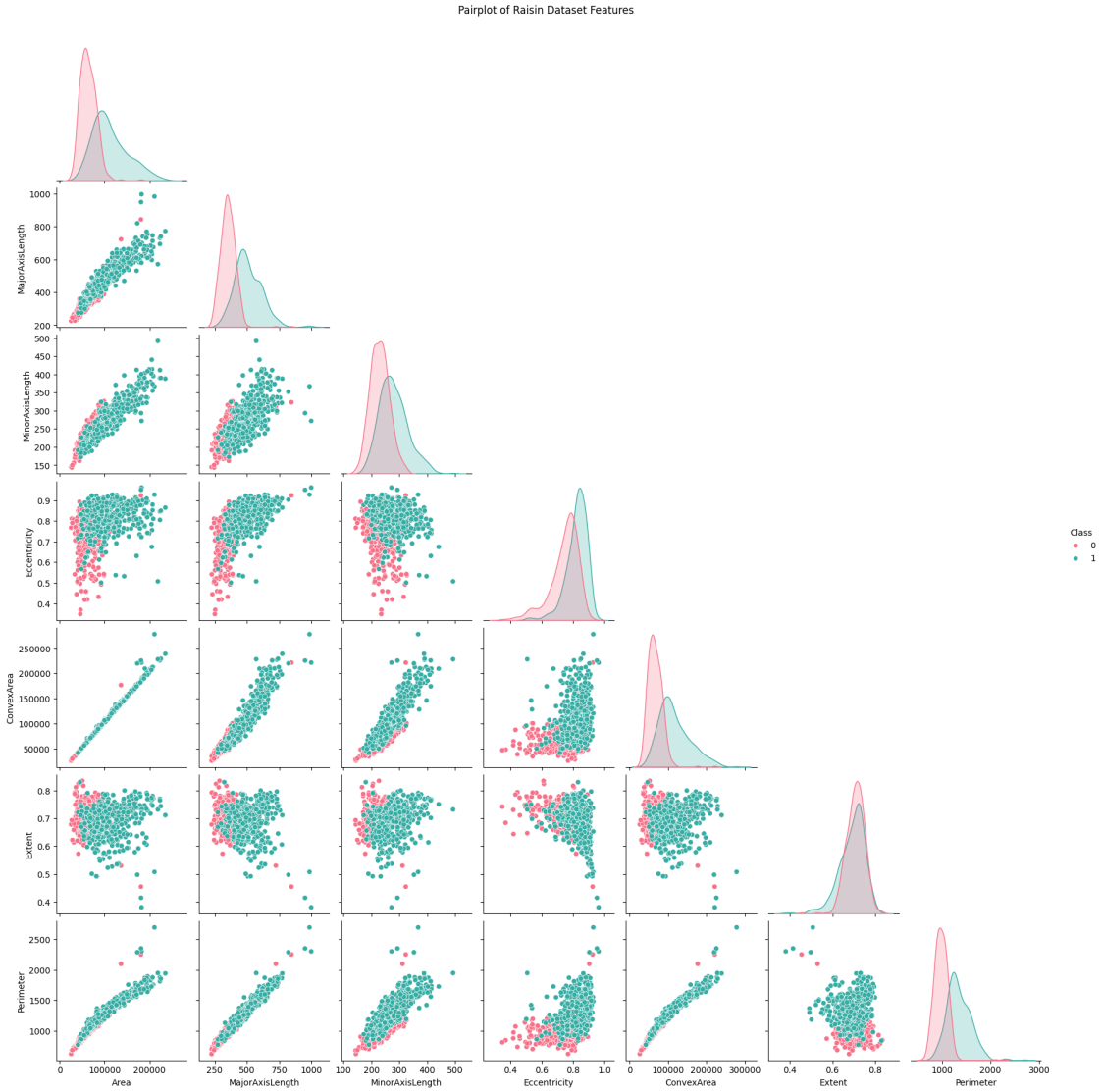
Scatter Plot Examines relationships between two key features, categorized by the class.

[11]:
```
plt.figure(figsize=(8, 6))
sns.scatterplot(data=data, x="MajorAxisLength", y="MinorAxisLength",␣
↪hue="Class", style="Class", palette="coolwarm", s=100)
plt.title("Scatter Plot: MajorAxisLength vs MinorAxisLength")
plt.xlabel("MajorAxisLength")
plt.ylabel("MinorAxisLength")
plt.legend()
plt.show()
```



