NAME : Sharath Shankar Rathakrishnan
NJIT UCID : sr2667
EMAIL ADDRESS : sr2667@njit.edu
DATE : 11/24/2024
PROFESSOR : Yasser Abduallah
CS 634101 Data Mining

FINAL TERM PROJECT REPORT

Implementation and Code Usage

Core Concepts and Principles:

Positive Examples (P): Total true positive samples.

Negative Examples (N): Total true negative samples.

True Positive Rate (TPR): Percentage of actual positives correctly identified by the model.

True Negative Rate (TNR): Percentage of actual negatives correctly identified by the model.

False Positive Rate (FPR): Percentage of negatives misclassified as positives.

False Negative Rate (FNR): Percentage of positives misclassified as negatives.

Accuracy: Overall correctness of the model's predictions.

Precision: Fraction of predicted positives that are true positives.

Recall: Fraction of actual positives correctly identified by the model (same as TPR).

F1-Score: Harmonic mean of precision and recall.

AUC: Area under the ROC curve measuring overall model performance.

Balanced Accuracy (BACC): Average of TPR and TNR.

True Skill Statistics (TSS): Measure of model skill compared to random guessing.

Heidke Skill Score (HSS): Skill metric accounting for random chance.

Brier Score (BS): Mean squared error of predicted probabilities.

Brier Skill Score (BSS): BS improvement over a baseline model.

Error Rate (ERR): Fraction of incorrect predictions.

Project Workflow:

Environment Setup:

- Python 3.9: The programming language utilized for building the machine learning models.

- Jupyter Notebook (ipykernel): The interactive platform used for writing and executing Python code.

Loading Dataset:
- Load the breast cancer dataset using sklearn.datasets.load_breast_cancer.
- Extract features (X) and labels (y).

K-Fold Cross-Validation Setup:
- Initialize KFold with 10 splits, shuffling, and a fixed random state.

Metrics and Helper Functions:
- Define calculate_evaluation_metrics to compute performance metrics.
- Define build_roc for building ROC curves.
- Define calculate_averages to calculate mean metrics across folds.

Model Training and Evaluation:
Random Forest (RF):
- Train a RandomForestClassifier for each fold.
- Predict class labels and probabilities.
- Calculate metrics and append results for all folds then print all.

Naive Bayes (NB):
- Train a GaussianNB model for each fold.
- Predict class labels and probabilities.
- Calculate metrics and append results for all folds then print all.

Long Short-Term Memory (LSTM):
- Scale and reshape data for LSTM.
- Define and train an LSTM model for each fold.
- Predict class probabilities, threshold them, and compute metrics then print all.

Results Compilation:
- Compute average metrics for each model using compute_averages.
- Create a metrics comparison table using pandas.DataFrame.

ROC Curve Plotting:
- Interpolate and average ROC curves for all models.
- Plot mean ROC curves with AUC values.

Visualization and Reporting:
- Print per-fold metrics for each model.

- Display a comparative metrics table.
- Show the ROC curve plot for all models.

Screenshots

Before running the code for the implementation of Random Forest, Naive Bayes and Long Short-Term Memory (LSTM) models, make sure the following packages are installed; ***scikit-learn, matplotlib, tensorflow, numpy, and pandas***. If they are not installed, you can do so by executing the following command:

pip install scikit-learn matplotlib tensorflow numpy pandas

Below are the screenshots of the code from the .ipynb file:

```
from sklearn.datasets import load_breast_cancer
import matplotlib.pyplot as plt
from sklearn.model_selection import KFold
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import roc_curve, auc, confusion_matrix, accuracy_score, precision_score, recall_score, f1_score, brier_score_loss
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
import numpy as np
import pandas as pd
```

1. In the project, the dataset is imported using the load_breast_cancer function from sklearn.datasets, and there is no data file explicitly in the zip folder as the data has been obtained directly from sklearn.

```
dataset = load_breast_cancer()
X = dataset.data
y = dataset.target
```

2. KFold is used to split the data into 10 shuffled parts for cross-validation, and an empty dictionary stores results for Random Forest (RF), Naive Bayes (NB), and LSTM models.

```
kf = KFold(n_splits=10, shuffle=True, random_state=42)
results = {'RF': [], 'NB': [], 'LSTM': []}
```

3. These lists are used to store the false positive rates (FPR), true positive rates (TPR), and area under the curve (AUC) values for the Random Forest (RF), Naive Bayes (NB), and LSTM models.

```
fpr_rf, tpr_rf, auc_rf = [], [], []
fpr_nb, tpr_nb, auc_nb = [], [], []
fpr_lstm, tpr_lstm, auc_lstm = [], [], []
```