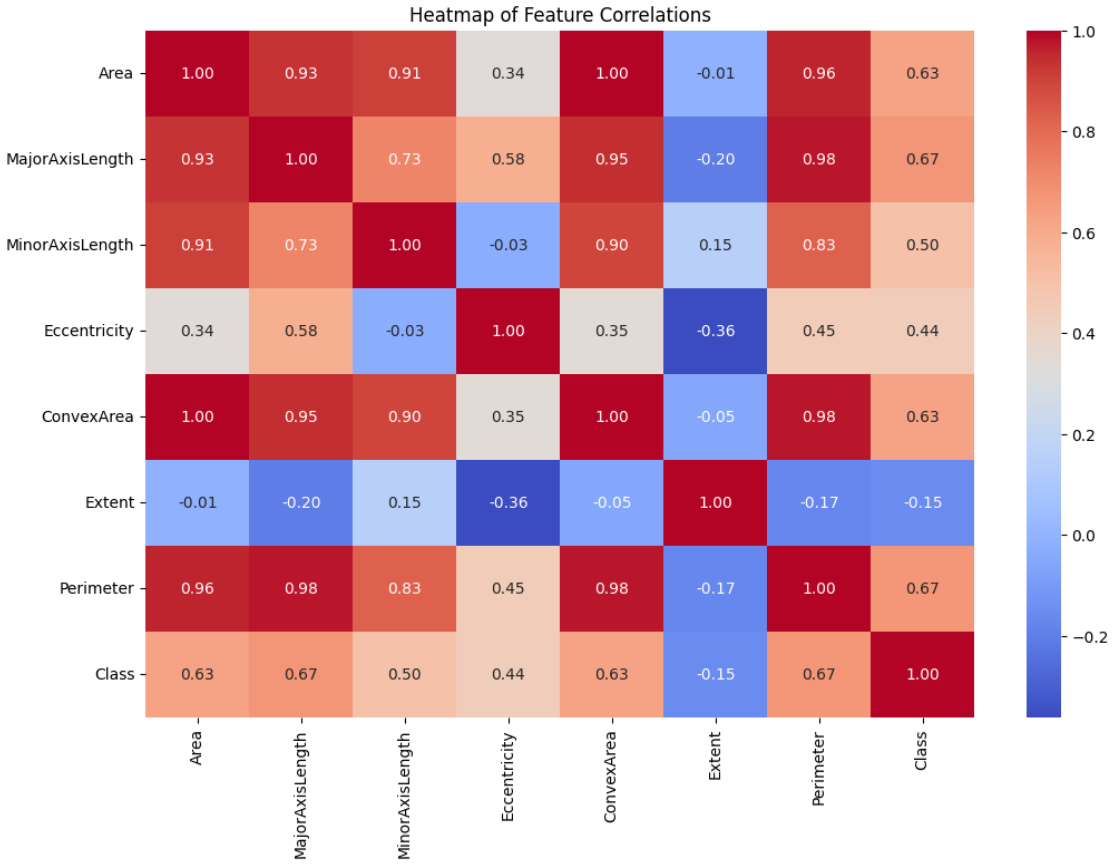
Pairplot

[12]:
```
# Pairplot for feature relationships with respect to the target
sns.pairplot(data, hue="Class", diag_kind='kde',  corner=True, palette='husl')
plt.suptitle("Pairplot of Raisin Dataset Features", y=1.02)
plt.show()
```



Heatmap to find correlation between features

[13]:
```
# Heatmap for Feature Correlations
plt.figure(figsize=(12, 8))
sns.heatmap(data.corr(), annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Heatmap of Feature Correlations")
plt.show()
```



Finding best features to train model

[14]:
```
# Select Top 6 Features Based on Correlation
correlation = data.corr()
top_features = correlation['Class'].abs().sort_values(ascending=False).index[1:
  ↪7]
print("Selected Top Features for Prediction:", list(top_features))

# Prepare Data
X = data[top_features]
y = data['Class']
```

```
Selected Top Features for Prediction: ['MajorAxisLength', 'Perimeter', 'Area',
'ConvexArea', 'MinorAxisLength', 'Eccentricity']
```

Standardization & Normalization

[15]:
```
scaler = StandardScaler()
X = scaler.fit_transform(X)
```

[16]:
```
# Normalize data if algorithms require it
from sklearn.preprocessing import MinMaxScaler
normalizer = MinMaxScaler()
X = normalizer.fit_transform(X)
```

Spliting Dataset into Train & Test Training Data - 75% & Testing Data - 25%

[17]:
```
# Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,␣
 ↪stratify=y, random_state=42)
```

1.0.5 Machine Learning Models - Random Forest & KNN

1. Train two machine learning models:

- K-Nearest Neighbors (KNN).
- Random Forest (RF).

2. Use RandomizedSearchCV for hyperparameter optimization.
3. Perform 10-fold cross-validation and evaluate metrics across all folds.

Random Search - Hyperparameters Tuning

[18]:
```
def knn_random_search(X, y):
    knn = KNeighborsClassifier()
    param_dist = {
        "n_neighbors": range(1, 10),
        "weights": ["uniform", "distance"]
    }
    search = RandomizedSearchCV(knn, param_distributions=param_dist, n_iter=20,␣
 ↪cv=5, scoring='roc_auc', random_state=42)
    search.fit(X, y)
    return search.best_estimator_, search.best_params_
```

[19]:
```
def rf_random_search(X, y):
    rf = RandomForestClassifier(random_state=42)
    param_dist = {
        "n_estimators": [10, 50, 100, 200],
        "max_depth": [None, 10, 20, 30],
        "min_samples_split": [2, 5, 10],
        "min_samples_leaf": [1, 2, 4]
    }
    search = RandomizedSearchCV(rf, param_distributions=param_dist, n_iter=20,␣
 ↪cv=5, scoring='roc_auc', random_state=42)
    search.fit(X, y)
    return search.best_estimator_, search.best_params_
```

Best Hyperparameters - KNN

[20]:
```
# Random Search and Evaluation for KNN
print("Tuning KNN...")
knn_model, knn_params = knn_random_search(X_train, y_train)
print("Best KNN Params:", knn_params)
```

```
Tuning KNN…
Best KNN Params: {'weights': 'uniform', 'n_neighbors': 9}
```

Best Hyperparameters - Random Forest

[21]:
```
# Random Search and Evaluation for Random Forest
print("\nTuning Random Forest...")
rf_model, rf_params = rf_random_search(X_train, y_train)
print("Best Random Forest Params:", rf_params)
```

```
Tuning Random Forest…
Best Random Forest Params: {'n_estimators': 100, 'min_samples_split': 5,
'min_samples_leaf': 4, 'max_depth': 10}
```

Helper Function to print all metrics

[25]:
```
def compute_metrics(y_true, y_pred, y_proba):
    cm = confusion_matrix(y_true, y_pred)

    tp, fn = cm[0][0], cm[0][1]
    fp, tn = cm[1][0], cm[1][1]
    tpr = tp / (tp + fn)
    tnr = tn / (tn + fp)
    fpr = fp / (tn + fp)
    fnr = fn / (tp + fn)
    precision = tp / (tp + fp)
    f1 = 2 * tp / (2 * tp + fp + fn)
    accuracy = (tp + tn) / (tp + fp + fn + tn)
    error_rate = (fp + fn) / (tp + fp + fn + tn)
    bacc = (tpr + tnr) / 2
    tss = tpr - fpr
    hss = 2 * (tp * tn - fp * fn) / ((tp + fn) * (fn + tn) + (tp + fp) * (fp +␣
 ↪tn))
    recall = tp / tp + fn

    roc = roc_auc_score(y_true, y_proba)

    return {
        "tp": tp, "tn": tn, "fp": fp, "fn": fn,
        "tpr": tpr, "tnr": tnr, "fpr": fpr, "fnr": fnr,
        "Accuracy": accuracy, "Precision": precision, "Error Rate": error_rate,
        "Recall": recall, "F1 Score": f1,
```

```
        "bacc": bacc, "tss": tss, "hss": hss, "roc": roc
    }
```

Helper Function to train model

[26]:
```
def evaluate_ml_model(model, X, y):
    skf = StratifiedKFold(n_splits=10, shuffle=True, random_state=42)
    all_metrics = []
    X = np.array(X)
    y = np.array(y)

    for fold, (train_idx, test_idx) in enumerate(skf.split(X, y), 1):
        X_train, X_val = X[train_idx], X[test_idx]
        y_train, y_val = y[train_idx], y[test_idx]

        model.fit(X_train, y_train)
        y_pred = model.predict(X_val)
        y_proba = model.predict_proba(X_val)[:, 1]

        metrics = compute_metrics(y_val, y_pred, y_proba)
        metrics["Fold"] = fold
        all_metrics.append(metrics)
        print(f"Fold {fold} Metrics:")
        for metric, value in metrics.items():
            print(f"{metric}: {value}")
        print('#'*30+'\n')

    return pd.DataFrame(all_metrics)
```

Training RandomForest Model

[27]:
```
rf_metrics = evaluate_ml_model(rf_model, X_train, y_train)
```

```
Fold 1 Metrics:
tp: 26
tn: 28
fp: 6
fn: 8
tpr: 0.7647058823529411
tnr: 0.8235294117647058
fpr: 0.17647058823529413
fnr: 0.23529411764705882
Accuracy: 0.7941176470588235
Precision: 0.8125
Error Rate: 0.20588235294117646
Recall: 9.0
F1 Score: 0.7878787878787878
bacc: 0.7941176470588235
```

```
tss: 0.588235294117647
hss: 0.5882352941176471
roc: 0.9238754325259515
Fold: 1
##############################

Fold 2 Metrics:
tp: 28
tn: 31
fp: 3
fn: 6
tpr: 0.8235294117647058
tnr: 0.9117647058823529
fpr: 0.08823529411764706
fnr: 0.17647058823529413
Accuracy: 0.8676470588235294
Precision: 0.9032258064516129
Error Rate: 0.1323529411764706
Recall: 7.0
F1 Score: 0.8615384615384616
bacc: 0.8676470588235294
tss: 0.7352941176470588
hss: 0.7352941176470589
roc: 0.9385813148788927
Fold: 2
##############################

Fold 3 Metrics:
tp: 28
tn: 27
fp: 7
fn: 6
tpr: 0.8235294117647058
tnr: 0.7941176470588235
fpr: 0.20588235294117646
fnr: 0.17647058823529413
Accuracy: 0.8088235294117647
Precision: 0.8
Error Rate: 0.19117647058823528
Recall: 7.0
F1 Score: 0.8115942028985508
bacc: 0.8088235294117647
tss: 0.6176470588235294
hss: 0.6176470588235294
roc: 0.8866782006920414
Fold: 3
##############################
```

```
Fold 4 Metrics:
tp: 30
tn: 26
fp: 8
fn: 4
tpr: 0.8823529411764706
tnr: 0.7647058823529411
fpr: 0.23529411764705882
fnr: 0.11764705882352941
Accuracy: 0.8235294117647058
Precision: 0.7894736842105263
Error Rate: 0.17647058823529413
Recall: 5.0
F1 Score: 0.8333333333333334
bacc: 0.8235294117647058
tss: 0.6470588235294117
hss: 0.6470588235294118
roc: 0.9013840830449826
Fold: 4
##############################

Fold 5 Metrics:
tp: 32
tn: 22
fp: 12
fn: 2
tpr: 0.9411764705882353
tnr: 0.6470588235294118
fpr: 0.35294117647058826
fnr: 0.058823529411764705
Accuracy: 0.7941176470588235
Precision: 0.7272727272727273
Error Rate: 0.20588235294117646
Recall: 3.0
F1 Score: 0.8205128205128205
bacc: 0.7941176470588236
tss: 0.588235294117647
hss: 0.5882352941176471
roc: 0.8944636678200691
Fold: 5
##############################

Fold 6 Metrics:
tp: 30
tn: 26
fp: 7
fn: 4
tpr: 0.8823529411764706
```

```
tnr: 0.7878787878787878
fpr: 0.21212121212121213
fnr: 0.11764705882352941
Accuracy: 0.835820895522388
Precision: 0.8108108108108109
Error Rate: 0.16417910447761194
Recall: 5.0
F1 Score: 0.8450704225352113
bacc: 0.8351158645276292
tss: 0.6702317290552584
hss: 0.6711289602855868
roc: 0.9242424242424243
Fold: 6
##############################

Fold 7 Metrics:
tp: 32
tn: 26
fp: 7
fn: 2
tpr: 0.9411764705882353
tnr: 0.7878787878787878
fpr: 0.21212121212121213
fnr: 0.058823529411764705
Accuracy: 0.8656716417910447
Precision: 0.8205128205128205
Error Rate: 0.13432835820895522
Recall: 3.0
F1 Score: 0.8767123287671232
bacc: 0.8645276292335116
tss: 0.7290552584670231
hss: 0.7306833407771326
roc: 0.963458110516934
Fold: 7
##############################

Fold 8 Metrics:
tp: 33
tn: 29
fp: 5
fn: 0
tpr: 1.0
tnr: 0.8529411764705882
fpr: 0.14705882352941177
fnr: 0.0
Accuracy: 0.9253731343283582
Precision: 0.868421052631579
Error Rate: 0.07462686567164178
```

```
Recall: 1.0
F1 Score: 0.9295774647887324
bacc: 0.9264705882352942
tss: 0.8529411764705882
hss: 0.851044908848377
roc: 0.9670231729055259
Fold: 8
##############################

Fold 9 Metrics:
tp: 28
tn: 28
fp: 6
fn: 5
tpr: 0.8484848484848485
tnr: 0.8235294117647058
fpr: 0.17647058823529413
fnr: 0.15151515151515152
Accuracy: 0.835820895522388
Precision: 0.8235294117647058
Error Rate: 0.16417910447761194
Recall: 6.0
F1 Score: 0.835820895522388
bacc: 0.8360071301247771
tss: 0.6720142602495544
hss: 0.6717149220489977
roc: 0.9090909090909091
Fold: 9
##############################

Fold 10 Metrics:
tp: 27
tn: 28
fp: 6
fn: 6
tpr: 0.8181818181818182
tnr: 0.8235294117647058
fpr: 0.17647058823529413
fnr: 0.18181818181818182
Accuracy: 0.8208955223880597
Precision: 0.8181818181818182
Error Rate: 0.1791044776119403
Recall: 7.0
F1 Score: 0.8181818181818182
bacc: 0.820855614973262
tss: 0.6417112299465241
hss: 0.6417112299465241
roc: 0.8997326203208557
```

```
Fold: 10
##############################
```