4. This function, ***calculate_evaluation_metrics***, calculates and returns various evaluation metrics. This function is used to calculate metrics such as Positive Examples (P), Negative Examples (N), True Positive Rate (TPR), True Negative Rate (TNR), False Positive Rate (FPR), False Negative Rate (FNR), Accuracy, Precision, Recall, F1-Score, AUC, Balanced Accuracy (BACC), True Skill Statistics (TSS), Heidke Skill Score (HSS), Brier Score (BS), Brier Skill Score (BSS), Error Rate (ERR).

```
def calculate_evaluation_metrics(y_true, y_pred, y_prob=None, fpr=None, tpr=None):
    cm = confusion_matrix(y_true, y_pred)
    TN, FP, FN, TP = cm.ravel()

    P = TP + FN  #Total positives (P)
    N = TN + FP  #Total negatives (N)

    tpr_value = TP / P if P > 0 else 0  # True Positive Rate (Sensitivity)
    tnr_value = TN / N if N > 0 else 0  # True Negative Rate (Specificity)
    fpr_value = FP / N if N > 0 else 0  # False Positive Rate
    fnr_value = FN / P if P > 0 else 0  # False Negative Rate
    error_rate = (FP + FN) / (P + N)  # Error Rate

    accuracy = accuracy_score(y_true, y_pred)
    precision = precision_score(y_true, y_pred, zero_division=0)
    recall = recall_score(y_true, y_pred, zero_division=0)
    f1 = f1_score(y_true, y_pred)
    auc_value = auc(fpr, tpr) if fpr is not None and tpr is not None else None

    bacc = (TP / (TP + FN) + TN / (TN + FP)) / 2  # Balanced Accuracy
    tss = (TP / (TP + FN)) - (FP / (FP + TN))  # True Skill Statistic
    hss = (2 * (TP * TN - FP * FN)) / ((TP + FN) * (FN + TN) + (TP + FP) * (FP + TN))  # Heidke Skill Score

    bs = brier_score_loss(y_true, y_prob) if y_prob is not None else None
    bss = None
    if y_prob is not None:
        y_mean = np.mean(y_true)
        bss = 1 - (bs / np.mean((y_true - y_mean) ** 2))

    return {
        'Positive Examples (P)': P,
        'Negative Examples (N)': N,
        'True Positive Rate (TPR)': tpr_value,
        'True Negative Rate (TNR)': tnr_value,
        'False Positive Rate (FPR)': fpr_value,
        'False Negative Rate (FNR)': fnr_value,
        'Accuracy': accuracy,
        'Precision': precision,
        'Recall': recall,
        'F1-Score': f1,
        'AUC': auc_value,
        'Balanced Accuracy (BACC)': bacc,
        'True Skill Statistics (TSS)': tss,
        'Heidke Skill Score (HSS)': hss,
        'Brier Score (BS)': bs,
        'Brier Skill Score (BSS)': bss,
        'Error Rate (ERR)': error_rate
    }
```

5. The ***build_roc*** function creates an average ROC curve by aligning and averaging multiple ROC curves. It interpolates tpr values for evenly spaced fpr points, ensures the curve starts at (0, 0), and returns the averaged fpr and tpr values.

```
def build_roc(fpr_list, tpr_list, num_points=100):
    mean_fpr = np.linspace(0, 1, num_points)
    mean_tpr = np.mean([np.interp(mean_fpr, fpr, tpr) for fpr, tpr in zip(fpr_list, tpr_list)], axis=0)
    mean_tpr = np.insert(mean_tpr, 0, 0)
    mean_fpr = np.insert(mean_fpr, 0, 0)
    return mean_fpr, mean_tpr
```

6. The ***calculate_averages*** function computes the average of each metric from a list of metric dictionaries. For every metric, it skips None values, calculates the mean of valid ones, and returns the averages in a new dictionary. This is later used to print the ‘Comparison of Evaluation Metrics’ table.

```python
def calculate_averages(metrics_list):
    averages = {}
    for metric in metrics_list[0]:
        valid_values = [metrics[metric] for metrics in metrics_list if metrics[metric] is not None]
        averages[metric] = np.mean(valid_values) if valid_values else None
    return averages
```

7. The code runs a 10-fold cross-validation for the Random Forest model. For each fold, it splits the data into training and testing sets, trains the model, and makes predictions. It calculates all the evaluation metrics using the ***calculate_evaluation_metrics*** function. Finally, it prints the evaluation metrics for each fold.

Random Forest Model

```python
all_evaluation_metrics_rf = []
print("\033[1m" + "\n10 FOLD METRICS FOR RANDOM FOREST\n" + "\033[0m")
print()
for i, (train_index, test_index) in enumerate(kf.split(X), start=1):
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]

    rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
    rf_model.fit(X_train, y_train)
    y_pred_rf = rf_model.predict(X_test)
    y_prob_rf = rf_model.predict_proba(X_test)[:, 1]

    #Calculating all evaluation metrics for Random Forest
    fpr, tpr, _ = roc_curve(y_test, y_prob_rf)
    metrics_rf = calculate_evaluation_metrics(y_test, y_pred_rf, y_prob_rf, fpr, tpr)
    fpr_rf.append(fpr)
    tpr_rf.append(tpr)
    auc_rf.append(auc(fpr, tpr))

    all_evaluation_metrics_rf.append(metrics_rf)

    #Printing metrics for all 10 folds
    print(f"Random Forest - Fold {i}")
    for metric, value in metrics_rf.items():
        print(f"{metric}: {value}")
    print()
print()
```

8. The code runs a 10-fold cross-validation for the Naive Bayes model. For each fold, it splits the data into training and testing sets, trains the model, and makes predictions. It calculates all the evaluation metrics using the ***calculate_evaluation_metrics*** function. Finally, it prints the evaluation metrics for each fold.