Training KNN Model

[28]:
```
knn_metrics = evaluate_ml_model(knn_model, X_train, y_train)
```

```
Fold 1 Metrics:
tp: 27
tn: 28
fp: 6
fn: 7
tpr: 0.7941176470588235
tnr: 0.8235294117647058
fpr: 0.17647058823529413
fnr: 0.20588235294117646
Accuracy: 0.8088235294117647
Precision: 0.8181818181818182
Error Rate: 0.19117647058823528
Recall: 8.0
F1 Score: 0.8059701492537313
bacc: 0.8088235294117647
tss: 0.6176470588235293
hss: 0.6176470588235294
roc: 0.9052768166089966
Fold: 1
##############################

Fold 2 Metrics:
tp: 29
tn: 30
fp: 4
fn: 5
tpr: 0.8529411764705882
tnr: 0.8823529411764706
fpr: 0.11764705882352941
fnr: 0.14705882352941177
Accuracy: 0.8676470588235294
Precision: 0.8787878787878788
Error Rate: 0.1323529411764706
Recall: 6.0
F1 Score: 0.8656716417910447
bacc: 0.8676470588235294
tss: 0.7352941176470588
hss: 0.7352941176470589
roc: 0.9143598615916955
Fold: 2
```

```
##############################

Fold 3 Metrics:
tp: 30
tn: 24
fp: 10
fn: 4
tpr: 0.8823529411764706
tnr: 0.7058823529411765
fpr: 0.29411764705882354
fnr: 0.11764705882352941
Accuracy: 0.7941176470588235
Precision: 0.75
Error Rate: 0.20588235294117646
Recall: 5.0
F1 Score: 0.8108108108108109
bacc: 0.7941176470588236
tss: 0.588235294117647
hss: 0.5882352941176471
roc: 0.8715397923875433
Fold: 3
##############################

Fold 4 Metrics:
tp: 31
tn: 26
fp: 8
fn: 3
tpr: 0.9117647058823529
tnr: 0.7647058823529411
fpr: 0.23529411764705882
fnr: 0.08823529411764706
Accuracy: 0.8382352941176471
Precision: 0.7948717948717948
Error Rate: 0.16176470588235295
Recall: 4.0
F1 Score: 0.8493150684931506
bacc: 0.838235294117647
tss: 0.6764705882352942
hss: 0.6764705882352942
roc: 0.8875432525951558
Fold: 4
##############################

Fold 5 Metrics:
tp: 32
tn: 23
fp: 11
```

fn: 2
tpr: 0.9411764705882353
tnr: 0.6764705882352942
fpr: 0.3235294117647059
fnr: 0.058823529411764705
Accuracy: 0.8088235294117647
Precision: 0.7441860465116279
Error Rate: 0.19117647058823528
Recall: 3.0
F1 Score: 0.8311688311688312
bacc: 0.8088235294117647
tss: 0.6176470588235294
hss: 0.6176470588235294
roc: 0.8568339100346021
Fold: 5
##############################

Fold 6 Metrics:
tp: 33
tn: 27
fp: 6
fn: 1
tpr: 0.9705882352941176
tnr: 0.8181818181818182
fpr: 0.18181818181818182
fnr: 0.029411764705882353
Accuracy: 0.8955223880597015
Precision: 0.8461538461538461
Error Rate: 0.1044776119402985
Recall: 2.0
F1 Score: 0.9041095890410958
bacc: 0.8943850267379679
tss: 0.7887700534759359
hss: 0.7905314872711031
roc: 0.9024064171122994
Fold: 6
##############################

Fold 7 Metrics:
tp: 33
tn: 25
fp: 8
fn: 1
tpr: 0.9705882352941176
tnr: 0.7575757575757576
fpr: 0.24242424242424243
fnr: 0.029411764705882353
Accuracy: 0.8656716417910447

```
Precision: 0.8048780487804879
Error Rate: 0.13432835820895522
Recall: 2.0
F1 Score: 0.88
bacc: 0.8640819964349375
tss: 0.7281639928698752
hss: 0.7304425569959767
roc: 0.9425133689839571
Fold: 7
##############################

Fold 8 Metrics:
tp: 33
tn: 28
fp: 6
fn: 0
tpr: 1.0
tnr: 0.8235294117647058
fpr: 0.17647058823529413
fnr: 0.0
Accuracy: 0.9104477611940298
Precision: 0.8461538461538461
Error Rate: 0.08955223880597014
Recall: 1.0
F1 Score: 0.9166666666666666
bacc: 0.9117647058823529
tss: 0.8235294117647058
hss: 0.8213333333333334
roc: 0.9380570409982175
Fold: 8
##############################

Fold 9 Metrics:
tp: 28
tn: 27
fp: 7
fn: 5
tpr: 0.8484848484848485
tnr: 0.7941176470588235
fpr: 0.20588235294117646
fnr: 0.15151515151515152
Accuracy: 0.8208955223880597
Precision: 0.8
Error Rate: 0.1791044776119403
Recall: 6.0
F1 Score: 0.8235294117647058
bacc: 0.821301247771836
tss: 0.642602495543672
```

```
hss: 0.6420302760463046
roc: 0.8908199643493762
Fold: 9
##############################

Fold 10 Metrics:
tp: 28
tn: 26
fp: 8
fn: 5
tpr: 0.8484848484848485
tnr: 0.7647058823529411
fpr: 0.23529411764705882
fnr: 0.15151515151515152
Accuracy: 0.8059701492537313
Precision: 0.7777777777777778
Error Rate: 0.19402985074626866
Recall: 6.0
F1 Score: 0.8115942028985508
bacc: 0.8065953654188949
tss: 0.6131907308377897
hss: 0.6123720516243881
roc: 0.8872549019607843
Fold: 10
##############################
```

1.0.6 DL Model - LSTM

Train a deep learning model using LSTM: * Sequential LSTM architecture for handling sequential or structured data. * Use random search to optimize batch size and epochs. * Perform 10-fold cross-validation to evaluate performance.

Helper Functions

[29]:
```
def build_lstm(input_shape):
    model = Sequential()
    model.add(LSTM(32, input_shape=input_shape))
    model.add(Dense(1, activation='sigmoid'))
    model.compile(optimizer='adam', loss='binary_crossentropy',␣
 ↪metrics=['accuracy'])
    return model
```

[30]:
```
def lstm_random_search(X, y):
    lstm = KerasClassifier(build_fn=lambda: build_lstm((X.shape[1], 1)),␣
 ↪verbose=0)
    param_dist = {
        "batch_size": [16, 32, 64],
```

```
        "epochs": [10, 20, 30]
    }
    search = RandomizedSearchCV(lstm, param_distributions=param_dist,␣
 ↪n_iter=10, cv=3, scoring='roc_auc', random_state=42)
    search.fit(X, y)
    return search.best_estimator_, search.best_params_
```

[31]:
```
def evaluate_lstm_model(X, y):
    skf = StratifiedKFold(n_splits=10, shuffle=True, random_state=42)
    all_metrics = []
    X = np.array(X)
    y = np.array(y)

    for fold, (train_idx, test_idx) in enumerate(skf.split(X, y), 1):
        X_train, X_val = np.expand_dims(X[train_idx], axis=2), np.
 ↪expand_dims(X[test_idx], axis=2)
        y_train, y_val = y[train_idx], y[test_idx]

        model = build_lstm((X_train.shape[1], 1))
        model.fit(X_train, y_train, epochs=10, batch_size=32, verbose=0)
        y_proba = model.predict(X_val).ravel()
        y_pred = (y_proba > 0.5).astype(int)

        metrics = compute_metrics(y_val, y_pred, y_proba)
        metrics["Fold"] = fold
        all_metrics.append(metrics)
        print(f"Fold {fold} Metrics:")
        for metric, value in metrics.items():
            print(f"{metric}: {value}")
        print('#'*30+'\n')

    return pd.DataFrame(all_metrics)
```

[32]:
```
import warnings
warnings.filterwarnings('ignore')
```

Hyperparameters Tuning - RandomSearch

[33]:
```
# Random Search and Evaluation for LSTM
print("\nTuning LSTM...")
lstm_model, lstm_params = lstm_random_search(np.expand_dims(X_train, axis=2),␣
 ↪y_train)
print("Best LSTM Params:", lstm_params)
```

```
Tuning LSTM…

WARNING:tensorflow:5 out of the last 24 calls to <function
```

```
TensorFlowTrainer.make_predict_function.<locals>.one_step_on_data_distributed at
0x7849410edab0> triggered tf.function retracing. Tracing is expensive and the
excessive number of tracings could be due to (1) creating @tf.function
repeatedly in a loop, (2) passing tensors with different shapes, (3) passing
Python objects instead of tensors. For (1), please define your @tf.function
outside of the loop. For (2), @tf.function has reduce_retracing=True option that
can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:5 out of the last 17 calls to <function
TensorFlowTrainer.make_predict_function.<locals>.one_step_on_data_distributed at
0x78493d5b12d0> triggered tf.function retracing. Tracing is expensive and the
excessive number of tracings could be due to (1) creating @tf.function
repeatedly in a loop, (2) passing tensors with different shapes, (3) passing
Python objects instead of tensors. For (1), please define your @tf.function
outside of the loop. For (2), @tf.function has reduce_retracing=True option that
can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

Best LSTM Params: {'epochs': 30, 'batch_size': 16}
```