Naive Bayes Model

```python
all_evaluation_metrics_nb = []
print("\033[1m" + "\n10 FOLD METRICS FOR NAIVE BAYES\n" + "\033[0m")
print()
for i, (train_index, test_index) in enumerate(kf.split(X), start=1):
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]

    nb_model = GaussianNB()
    nb_model.fit(X_train, y_train)
    y_pred_nb = nb_model.predict(X_test)
    y_prob_nb = nb_model.predict_proba(X_test)[:, 1]

    #Calculating all evaluation metrics for Naive Bayes
    fpr, tpr, _ = roc_curve(y_test, y_prob_nb)
    metrics_nb = calculate_evaluation_metrics(y_test, y_pred_nb, y_prob_nb, fpr, tpr)
    fpr_nb.append(fpr)
    tpr_nb.append(tpr)
    auc_nb.append(auc(fpr, tpr))

    all_evaluation_metrics_nb.append(metrics_nb)

    #Printing metrics for all 10 folds
    print(f"Naive Bayes - Fold {i}")
    for metric, value in metrics_nb.items():
        print(f"{metric}: {value}")
    print()
```

9. The code runs a 10-fold cross-validation for the LSTM model. For each fold, it splits the data into training and testing sets, scales and reshapes the data for LSTM, and trains the model. It makes predictions and calculates all the evaluation metrics using the ***calculate_evaluation_metrics*** function. Finally, it prints the evaluation metrics for each fold.

LSTM Model

```
all_evaluation_metrics_lstm = []
print("\033[1m" + "\n10 FOLD METRICS FOR LSTM\n" + "\033[0m")
print()
for i, (train_index, test_index) in enumerate(kf.split(X), start=1):
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]

    #Scaling and reshaping for LSTM
    scaler = MinMaxScaler()
    X_train = scaler.fit_transform(X_train)
    X_test = scaler.transform(X_test)
    X_train = X_train.reshape((X_train.shape[0], X_train.shape[1], 1))
    X_test = X_test.reshape((X_test.shape[0], X_test.shape[1], 1))

    #Defining the LSTM model with the Input Layer
    from tensorflow.keras.layers import Input

    lstm_model = Sequential([
        Input(shape=(X_train.shape[1], 1)),
        LSTM(50, activation='relu'),
        Dense(1, activation='sigmoid')
    ])
    lstm_model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
    lstm_model.fit(X_train, y_train, epochs=20, verbose=0)

    y_prob_lstm = lstm_model.predict(X_test).ravel()
    y_pred_lstm = (y_prob_lstm > 0.5).astype(int)

    #Calculating all evaluation metrics for LSTM
    fpr, tpr, _ = roc_curve(y_test, y_prob_lstm)
    metrics_lstm = calculate_evaluation_metrics(y_test, y_pred_lstm, y_prob_lstm, fpr, tpr)
    fpr_lstm.append(fpr)
    tpr_lstm.append(tpr)
    auc_lstm.append(auc(fpr, tpr))

    all_evaluation_metrics_lstm.append(metrics_lstm)

    #Printing metrics for all 10 folds
    print(f"LSTM - Fold {i}")
    for metric, value in metrics_lstm.items():
        print(f"{metric}: {value}")
    print()
```

10. The code calculates the average evaluation metrics for each model (Random Forest, Naive Bayes, and LSTM) across all folds by calling the ***calculate_averages*** function. Then, it creates a comparison table using pandas.DataFrame where each column corresponds to the averaged metrics of one model. The table is rounded to four decimal places and printed to show a comparison of the evaluation metrics for all models.

```
averages_rf = calculate_averages(all_evaluation_metrics_rf)
averages_nb = calculate_averages(all_evaluation_metrics_nb)
averages_lstm = calculate_averages(all_evaluation_metrics_lstm)

metrics_table = pd.DataFrame({
    "Random Forest": averages_rf,
    "Naive Bayes": averages_nb,
    "LSTM": averages_lstm,
})

metrics_table = metrics_table.round(4)
print("\033[1m" + "\nCOMPARISON OF EVALUATION METRICS\n" + "\033[0m")
print()
print(metrics_table.to_string())
print("\n\n\n")
```

11. The code computes the mean ROC curves for Random Forest, Naive Bayes, and LSTM using build_roc, then plots them with their AUC values. A reference diagonal line is included, and the graph is labeled with "ROC Curve" as the title.

```
mean_fpr_rf, mean_tpr_rf = build_roc(fpr_rf, tpr_rf)
mean_fpr_nb, mean_tpr_nb = build_roc(fpr_nb, tpr_nb)
mean_fpr_lstm, mean_tpr_lstm = build_roc(fpr_lstm, tpr_lstm)

#Plotting the ROC Curve
plt.figure(figsize=(10, 8))
plt.plot(mean_fpr_rf, mean_tpr_rf, color='b', label=f'Random Forest (AUC = {np.mean(auc_rf):.2f})')
plt.plot(mean_fpr_nb, mean_tpr_nb, color='g', label=f'Naive Bayes (AUC = {np.mean(auc_nb):.2f})')
plt.plot(mean_fpr_lstm, mean_tpr_lstm, color='r', label=f'LSTM (AUC = {np.mean(auc_lstm):.2f})')
plt.plot([0, 1], [0, 1], linestyle='--', color='k')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve')
plt.legend(loc='lower right')
plt.show()
```

Below are the screenshots of the outputs:

```
10 FOLD METRICS FOR RANDOM FOREST


Random Forest - Fold 1
Positive Examples (P): 40
Negative Examples (N): 17
True Positive Rate (TPR): 0.975
True Negative Rate (TNR): 0.9411764705882353
False Positive Rate (FPR): 0.058823529411764705
False Negative Rate (FNR): 0.025
Accuracy: 0.9649122807017544
Precision: 0.975
Recall: 0.975
F1-Score: 0.975
AUC: 0.9941176470588236
Balanced Accuracy (BACC): 0.9580882352941176
True Skill Statistics (TSS): 0.9161764705882353
Heidke Skill Score (HSS): 0.9161764705882353
Brier Score (BS): 0.026831578947368425
Brier Skill Score (BSS): 0.871800294117647
Error Rate (ERR): 0.03508771929824561

Random Forest - Fold 2
Positive Examples (P): 31
Negative Examples (N): 26
True Positive Rate (TPR): 1.0
True Negative Rate (TNR): 0.9230769230769231
False Positive Rate (FPR): 0.07692307692307693
False Negative Rate (FNR): 0.0
Accuracy: 0.9649122807017544
Precision: 0.9393939393939394
Recall: 1.0
F1-Score: 0.96875
AUC: 0.9975186104218362
Balanced Accuracy (BACC): 0.9615384615384616
True Skill Statistics (TSS): 0.9230769230769231
Heidke Skill Score (HSS): 0.9288389513108615
Brier Score (BS): 0.025773684210526313
Brier Skill Score (BSS): 0.8961058312655087
Error Rate (ERR): 0.03508771929824561
```

```
Random Forest - Fold 3
Positive Examples (P): 37
Negative Examples (N): 20
True Positive Rate (TPR): 1.0
True Negative Rate (TNR): 0.95
False Positive Rate (FPR): 0.05
False Negative Rate (FNR): 0.0
Accuracy: 0.9824561403508771
Precision: 0.9736842105263158
Recall: 1.0
F1-Score: 0.9866666666666667
AUC: 0.9993243243243243
Balanced Accuracy (BACC): 0.975
True Skill Statistics (TSS): 0.95
Heidke Skill Score (HSS): 0.961038961038961
Brier Score (BS): 0.021975438596491228
Brier Skill Score (BSS): 0.903515945945946
Error Rate (ERR): 0.017543859649122806

Random Forest - Fold 4
Positive Examples (P): 40
Negative Examples (N): 17
True Positive Rate (TPR): 0.975
True Negative Rate (TNR): 0.9411764705882353
False Positive Rate (FPR): 0.058823529411764705
False Negative Rate (FNR): 0.025
Accuracy: 0.9649122807017544
Precision: 0.975
Recall: 0.975
F1-Score: 0.975
AUC: 0.9970588235294118
Balanced Accuracy (BACC): 0.9580882352941176
True Skill Statistics (TSS): 0.9161764705882353
Heidke Skill Score (HSS): 0.9161764705882353
Brier Score (BS): 0.019733333333333335
Brier Skill Score (BSS): 0.9057152941176471
Error Rate (ERR): 0.03508771929824561


Random Forest - Fold 5
Positive Examples (P): 39
Negative Examples (N): 18
True Positive Rate (TPR): 0.9743589743589743
True Negative Rate (TNR): 0.9444444444444444
False Positive Rate (FPR): 0.05555555555555555
False Negative Rate (FNR): 0.02564102564102564
Accuracy: 0.9649122807017544
Precision: 0.9743589743589743
Recall: 0.9743589743589743
F1-Score: 0.9743589743589743
AUC: 0.9957264957264957
Balanced Accuracy (BACC): 0.9594017094017093
True Skill Statistics (TSS): 0.9188034188034188
Heidke Skill Score (HSS): 0.9188034188034188
Brier Score (BS): 0.034710526315789476
Brier Skill Score (BSS): 0.8393525641025641
Error Rate (ERR): 0.03508771929824561

Random Forest - Fold 6
Positive Examples (P): 32
Negative Examples (N): 25
True Positive Rate (TPR): 0.96875
True Negative Rate (TNR): 0.92
False Positive Rate (FPR): 0.08
False Negative Rate (FNR): 0.03125
Accuracy: 0.9473684210526315
Precision: 0.9393939393939394
Recall: 0.96875
F1-Score: 0.9538461538461539
AUC: 0.98
Balanced Accuracy (BACC): 0.944375
True Skill Statistics (TSS): 0.88875
Heidke Skill Score (HSS): 0.8926553672316384
Brier Score (BS): 0.04646666666666667
Brier Skill Score (BSS): 0.81128725
Error Rate (ERR): 0.05263157894736842
```

```
Random Forest - Fold 7
Positive Examples (P): 40
Negative Examples (N): 17
True Positive Rate (TPR): 0.975
True Negative Rate (TNR): 0.9411764705882353
False Positive Rate (FPR): 0.058823529411764705
False Negative Rate (FNR): 0.025
Accuracy: 0.9649122807017544
Precision: 0.975
Recall: 0.975
F1-Score: 0.975
AUC: 0.9955882352941176
Balanced Accuracy (BACC): 0.9580882352941176
True Skill Statistics (TSS): 0.9161764705882353
Heidke Skill Score (HSS): 0.9161764705882353
Brier Score (BS): 0.026799999999999997
Brier Skill Score (BSS): 0.8719511764705883
Error Rate (ERR): 0.03508771929824561

Random Forest - Fold 8
Positive Examples (P): 31
Negative Examples (N): 26
True Positive Rate (TPR): 0.967741935483871
True Negative Rate (TNR): 0.9230769230769231
False Positive Rate (FPR): 0.07692307692307693
False Negative Rate (FNR): 0.03225806451612903
Accuracy: 0.9473684210526315
Precision: 0.9375
Recall: 0.967741935483871
F1-Score: 0.9523809523809523
AUC: 0.9962779156327544
Balanced Accuracy (BACC): 0.9454094292803971
True Skill Statistics (TSS): 0.8908188585607941
Heidke Skill Score (HSS): 0.8935905413814561
Brier Score (BS): 0.029673684210526317
Brier Skill Score (BSS): 0.8803848635235731
Error Rate (ERR): 0.05263157894736842


Random Forest - Fold 9
Positive Examples (P): 30
Negative Examples (N): 27
True Positive Rate (TPR): 1.0
True Negative Rate (TNR): 0.9259259259259259
False Positive Rate (FPR): 0.07407407407407407
False Negative Rate (FNR): 0.0
Accuracy: 0.9649122807017544
Precision: 0.9375
Recall: 1.0
F1-Score: 0.967741935483871
AUC: 0.9820987654320988
Balanced Accuracy (BACC): 0.962962962962963
True Skill Statistics (TSS): 0.9259259259259259
Heidke Skill Score (HSS): 0.929368029739777
Brier Score (BS): 0.04284561403508772
Brier Skill Score (BSS): 0.8281414814814815
Error Rate (ERR): 0.03508771929824561

Random Forest - Fold 10
Positive Examples (P): 37
Negative Examples (N): 19
True Positive Rate (TPR): 0.972972972972973
True Negative Rate (TNR): 0.9473684210526315
False Positive Rate (FPR): 0.05263157894736842
False Negative Rate (FNR): 0.02702702702702703
Accuracy: 0.9642857142857143
Precision: 0.972972972972973
Recall: 0.972972972972973
F1-Score: 0.972972972972973
AUC: 0.988620199146515
Balanced Accuracy (BACC): 0.9601706970128023
True Skill Statistics (TSS): 0.9203413940256047
Heidke Skill Score (HSS): 0.9203413940256046
Brier Score (BS): 0.04114285714285714
Brier Skill Score (BSS): 0.8164665718349929
Error Rate (ERR): 0.03571428571428571
```

10 FOLD METRICS FOR NAIVE BAYES

Naive Bayes - Fold 1
Positive Examples (P): 40
Negative Examples (N): 17
True Positive Rate (TPR): 1.0
True Negative Rate (TNR): 0.9411764705882353
False Positive Rate (FPR): 0.058823529411764705
False Negative Rate (FNR): 0.0
Accuracy: 0.9824561403508771
Precision: 0.975609756097561
Recall: 1.0
F1-Score: 0.9876543209876543
AUC: 0.9955882352941177
Balanced Accuracy (BACC): 0.9705882352941176
True Skill Statistics (TSS): 0.9411764705882353
Heidke Skill Score (HSS): 0.9573672400897532
Brier Score (BS): 0.018146028503391105
Brier Skill Score (BSS): 0.9132993432242387
Error Rate (ERR): 0.017543859649122806

Naive Bayes - Fold 2
Positive Examples (P): 31
Negative Examples (N): 26
True Positive Rate (TPR): 1.0
True Negative Rate (TNR): 0.9230769230769231
False Positive Rate (FPR): 0.07692307692307693
False Negative Rate (FNR): 0.0
Accuracy: 0.9649122807017544
Precision: 0.9393939393939394
Recall: 1.0
F1-Score: 0.96875
AUC: 1.0
Balanced Accuracy (BACC): 0.9615384615384616
True Skill Statistics (TSS): 0.9230769230769231
Heidke Skill Score (HSS): 0.9288389513108615
Brier Score (BS): 0.03675828842710673
Brier Skill Score (BSS): 0.8518267008688961
Error Rate (ERR): 0.03508771929824561

Naive Bayes - Fold 3
Positive Examples (P): 37
Negative Examples (N): 20
True Positive Rate (TPR): 0.8918918918918919
True Negative Rate (TNR): 0.85
False Positive Rate (FPR): 0.15
False Negative Rate (FNR): 0.10810810810810811
Accuracy: 0.8771929824561403
Precision: 0.9166666666666666
Recall: 0.8918918918918919
F1-Score: 0.9041095890410958
AUC: 0.9743243243243244
Balanced Accuracy (BACC): 0.8709459459459459
True Skill Statistics (TSS): 0.7418918918918919
Heidke Skill Score (HSS): 0.7334669338677354
Brier Score (BS): 0.11030529480239191
Brier Skill Score (BSS): 0.5157001313338225
Error Rate (ERR): 0.12280701754385964

Naive Bayes - Fold 4
Positive Examples (P): 40
Negative Examples (N): 17
True Positive Rate (TPR): 0.975
True Negative Rate (TNR): 0.9411764705882353
False Positive Rate (FPR): 0.058823529411764705
False Negative Rate (FNR): 0.025
Accuracy: 0.9649122807017544
Precision: 0.975
Recall: 0.975
F1-Score: 0.975
AUC: 0.9970588235294118
Balanced Accuracy (BACC): 0.9580882352941176
True Skill Statistics (TSS): 0.9161764705882353
Heidke Skill Score (HSS): 0.9161764705882353
Brier Score (BS): 0.03632014672971071
Brier Skill Score (BSS): 0.8264644754046616
Error Rate (ERR): 0.03508771929824561

Naive Bayes - Fold 5
Positive Examples (P): 39
Negative Examples (N): 18
True Positive Rate (TPR): 0.9487179487179487
True Negative Rate (TNR): 0.8888888888888888
False Positive Rate (FPR): 0.1111111111111111
False Negative Rate (FNR): 0.05128205128205128
Accuracy: 0.9298245614035088
Precision: 0.9487179487179487
Recall: 0.9487179487179487
F1-Score: 0.9487179487179487
AUC: 0.9814814814814815
Balanced Accuracy (BACC): 0.9188034188034188
True Skill Statistics (TSS): 0.8376068376068375
Heidke Skill Score (HSS): 0.8376068376068376
Brier Score (BS): 0.07062257372185683
Brier Skill Score (BSS): 0.6731442421334575
Error Rate (ERR): 0.07017543859649122

Naive Bayes - Fold 6
Positive Examples (P): 32
Negative Examples (N): 25
True Positive Rate (TPR): 0.96875
True Negative Rate (TNR): 0.96
False Positive Rate (FPR): 0.04
False Negative Rate (FNR): 0.03125
Accuracy: 0.9649122807017544
Precision: 0.96875
Recall: 0.96875
F1-Score: 0.96875
AUC: 0.99
Balanced Accuracy (BACC): 0.964375
True Skill Statistics (TSS): 0.92875
Heidke Skill Score (HSS): 0.92875
Brier Score (BS): 0.02385864884368421
Brier Skill Score (BSS): 0.9031040623835875
Error Rate (ERR): 0.03508771929824561

Naive Bayes - Fold 7
Positive Examples (P): 40
Negative Examples (N): 17
True Positive Rate (TPR): 0.95
True Negative Rate (TNR): 0.8823529411764706
False Positive Rate (FPR): 0.11764705882352941
False Negative Rate (FNR): 0.05
Accuracy: 0.9298245614035088
Precision: 0.95
Recall: 0.95
F1-Score: 0.95
AUC: 0.9897058823529412
Balanced Accuracy (BACC): 0.9161764705882353
True Skill Statistics (TSS): 0.8323529411764705
Heidke Skill Score (HSS): 0.8323529411764706
Brier Score (BS): 0.06903321779536405
Brier Skill Score (BSS): 0.6701633461512679
Error Rate (ERR): 0.07017543859649122

Naive Bayes - Fold 8
Positive Examples (P): 31
Negative Examples (N): 26
True Positive Rate (TPR): 1.0
True Negative Rate (TNR): 0.8461538461538461
False Positive Rate (FPR): 0.15384615384615385
False Negative Rate (FNR): 0.0
Accuracy: 0.9298245614035088
Precision: 0.8857142857142857
Recall: 1.0
F1-Score: 0.9393939393939394
AUC: 0.9913151364764268
Balanced Accuracy (BACC): 0.9230769230769231
True Skill Statistics (TSS): 0.8461538461538461
Heidke Skill Score (HSS): 0.8567839195979899
Brier Score (BS): 0.06183711573790027
Brier Skill Score (BSS): 0.7507335123667023
Error Rate (ERR): 0.07017543859649122

```
Naive Bayes - Fold 9
Positive Examples (P): 30
Negative Examples (N): 27
True Positive Rate (TPR): 0.9666666666666667
True Negative Rate (TNR): 0.9259259259259259
False Positive Rate (FPR): 0.07407407407407407
False Negative Rate (FNR): 0.03333333333333333
Accuracy: 0.9473684210526315
Precision: 0.9354838709677419
Recall: 0.9666666666666667
F1-Score: 0.9508196721311475
AUC: 0.9814814814814815
Balanced Accuracy (BACC): 0.9462962962962963
True Skill Statistics (TSS): 0.8925925925925926
Heidke Skill Score (HSS): 0.8942486085343229
Brier Score (BS): 0.04598944464411785
Brier Skill Score (BSS): 0.8155312275941495
Error Rate (ERR): 0.05263157894736842

Naive Bayes - Fold 10
Positive Examples (P): 37
Negative Examples (N): 19
True Positive Rate (TPR): 0.972972972972973
True Negative Rate (TNR): 0.7368421052631579
False Positive Rate (FPR): 0.2631578947368421
False Negative Rate (FNR): 0.02702702702702703
Accuracy: 0.8928571428571429
Precision: 0.8780487804878049
Recall: 0.972972972972973
F1-Score: 0.9230769230769231
AUC: 0.9786628733997155
Balanced Accuracy (BACC): 0.8549075391180654
True Skill Statistics (TSS): 0.709815078236131
Heidke Skill Score (HSS): 0.7481259370314842
Brier Score (BS): 0.09627440544310772
Brier Skill Score (BSS): 0.5705312439977442
Error Rate (ERR): 0.10714285714285714
```

10 FOLD METRICS FOR LSTM

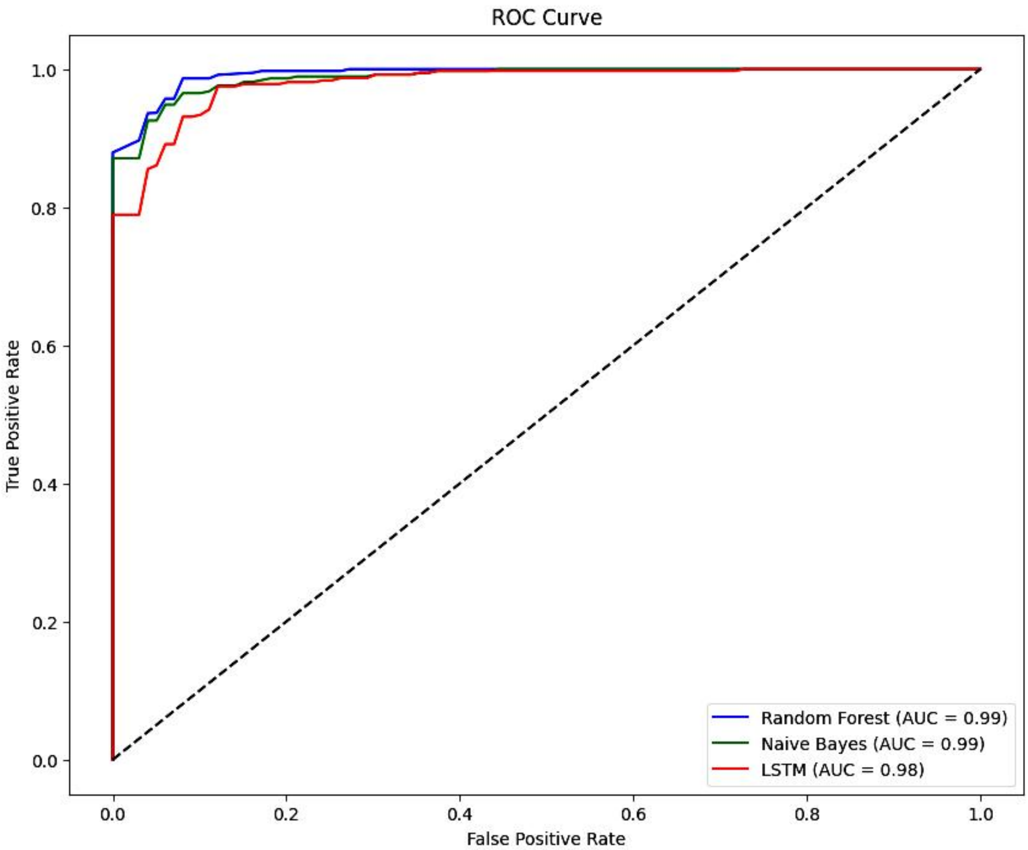
```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 138ms/step
LSTM - Fold 1
Positive Examples (P): 40
Negative Examples (N): 17
True Positive Rate (TPR): 0.95
True Negative Rate (TNR): 0.8823529411764706
False Positive Rate (FPR): 0.11764705882352941
False Negative Rate (FNR): 0.05
Accuracy: 0.9298245614035088
Precision: 0.95
Recall: 0.95
F1-Score: 0.95
AUC: 0.9838235294117647
Balanced Accuracy (BACC): 0.9161764705882353
True Skill Statistics (TSS): 0.8323529411764705
Heidke Skill Score (HSS): 0.8323529411764706
Brier Score (BS): 0.04934906032849617
Brier Skill Score (BSS): 0.7642130926363471
Error Rate (ERR): 0.07017543859649122

2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 132ms/step
LSTM - Fold 2
Positive Examples (P): 31
Negative Examples (N): 26
True Positive Rate (TPR): 1.0
True Negative Rate (TNR): 0.8461538461538461
False Positive Rate (FPR): 0.15384615384615385
False Negative Rate (FNR): 0.0
Accuracy: 0.9298245614035088
Precision: 0.8857142857142857
Recall: 1.0
F1-Score: 0.9393939393939394
AUC: 0.9937965260545906
Balanced Accuracy (BACC): 0.9230769230769231
True Skill Statistics (TSS): 0.8461538461538461
Heidke Skill Score (HSS): 0.8567839195979899
Brier Score (BS): 0.048163068609056656
Brier Skill Score (BSS): 0.8058538338575372
Error Rate (ERR): 0.07017543859649122
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 134ms/step
LSTM - Fold 3
Positive Examples (P): 37
Negative Examples (N): 20
True Positive Rate (TPR): 0.918918918918919
True Negative Rate (TNR): 0.9
False Positive Rate (FPR): 0.1
False Negative Rate (FNR): 0.08108108108108109
Accuracy: 0.9122807017543859
Precision: 0.9444444444444444
Recall: 0.918918918918919
F1-Score: 0.9315068493150684
AUC: 0.981081081081081
Balanced Accuracy (BACC): 0.9094594594594595
True Skill Statistics (TSS): 0.818918918918919
Heidke Skill Score (HSS): 0.8096192384769539
Brier Score (BS): 0.05657523393913514
Brier Skill Score (BSS): 0.7516041417996621
Error Rate (ERR): 0.08771929824561403

2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 132ms/step
LSTM - Fold 4
Positive Examples (P): 40
Negative Examples (N): 17
True Positive Rate (TPR): 0.95
True Negative Rate (TNR): 0.9411764705882353
False Positive Rate (FPR): 0.058823529411764705
False Negative Rate (FNR): 0.05
Accuracy: 0.9473684210526315
Precision: 0.9743589743589743
Recall: 0.95
F1-Score: 0.9620253164556962
AUC: 0.9852941176470588
Balanced Accuracy (BACC): 0.9455882352941176
True Skill Statistics (TSS): 0.8911764705882352
Heidke Skill Score (HSS): 0.8763557483731019
Brier Score (BS): 0.04331746763367835
Brier Skill Score (BSS): 0.7930316877326162
Error Rate (ERR): 0.05263157894736842

2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 168ms/step
LSTM - Fold 5
Positive Examples (P): 39
Negative Examples (N): 18
True Positive Rate (TPR): 0.9743589743589743
True Negative Rate (TNR): 0.8888888888888888
False Positive Rate (FPR): 0.1111111111111111
False Negative Rate (FNR): 0.02564102564102564
Accuracy: 0.9473684210526315
Precision: 0.95
Recall: 0.9743589743589743
F1-Score: 0.9620253164556962
AUC: 0.972934472934473
Balanced Accuracy (BACC): 0.9316239316239316
True Skill Statistics (TSS): 0.8632478632478633
Heidke Skill Score (HSS): 0.8763557483731019
Brier Score (BS): 0.0498097191150152
Brier Skill Score (BSS): 0.7694704025574296
Error Rate (ERR): 0.05263157894736842

2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 130ms/step
LSTM - Fold 6
Positive Examples (P): 32
Negative Examples (N): 25
True Positive Rate (TPR): 1.0
True Negative Rate (TNR): 0.88
False Positive Rate (FPR): 0.12
False Negative Rate (FNR): 0.0
Accuracy: 0.9473684210526315
Precision: 0.9142857142857143
Recall: 1.0
F1-Score: 0.9552238805970149
AUC: 0.97125
Balanced Accuracy (BACC): 0.94
True Skill Statistics (TSS): 0.88
Heidke Skill Score (HSS): 0.8917036098796707
Brier Score (BS): 0.05692696393753381
Brier Skill Score (BSS): 0.7688053677086908
Error Rate (ERR): 0.05263157894736842
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 130ms/step
LSTM - Fold 7
Positive Examples (P): 40
Negative Examples (N): 17
True Positive Rate (TPR): 0.975
True Negative Rate (TNR): 0.8823529411764706
False Positive Rate (FPR): 0.11764705882352941
False Negative Rate (FNR): 0.025
Accuracy: 0.9473684210526315
Precision: 0.9512195121951219
Recall: 0.975
F1-Score: 0.9629629629629629
AUC: 0.9794117647058822
Balanced Accuracy (BACC): 0.9286764705882353
True Skill Statistics (TSS): 0.8573529411764705
Heidke Skill Score (HSS): 0.8721017202692596
Brier Score (BS): 0.0510191932108073
Brier Skill Score (BSS): 0.7562332959677751
Error Rate (ERR): 0.05263157894736842

2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 131ms/step
LSTM - Fold 8
Positive Examples (P): 31
Negative Examples (N): 26
True Positive Rate (TPR): 0.9354838709677419
True Negative Rate (TNR): 0.9230769230769231
False Positive Rate (FPR): 0.07692307692307693
False Negative Rate (FNR): 0.06451612903225806
Accuracy: 0.9298245614035088
Precision: 0.9354838709677419
Recall: 0.9354838709677419
F1-Score: 0.9354838709677419
AUC: 0.9851116625310173
Balanced Accuracy (BACC): 0.9292803970223324
True Skill Statistics (TSS): 0.8585607940446649
Heidke Skill Score (HSS): 0.858560794044665
Brier Score (BS): 0.04852698515666815
Brier Skill Score (BSS): 0.8043868799329841
Error Rate (ERR): 0.07017543859649122

2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 138ms/step
LSTM - Fold 9
Positive Examples (P): 30
Negative Examples (N): 27
True Positive Rate (TPR): 0.8666666666666667
True Negative Rate (TNR): 0.9259259259259259
False Positive Rate (FPR): 0.07407407407407407
False Negative Rate (FNR): 0.13333333333333333
Accuracy: 0.8947368421052632
Precision: 0.9285714285714286
Recall: 0.8666666666666667
F1-Score: 0.896551724137931
AUC: 0.9617283950617285
Balanced Accuracy (BACC): 0.8962962962962964
True Skill Statistics (TSS): 0.7925925925925926
Heidke Skill Score (HSS): 0.7896678966789668
Brier Score (BS): 0.08034042333549174
Brier Skill Score (BSS): 0.6777456352876388
Error Rate (ERR): 0.10526315789473684

2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 131ms/step
LSTM - Fold 10
Positive Examples (P): 37
Negative Examples (N): 19
True Positive Rate (TPR): 1.0
True Negative Rate (TNR): 0.631578947368421
False Positive Rate (FPR): 0.3684210526315789
False Negative Rate (FNR): 0.0
Accuracy: 0.875
Precision: 0.8409090909090909
Recall: 1.0
F1-Score: 0.9135802469135802
AUC: 0.9815078236130867
Balanced Accuracy (BACC): 0.8157894736842105
True Skill Statistics (TSS): 0.631578947368421
Heidke Skill Score (HSS): 0.69375
Brier Score (BS): 0.11379763642458551
Brier Skill Score (BSS): 0.49236217947724015
Error Rate (ERR): 0.125
```

COMPARISON OF EVALUATION METRICS

	Random Forest	Naive Bayes	LSTM
Positive Examples (P)	35.7000	35.7000	35.7000
Negative Examples (N)	21.2000	21.2000	21.2000
True Positive Rate (TPR)	0.9809	0.9674	0.9570
True Negative Rate (TNR)	0.9357	0.8896	0.8702
False Positive Rate (FPR)	0.0643	0.1104	0.1298
False Negative Rate (FNR)	0.0191	0.0326	0.0430
Accuracy	0.9631	0.9384	0.9261
Precision	0.9600	0.9373	0.9275
Recall	0.9809	0.9674	0.9570
F1-Score	0.9702	0.9516	0.9409
AUC	0.9926	0.9880	0.9796
Balanced Accuracy (BACC)	0.9583	0.9285	0.9136
True Skill Statistics (TSS)	0.9166	0.8570	0.8272
Heidke Skill Score (HSS)	0.9193	0.8634	0.8357
Brier Score (BS)	0.0316	0.0569	0.0598
Brier Skill Score (BSS)	0.8625	0.7490	0.7384
Error Rate (ERR)	0.0369	0.0616	0.0739



Conclusion:

The Random Forest model performs the best overall, with the highest True Positive Rate (TPR) (0.9809) and True Negative Rate (TNR) (0.9357), Accuracy (0.9631), Precision (0.9600), F1-Score (0.9702), Balanced Accuracy (BACC) (0.9583), True Skill Statistics (TSS) (0.9166), Heidke Skill Score (HSS) (0.9193), and the least Error Rate (ERR) (0.0369) along with the highest AUC (0.9926). However, LSTM outperforms both models in the Brier Score (BSS) (0.0598).

GitHub Link:

https://github.com/SharathShankarRathakrishnan/Rathakrishnan_SharathShankar_finaltermproj