Model Training

[34]:
```
lstm_metrics = evaluate_lstm_model(X_train, y_train)
```

```
3/3             0s 70ms/step
Fold 1 Metrics:
tp: 28
tn: 28
fp: 6
fn: 6
tpr: 0.8235294117647058
tnr: 0.8235294117647058
fpr: 0.17647058823529413
fnr: 0.17647058823529413
Accuracy: 0.8235294117647058
Precision: 0.8235294117647058
Error Rate: 0.17647058823529413
Recall: 7.0
F1 Score: 0.8235294117647058
bacc: 0.8235294117647058
tss: 0.6470588235294117
hss: 0.6470588235294118
roc: 0.8979238754325258
Fold: 1
##############################

3/3             0s 71ms/step
```

```
Fold 2 Metrics:
tp: 28
tn: 31
fp: 3
fn: 6
tpr: 0.8235294117647058
tnr: 0.9117647058823529
fpr: 0.08823529411764706
fnr: 0.17647058823529413
Accuracy: 0.8676470588235294
Precision: 0.9032258064516129
Error Rate: 0.1323529411764706
Recall: 7.0
F1 Score: 0.8615384615384616
bacc: 0.8676470588235294
tss: 0.7352941176470588
hss: 0.7352941176470589
roc: 0.9256055363321799
Fold: 2
##############################

3/3             0s 120ms/step
Fold 3 Metrics:
tp: 30
tn: 25
fp: 9
fn: 4
tpr: 0.8823529411764706
tnr: 0.7352941176470589
fpr: 0.2647058823529412
fnr: 0.11764705882352941
Accuracy: 0.8088235294117647
Precision: 0.7692307692307693
Error Rate: 0.19117647058823528
Recall: 5.0
F1 Score: 0.821917808219178
bacc: 0.8088235294117647
tss: 0.6176470588235294
hss: 0.6176470588235294
roc: 0.8866782006920416
Fold: 3
##############################

3/3             0s 70ms/step
Fold 4 Metrics:
tp: 31
tn: 26
fp: 8
```

```
fn: 3
tpr: 0.9117647058823529
tnr: 0.7647058823529411
fpr: 0.23529411764705882
fnr: 0.08823529411764706
Accuracy: 0.8382352941176471
Precision: 0.7948717948717948
Error Rate: 0.16176470588235295
Recall: 4.0
F1 Score: 0.8493150684931506
bacc: 0.838235294117647
tss: 0.6764705882352942
hss: 0.6764705882352942
roc: 0.8875432525951557
Fold: 4
##############################

3/3             0s 69ms/step
Fold 5 Metrics:
tp: 32
tn: 21
fp: 13
fn: 2
tpr: 0.9411764705882353
tnr: 0.6176470588235294
fpr: 0.38235294117647056
fnr: 0.058823529411764705
Accuracy: 0.7794117647058824
Precision: 0.7111111111111111
Error Rate: 0.22058823529411764
Recall: 3.0
F1 Score: 0.810126582278481
bacc: 0.7794117647058824
tss: 0.5588235294117647
hss: 0.5588235294117647
roc: 0.9143598615916955
Fold: 5
##############################

3/3             0s 71ms/step
Fold 6 Metrics:
tp: 31
tn: 26
fp: 7
fn: 3
tpr: 0.9117647058823529
tnr: 0.7878787878787878
fpr: 0.21212121212121213
```

```
fnr: 0.08823529411764706
Accuracy: 0.8507462686567164
Precision: 0.8157894736842105
Error Rate: 0.14925373134328357
Recall: 4.0
F1 Score: 0.8611111111111112
bacc: 0.8498217468805704
tss: 0.6996434937611408
hss: 0.7008928571428571
roc: 0.9099821746880571
Fold: 6
##############################

3/3             0s 127ms/step
Fold 7 Metrics:
tp: 33
tn: 25
fp: 8
fn: 1
tpr: 0.9705882352941176
tnr: 0.7575757575757576
fpr: 0.24242424242424243
fnr: 0.029411764705882353
Accuracy: 0.8656716417910447
Precision: 0.8048780487804879
Error Rate: 0.13432835820895522
Recall: 2.0
F1 Score: 0.88
bacc: 0.8640819964349375
tss: 0.7281639928698752
hss: 0.7304425569959767
roc: 0.9509803921568628
Fold: 7
##############################

3/3             0s 70ms/step
Fold 8 Metrics:
tp: 33
tn: 26
fp: 8
fn: 0
tpr: 1.0
tnr: 0.7647058823529411
fpr: 0.23529411764705882
fnr: 0.0
Accuracy: 0.8805970149253731
Precision: 0.8048780487804879
Error Rate: 0.11940298507462686
```

```
Recall: 1.0
F1 Score: 0.8918918918918919
bacc: 0.8823529411764706
tss: 0.7647058823529411
hss: 0.7619893428063943
roc: 0.9661319073083778
Fold: 8
##############################

3/3             0s 71ms/step
Fold 9 Metrics:
tp: 30
tn: 26
fp: 8
fn: 3
tpr: 0.9090909090909091
tnr: 0.7647058823529411
fpr: 0.23529411764705882
fnr: 0.09090909090909091
Accuracy: 0.835820895522388
Precision: 0.7894736842105263
Error Rate: 0.16417910447761194
Recall: 4.0
F1 Score: 0.8450704225352113
bacc: 0.8368983957219251
tss: 0.6737967914438503
hss: 0.6722987994664296
roc: 0.8921568627450981
Fold: 9
##############################

3/3             0s 70ms/step
Fold 10 Metrics:
tp: 29
tn: 24
fp: 10
fn: 4
tpr: 0.8787878787878788
tnr: 0.7058823529411765
fpr: 0.29411764705882354
fnr: 0.12121212121212122
Accuracy: 0.7910447761194029
Precision: 0.7435897435897436
Error Rate: 0.208955223880597
Recall: 5.0
F1 Score: 0.8055555555555556
bacc: 0.7923351158645277
tss: 0.5846702317290553
```

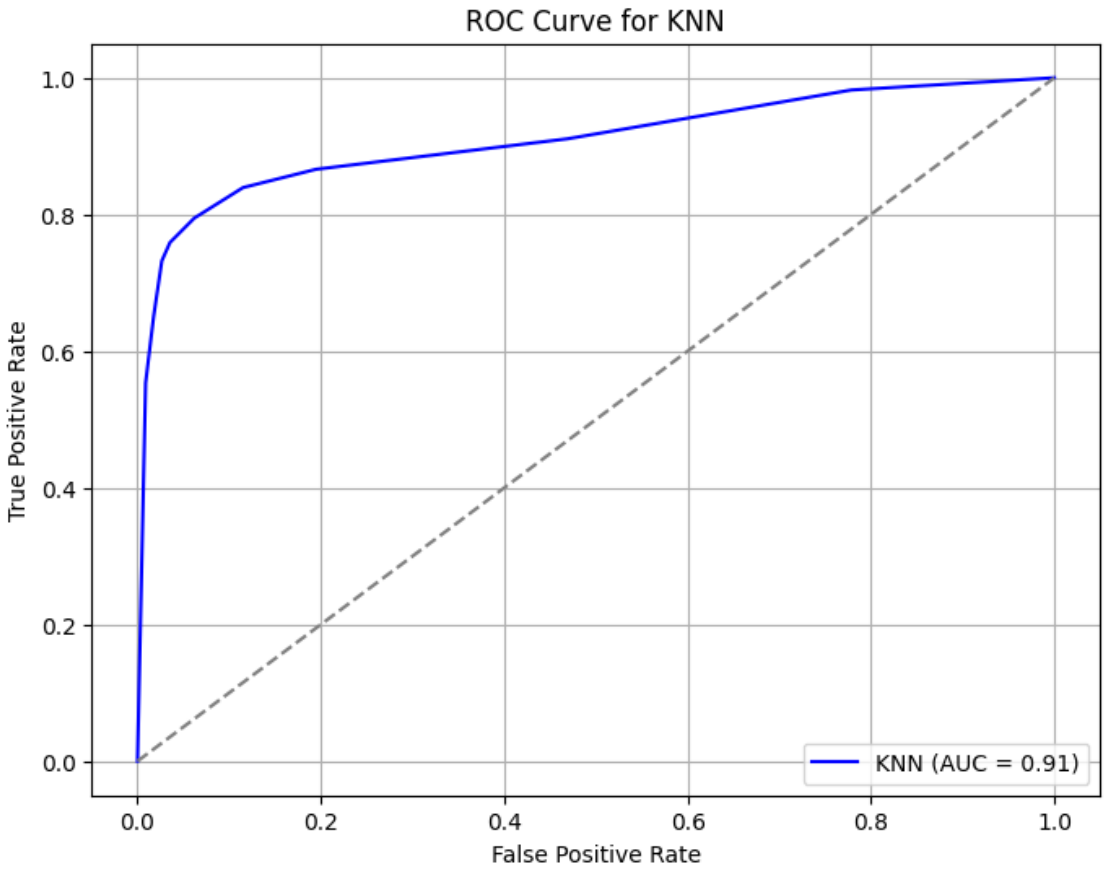
```
hss: 0.5831111111111111
roc: 0.8796791443850268
Fold: 10
##############################
```

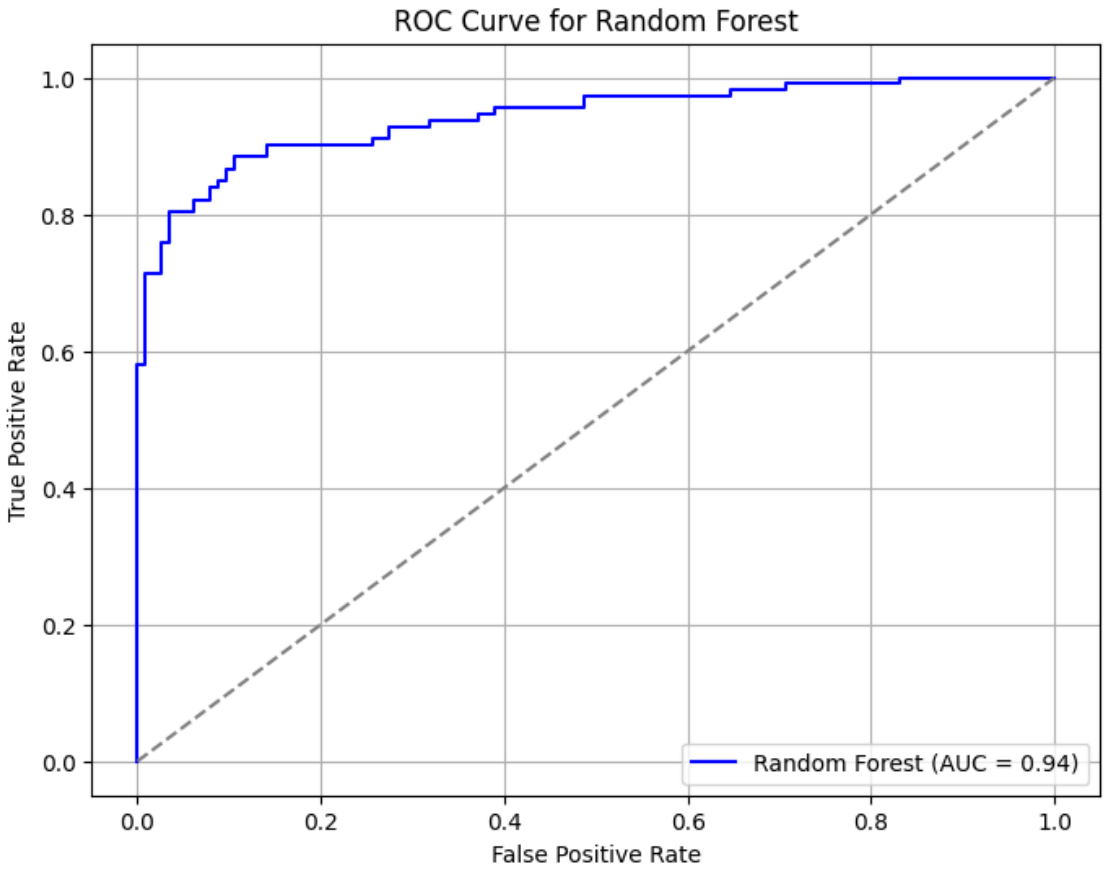
1.0.7 ROC Curve

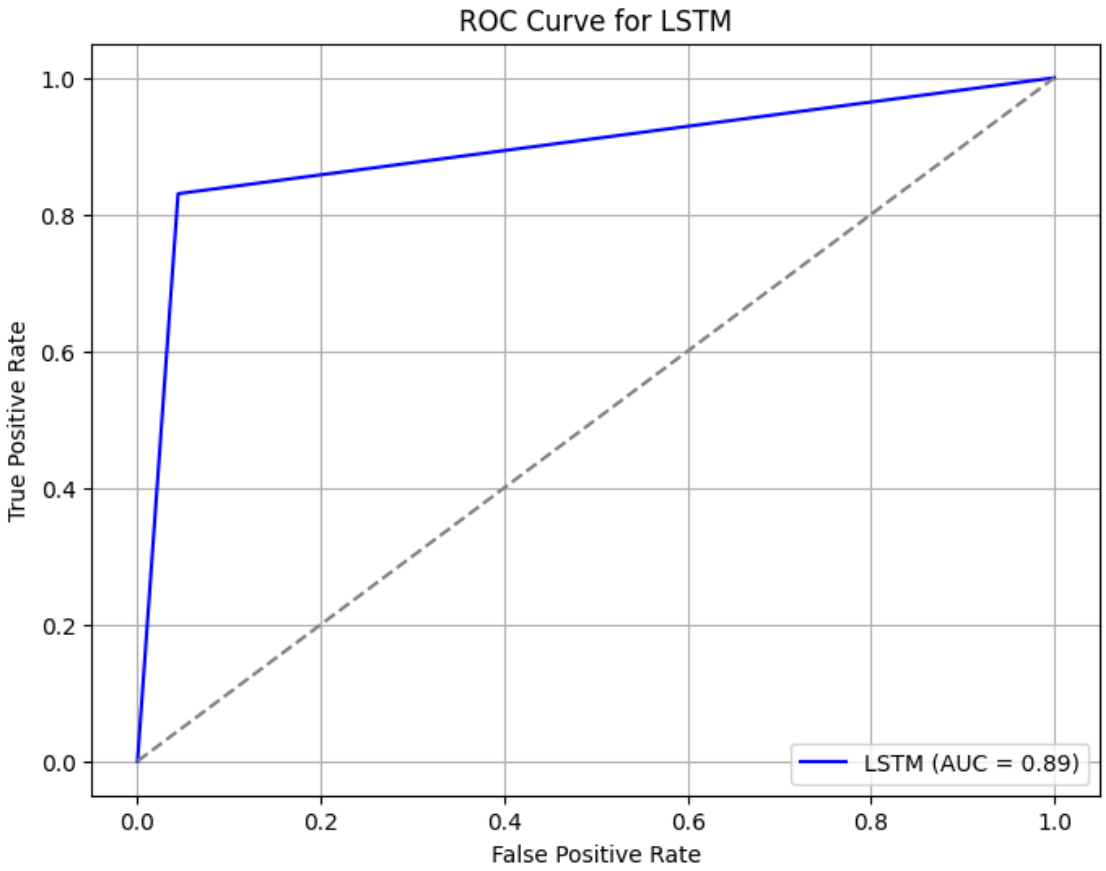
[35]:
```
def plot_roc_curve(model_name, y_test, y_proba):
    fpr, tpr, _ = roc_curve(y_test, y_proba)
    roc_auc = auc(fpr, tpr)
    plt.figure(figsize=(8, 6))
    plt.plot(fpr, tpr, color='blue', label=f"{model_name} (AUC = {roc_auc:.
 ↪2f})")
    plt.plot([0, 1], [0, 1], color='gray', linestyle='--')
    plt.title(f"ROC Curve for {model_name}")
    plt.xlabel("False Positive Rate")
    plt.ylabel("True Positive Rate")
    plt.legend(loc="lower right")
    plt.grid()
    plt.show()
```

[36]:
```
# Plot ROC Curves
print("\nPlotting ROC Curves...")
plot_roc_curve("KNN", y_test, knn_model.predict_proba(X_test)[:, 1])
plot_roc_curve("Random Forest", y_test, rf_model.predict_proba(X_test)[:, 1])
plot_roc_curve("LSTM", y_test, lstm_model.predict(np.expand_dims(X_test,␣
 ↪axis=2)).ravel())
```

```
Plotting ROC Curves…
```


ROC Curve for KNN
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
KNN (AUC = 0.91)


ROC Curve for Random Forest
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Random Forest (AUC = 0.94)



1.0.8 Models Comparision

Compare the performance of KNN, Random Forest, and LSTM models using the metrics from the 10th fold.

[37]:
```
rf_metrics['Model'] = 'Random Forest'
knn_metrics['Model'] = 'KNN'
lstm_metrics['Model'] = 'LSTM'
all_metrics = pd.concat([rf_metrics, knn_metrics, lstm_metrics],␣
 ↪ignore_index=True)
```

[38]:
```
all_metrics = all_metrics[all_metrics['Fold']==10]
all_metrics.set_index('Model', inplace=True)
all_metrics.T
```

[38]:

Model	Random Forest	KNN	LSTM
tp	27.000000	28.000000	29.000000
tn	28.000000	26.000000	24.000000
fp	6.000000	8.000000	10.000000
fn	6.000000	5.000000	4.000000

```
tpr             0.818182   0.848485   0.878788
tnr             0.823529   0.764706   0.705882
fpr             0.176471   0.235294   0.294118
fnr             0.181818   0.151515   0.121212
Accuracy        0.820896   0.805970   0.791045
Precision       0.818182   0.777778   0.743590
Error Rate      0.179104   0.194030   0.208955
Recall          7.000000   6.000000   5.000000
F1 Score        0.818182   0.811594   0.805556
bacc            0.820856   0.806595   0.792335
tss             0.641711   0.613191   0.584670
hss             0.641711   0.612372   0.583111
roc             0.899733   0.887255   0.879679
Fold           10.000000  10.000000  10.000000
```

[38]: