Data Mining Final Project - Anish Panicker (ap2938)

Introduction

This project aims to implement three classification algorithms;Random Forest, Naive Bayes, and Long Short-Term Memory (LSTM) ,to predict the likelihood of an individual being a drinker. Utilizing diagnostic measurements collected from the National Health Insurance Service in Korea .The primary objectives of the dataset are the analysis of body signals and the classification of smoking and drinking behaviors. The performance of each model is evaluated using 10-fold cross-validation, focusing on metrics such as accuracy, precision, and Area Under the ROC Curve (AUC).

Dataset Description:

The dataset used in this project includes various health-related features and behavioral indicators of individuals. The main goal is to predict the target variable DRK_YN, which shows whether a person is a drinker (Y) or not (N). The dataset contains the following columns:

Demographics

- **sex:** Gender of the individual (Male/Female)
- **age:** Age in years
- **height:** Height in centimeters
- **weight:** Weight in kilograms
- **waistline:** Waist circumference in centimeters

Health Metrics

- **sight_left, sight_right:** Visual acuity measurements for the left and right eyes
- **hear_left, hear_right:** Hearing ability measurements for the left and right ears
- **SBP, DBP:** Systolic and Diastolic Blood Pressure
- **BLDS:** Blood Sugar Level
- **tot_chole, HDL_chole, LDL_chole, triglyceride:** Various cholesterol levels
- **hemoglobin:** Hemoglobin levels in the blood
- **urine_protein:** Presence of protein in urine
- **serum_creatinine:** Serum creatinine levels
- **SGOT_AST, SGOT_ALT, gamma_GTP:** Levels of liver enzymes

Behavioral Indicators

- **SMK_stat_type_cd:** Smoking status code

Target Variable

- **DRK_YN:** Drinker status (Y for Yes, N for No)

Data Mining Final Project - Anish Panicker (ap2938)

```
First few rows of the dataset:
    sex  age  height  weight  waistline  sight_left  sight_right  hear_left  \
0  Male   35     170      75       90.0         1.0          1.0        1.0
1  Male   30     180      80       89.0         0.9          1.2        1.0
2  Male   40     165      75       91.0         1.2          1.5        1.0
3  Male   50     175      80       91.0         1.5          1.2        1.0
4  Male   50     165      60       80.0         1.0          1.2        1.0

   hear_right    SBP  ...  LDL_chole  triglyceride  hemoglobin  urine_protein  \
0         1.0  120.0  ...      126.0          92.0        17.1            1.0
1         1.0  130.0  ...      148.0         121.0        15.8            1.0
2         1.0  120.0  ...       74.0         104.0        15.8            1.0
3         1.0  145.0  ...      104.0         106.0        17.6            1.0
4         1.0  138.0  ...      117.0         104.0        13.8            1.0

   serum_creatinine  SGOT_AST  SGOT_ALT  gamma_GTP  SMK_stat_type_cd  DRK_YN
0               1.0      21.0      35.0       40.0               1.0       Y
1               0.9      20.0      36.0       27.0               3.0       N
2               0.9      47.0      32.0       68.0               1.0       N
3               1.1      29.0      34.0       18.0               1.0       N
4               0.8      19.0      12.0       25.0               1.0       N

[5 rows x 24 columns]
```

Sample Data

Data Preprocessing

The dataset obtained from Kaggle was already well-structured and free of missing values. However, I conducted additional checks and performed some preprocessing to understand the data better for training purposes.

Encoding Categorical Variables

The dataset contains categorical variables that need to be converted into numerical values for the machine learning models to process them effectively. Specifically, the sex and DRK_YN columns were encoded using LabelEncoder from the sklearn.preprocessing module.

- **Sex Encoding:**
  - **Male:** Encoded as 1
  - **Female:** Encoded as 0
- **DRK_YN Encoding:**
  - **Y (Yes):** Encoded as 1
  - **N (No):** Encoded as 0

Data Mining Final Project - Anish Panicker (ap2938)

```
Encoded 'sex' and 'DRK_YN' columns:
   sex  DRK_YN
0    1       1
1    1       0
2    1       0
3    1       0
4    1       0
```

Data Encoding

Feature Selection

After encoding the categorical variables, the next step was to identify and select the relevant features that will be used to predict the target variable DRK_YN. The selected feature columns include:

- **Demographics:**
  - sex, age, height, weight, waistline
- **Health Metrics:**
  - sight_left, sight_right, hear_left, hear_right
  - SBP, DBP, BLDS, tot_chole
  - HDL_chole, LDL_chole, triglyceride
  - hemoglobin, urine_protein, serum_creatinine
  - SGOT_AST, SGOT_ALT, gamma_GTP
- **Behavioral Indicators:**
  - SMK_stat_type_cd

These features encompass a range of demographic information, health-related measurements, and behavioral indicators, providing a comprehensive basis for predicting drinking behavior.

Handling Missing Values

Upon inspecting the dataset, it was determined that there are no missing values. and there was no need for data cleaning in this regard.

Feature Scaling

To ensure that all features contribute equally to the model training process and to improve the convergence speed of the algorithms, feature scaling was performed using StandardScaler from the sklearn.preprocessing module. This scaler standardizes the features by removing the mean and scaling to unit variance.

Data Mining Final Project - Anish Panicker (ap2938)

The scaling process transforms the data so that each feature has a mean of zero and a standard deviation of one. This normalization is particularly important for algorithms like LSTM, which are sensitive to the scale of input data.

Final Preparation

After encoding, selecting, and scaling the features, the data was converted into NumPy arrays to make it compatible with the machine learning and deep learning models used in this project. The feature matrix X and the target vector y were extracted and transformed

```
# Extract features and target
X = df[feature_columns]
y = df[target]

# Handles missing values
imputer = SimpleImputer(strategy='mean')
X_imputed = imputer.fit_transform(X)

# Feature Scaling
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X_imputed)

# Convert to numpy arrays
X_scaled = np.array(X_scaled)
y = np.array(y)
```

Code for data preprocessing

Data Mining Final Project - Anish Panicker (ap2938)

Evaluation Metrics

To assess the performance of the classification models, I used several evaluation metrics derived from the confusion matrix. These metrics provide a comprehensive understanding of how well the models are predicting drinker status. Below is a description of each metric and its significance:

Confusion Matrix Components

1. **True Positives (TP):**
   The number of individuals correctly identified as drinkers.
2. **True Negatives (TN):**
   The number of individuals correctly identified as non-drinkers.
3. **False Positives (FP):**
   The number of individuals incorrectly identified as drinkers when they are not.
4. **False Negatives (FN):**
   The number of individuals incorrectly identified as non-drinkers when they actually are drinkers.

Formulas of Metrics used

To evaluate the performance of the classification models, I used several derived metrics based on the confusion matrix.

1. **True Positive Rate (TPR) / Sensitivity:**
   - **Description:** Measures the proportion of actual drinkers that were correctly identified.
   - **Formula:**
     TPR = TP / (TP + FN)
2. **True Negative Rate (TNR) / Specificity:**
   - **Description:** Measures the proportion of actual non-drinkers that were correctly identified.
   - **Formula:**
     TNR = TN / (TN + FP)
3. **False Positive Rate (FPR):**
   - **Description:** Indicates the proportion of non-drinkers incorrectly classified as drinkers.
   - **Formula:**
     FPR = FP / (TN + FP)
4. **False Negative Rate (FNR):**
   - **Description:** Indicates the proportion of drinkers incorrectly classified as non-drinkers.
   - **Formula:**
     FNR = FN / (TP + FN)
5. **Precision:**
   - **Description:** Measures the accuracy of the drinker predictions. It shows how many of the individuals predicted as drinkers are actually drinkers.

   - **Formula:**
     Precision = TP / (TP + FP)
6. **F1 Measure:**
   - **Description:** Combines precision and sensitivity into a single metric by taking their harmonic mean. It provides a balance between the two, especially useful when dealing with imbalanced datasets.
   - **Formula:**
     F1 Measure = 2 × (Precision × TPR) / (Precision + TPR)
7. **Accuracy:**
   - **Description:** Represents the overall correctness of the model by calculating the proportion of true results (both true positives and true negatives) among the total number of cases examined.
   - **Formula:**
     Accuracy = (TP + TN) / (TP + FP + FN + TN)
8. **Error Rate:**
   - **Description:** Shows the proportion of all incorrect predictions (both false positives and false negatives) out of the total predictions made.
   - **Formula:**
     Error Rate = (FP + FN) / (TP + FP + FN + TN)
9. **Balanced Accuracy (BACC):**
   - **Description:** Averages the true positive rate and true negative rate, providing a balanced measure even if the classes are imbalanced.
   - **Formula:**
     BACC = (TPR + TNR) / 2
10. **True Skill Statistic (TSS):**
    - **Description:** Reflects the model's ability to distinguish between classes by subtracting the false positive rate from the true positive rate.
    - **Formula:**
      TSS = TPR − FPR
11. **Heidke Skill Score (HSS):**
    - **Description:** Measures the accuracy of the predictions while accounting for the accuracy that would be expected by random chance.
    - **Formula:**
      HSS = [2 × (TP × TN − FP × FN)] / [(TP + FN) × (FN + TN) + (TP + FP) × (FP + TN)]
12. **Brier Score:**
    - **Description:** Evaluates the accuracy of probabilistic predictions by calculating the mean squared difference between the predicted probabilities and the actual outcomes. A lower Brier score indicates better calibration of the predicted probabilities.
    - **Formula:**
      Brier Score = $(1/N) \times \Sigma (f_i - o_i)^2$
      where f_i is the predicted probability for instance i, and o_i is the actual outcome (0 or 1).

Data Mining Final Project - Anish Panicker (ap2938)

Purpose of Metrics

These metrics collectively provide a detailed picture of the model's performance from different perspectives:

- **Sensitivity and Specificity** help understand how well the model identifies positive and negative cases.
- **Precision and F1 Measure** are crucial when the cost of false positives is high.
- **Accuracy and Error Rate** offer a general overview but can be misleading in imbalanced datasets.
- **Balanced Accuracy** ensures that both classes are treated equally, which is important in cases of class imbalance.
- **TSS and HSS** provide insights into the model's skill beyond random chance.
- **Brier Score** assesses the quality of the predicted probabilities, which is important for probabilistic interpretations.

```python
def calculate_metrics(y_true, y_pred, y_pred_prob):
    cm = confusion_matrix(y_true, y_pred, labels=[0,1])
    TP, FN = cm[1,1], cm[1,0]
    FP, TN = cm[0,1], cm[0,0]

    # Calculate Metrics
    TPR = TP / (TP + FN) if (TP + FN) > 0 else 0  # Sensitivity
    TNR = TN / (TN + FP) if (TN + FP) > 0 else 0  # Specificity
    FPR = FP / (TN + FP) if (TN + FP) > 0 else 0  # False Positive Rate
    FNR = FN / (TP + FN) if (TP + FN) > 0 else 0  # False Negative Rate
    Precision = TP / (TP + FP) if (TP + FP) > 0 else 0  # Precision
    F1_measure = 2 * TP / (2 * TP + FP + FN) if (2 * TP + FP + FN) > 0 else 0  # F1 Score
    Accuracy = (TP + TN) / (TP + FP + FN + TN) if (TP + FP + FN + TN) > 0 else 0  # Accuracy
    Error_rate = (FP + FN) / (TP + FP + FN + TN) if (TP + FP + FN + TN) > 0 else 0  # Error Rate
    BACC = (TPR + TNR) / 2 if (TPR + TNR) > 0 else 0  # Balanced Accuracy
    TSS = TPR - FPR if (TPR + FPR) > 0 else 0  # True Skill Statistic
    # Heidke Skill Score (HSS)
    denominator = ((TP + FN) * (FN + TN) + (TP + FP) * (FP + TN))
    numerator = 2 * (TP * TN - FP * FN)
    HSS = numerator / denominator if denominator != 0 else 0

    # Brier Score
    Brier_score = brier_score_loss(y_true, y_pred_prob)

    # Compile metrics into a dictionary
    metrics = {
        'TP': TP,
        'TN': TN,
        'FP': FP,
        'FN': FN,
        'TPR': TPR,
        'TNR': TNR,
        'FPR': FPR,
        'FNR': FNR,
        'Precision': Precision,
        'F1_measure': F1_measure,
        'Accuracy': Accuracy,
        'Error_rate': Error_rate,
        'BACC': BACC,
        'TSS': TSS,
        'HSS': HSS,
        'Brier_score': Brier_score
    }

    return metrics
```

Code for calculating metrics

Data Mining Final Project - Anish Panicker (ap2938)

Classification Algorithms

In this project, I implemented three different algorithms to classify individuals as drinkers or non-drinkers.

Random Forest

Random Forest is an ensemble learning method that builds multiple decision trees to improve prediction accuracy and control overfitting.

- **Number of Trees (n_estimators)**: 50
  I used 50 trees to balance performance and training time.
- **Maximum Depth (max_depth)**: 10
  Limiting the depth helps prevent the trees from becoming too complex.
- **Minimum Samples to Split (min_samples_split)**: 10
  A minimum of 10 samples is required to split a node, which reduces overfitting.
- **Minimum Samples per Leaf (min_samples_leaf)**: 5
  Each leaf node must have at least 5 samples to ensure generalization.
- **Class Weight**: Balanced
  This adjusts the weights inversely proportional to class frequencies to handle any imbalance in the dataset.
- **Random State**: 42
  Setting a random state ensures reproducibility of results.
- **Number of Jobs (n_jobs)**: -1
  Utilizes all available CPU cores to speed up training.
- **Verbose**: 0
  Disables verbose output during training for cleaner logs.

Naive Bayes.

- **Variant Used**: Gaussian Naive Bayes
  As it is suitable for continuous data by assuming a Gaussian distribution of the features.
- **Priors**: Calculated from Training Data
  The prior probabilities are determined based on the distribution of classes in the training set to handle class imbalance.

Long Short-Term Memory (LSTM).

The Deep learning algorithm i used in this project is LSTM

- **Model Architecture:**
  - **LSTM Layer**: 64 units with ReLU activation
    Captures complex patterns in the data.

    - **Dropout Layer**: 50% dropout rate
      Prevents overfitting by randomly dropping neurons during training.
    - **Dense Output Layer**: 1 unit with sigmoid activation
      Outputs probabilities for binary classification.
- **Training Parameters:**
    - **Epochs**: 5
      Limited to 5 epochs to reduce training time while still allowing the model to learn.
    - **Batch Size**: 16
      Smaller batch sizes help in faster convergence.
    - **Validation Split**: 20%
      Allocates 20% of the training data for validating the model during training.
    - **Callbacks**: Early Stopping with Patience of 2
      Stops training if the validation loss does not improve for 2 consecutive epochs, preventing overfitting.
    - **Class Weight**: Balanced
      Adjusts the weights to handle class imbalance based on the training data.

Training Results

Per-Fold Metrics

The classification performance was assessed using 10-fold cross-validation. The following metrics were recorded for each algorithm throughout the folds.

Data Mining Final Project - Anish Panicker (ap2938)

Fold 1

```
----- Metrics for All Algorithms in Iteration 1 -----
Metric               Random Forest   Naive Bayes     LSTM
TP                   36759.00        33601.00        36671.00
TN                   35189.00        34856.00        36365.00
FP                   14397.00        14730.00        13221.00
FN                   12790.00        15948.00        12878.00
TPR                  0.74            0.68            0.74
TNR                  0.71            0.70            0.73
FPR                  0.29            0.30            0.27
FNR                  0.26            0.32            0.26
Precision            0.72            0.70            0.74
F1_measure           0.73            0.69            0.74
Acc_by_package_fn    0.73            0.69            0.74
Error_rate           0.27            0.31            0.26
BACC                 0.73            0.69            0.74
TSS                  0.45            0.38            0.47
HSS                  0.45            0.38            0.47
Brier_score          0.18            0.26            0.17
AUC                  0.81            0.74            0.82
```

Fold2

```
----- Metrics for All Algorithms in Iteration 2 -----
Metric               Random Forest   Naive Bayes     LSTM
TP                   37008.00        33538.00        36541.00
TN                   35209.00        35331.00        36848.00
FP                   14377.00        14255.00        12738.00
FN                   12541.00        16011.00        13008.00
TPR                  0.75            0.68            0.74
TNR                  0.71            0.71            0.74
FPR                  0.29            0.29            0.26
FNR                  0.25            0.32            0.26
Precision            0.72            0.70            0.74
F1_measure           0.73            0.69            0.74
Acc_by_package_fn    0.73            0.69            0.74
Error_rate           0.27            0.31            0.26
BACC                 0.73            0.69            0.74
TSS                  0.46            0.39            0.48
HSS                  0.46            0.39            0.48
Brier_score          0.18            0.26            0.17
AUC                  0.81            0.75            0.82
```

Data Mining Final Project - Anish Panicker (ap2938)

Fold 3

```
----- Metrics for All Algorithms in Iteration 3 -----
Metric               Random Forest   Naive Bayes     LSTM
TP                   36979.00        33566.00        36984.00
TN                   35122.00        34950.00        36267.00
FP                   14464.00        14636.00        13319.00
FN                   12570.00        15983.00        12565.00
TPR                  0.75            0.68            0.75
TNR                  0.71            0.70            0.73
FPR                  0.29            0.30            0.27
FNR                  0.25            0.32            0.25
Precision            0.72            0.70            0.74
F1_measure           0.73            0.69            0.74
Acc_by_package_fn    0.73            0.69            0.74
Error_rate           0.27            0.31            0.26
BACC                 0.73            0.69            0.74
TSS                  0.45            0.38            0.48
HSS                  0.45            0.38            0.48
Brier_score          0.18            0.26            0.17
AUC                  0.81            0.74            0.82
```

Fold 4

```
----- Metrics for All Algorithms in Iteration 4 -----
Metric               Random Forest   Naive Bayes     LSTM
TP                   36969.00        33886.00        37604.00
TN                   35396.00        34807.00        35711.00
FP                   14190.00        14779.00        13875.00
FN                   12580.00        15663.00        11945.00
TPR                  0.75            0.68            0.76
TNR                  0.71            0.70            0.72
FPR                  0.29            0.30            0.28
FNR                  0.25            0.32            0.24
Precision            0.72            0.70            0.73
F1_measure           0.73            0.69            0.74
Acc_by_package_fn    0.73            0.69            0.74
Error_rate           0.27            0.31            0.26
BACC                 0.73            0.69            0.74
TSS                  0.46            0.39            0.48
HSS                  0.46            0.39            0.48
Brier_score          0.18            0.26            0.17
AUC                  0.81            0.74            0.82
```

Data Mining Final Project - Anish Panicker (ap2938)

Fold 5

```
----- Metrics for All Algorithms in Iteration 5 -----
Metric               Random Forest   Naive Bayes     LSTM
TP                   36788.00        33528.00        36573.00
TN                   35432.00        35144.00        36684.00
FP                   14154.00        14442.00        12902.00
FN                   12761.00        16021.00        12976.00
TPR                  0.74            0.68            0.74
TNR                  0.71            0.71            0.74
FPR                  0.29            0.29            0.26
FNR                  0.26            0.32            0.26
Precision            0.72            0.70            0.74
F1_measure           0.73            0.69            0.74
Acc_by_package_fn    0.73            0.69            0.74
Error_rate           0.27            0.31            0.26
BACC                 0.73            0.69            0.74
TSS                  0.46            0.39            0.48
HSS                  0.46            0.39            0.48
Brier_score          0.18            0.26            0.17
AUC                  0.81            0.74            0.82
```

Fold 6

```
----- Metrics for All Algorithms in Iteration 6 -----
Metric               Random Forest   Naive Bayes     LSTM
TP                   36732.00        33300.00        36209.00
TN                   35289.00        35234.00        36797.00
FP                   14297.00        14352.00        12789.00
FN                   12817.00        16249.00        13340.00
TPR                  0.74            0.67            0.73
TNR                  0.71            0.71            0.74
FPR                  0.29            0.29            0.26
FNR                  0.26            0.33            0.27
Precision            0.72            0.70            0.74
F1_measure           0.73            0.69            0.73
Acc_by_package_fn    0.73            0.69            0.74
Error_rate           0.27            0.31            0.26
BACC                 0.73            0.69            0.74
TSS                  0.45            0.38            0.47
HSS                  0.45            0.38            0.47
Brier_score          0.18            0.26            0.17
AUC                  0.81            0.74            0.82
```

Data Mining Final Project - Anish Panicker (ap2938)

Fold 7

```
----- Metrics for All Algorithms in Iteration 7 -----
Metric               Random Forest   Naive Bayes     LSTM
TP                   36619.00        33205.00        36968.00
TN                   35217.00        35236.00        36128.00
FP                   14368.00        14349.00        13457.00
FN                   12930.00        16344.00        12581.00
TPR                  0.74            0.67            0.75
TNR                  0.71            0.71            0.73
FPR                  0.29            0.29            0.27
FNR                  0.26            0.33            0.25
Precision            0.72            0.70            0.73
F1_measure           0.73            0.68            0.74
Acc_by_package_fn    0.72            0.69            0.74
Error_rate           0.28            0.31            0.26
BACC                 0.72            0.69            0.74
TSS                  0.45            0.38            0.47
HSS                  0.45            0.38            0.47
Brier_score          0.18            0.26            0.17
AUC                  0.81            0.74            0.82
```

Fold 8

```
----- Metrics for All Algorithms in Iteration 8 -----
Metric               Random Forest   Naive Bayes     LSTM
TP                   36702.00        33830.00        36801.00
TN                   35322.00        34872.00        36471.00
FP                   14263.00        14713.00        13114.00
FN                   12847.00        15719.00        12748.00
TPR                  0.74            0.68            0.74
TNR                  0.71            0.70            0.74
FPR                  0.29            0.30            0.26
FNR                  0.26            0.32            0.26
Precision            0.72            0.70            0.74
F1_measure           0.73            0.69            0.74
Acc_by_package_fn    0.73            0.69            0.74
Error_rate           0.27            0.31            0.26
BACC                 0.73            0.69            0.74
TSS                  0.45            0.39            0.48
HSS                  0.45            0.39            0.48
Brier_score          0.18            0.26            0.17
AUC                  0.81            0.74            0.82
```

Data Mining Final Project - Anish Panicker (ap2938)

Fold 9

```
----- Metrics for All Algorithms in Iteration 9 -----
Metric               Random Forest   Naive Bayes     LSTM
TP                   36879.00        33714.00        37400.00
TN                   35165.00        34956.00        35719.00
FP                   14421.00        14630.00        13867.00
FN                   12669.00        15834.00        12148.00
TPR                  0.74            0.68            0.75
TNR                  0.71            0.70            0.72
FPR                  0.29            0.30            0.28
FNR                  0.26            0.32            0.25
Precision            0.72            0.70            0.73
F1_measure           0.73            0.69            0.74
Acc_by_package_fn    0.73            0.69            0.74
Error_rate           0.27            0.31            0.26
BACC                 0.73            0.69            0.74
TSS                  0.45            0.39            0.48
HSS                  0.45            0.39            0.48
Brier_score          0.18            0.26            0.17
AUC                  0.81            0.74            0.82
```

Fold 10

```
----- Metrics for All Algorithms in Iteration 10 -----
Metric               Random Forest   Naive Bayes     LSTM
TP                   36765.00        33558.00        36863.00
TN                   35427.00        35378.00        36374.00
FP                   14159.00        14208.00        13212.00
FN                   12783.00        15990.00        12685.00
TPR                  0.74            0.68            0.74
TNR                  0.71            0.71            0.73
FPR                  0.29            0.29            0.27
FNR                  0.26            0.32            0.26
Precision            0.72            0.70            0.74
F1_measure           0.73            0.69            0.74
Acc_by_package_fn    0.73            0.70            0.74
Error_rate           0.27            0.30            0.26
BACC                 0.73            0.70            0.74
TSS                  0.46            0.39            0.48
HSS                  0.46            0.39            0.48
Brier_score          0.18            0.26            0.17
AUC                  0.81            0.74            0.82
```

Data Mining Final Project - Anish Panicker (ap2938)

Average Metrics Across Folds

After completing the 10-fold cross-validation, the average performance metrics for each algorithm are as follows:

```
----- Average Metrics for Each Algorithm Across 10 Folds -----
           Metric  Random Forest  Naive Bayes     LSTM
               TP       36820.00     33572.60 36861.40
               TN       35276.80     35076.40 36336.40
               FP       14309.00     14509.40 13249.40
               FN       12728.80     15976.20 12687.40
              TPR           0.74         0.68     0.74
              TNR           0.71         0.71     0.73
              FPR           0.29         0.29     0.27
              FNR           0.26         0.32     0.26
        Precision           0.72         0.70     0.74
       F1_measure           0.73         0.69     0.74
Acc_by_package_fn           0.73         0.69     0.74
       Error_rate           0.27         0.31     0.26
             BACC           0.73         0.69     0.74
              TSS           0.45         0.38     0.48
              HSS           0.45         0.38     0.48
      Brier_score           0.18         0.26     0.17
              AUC           0.81         0.74     0.82
```

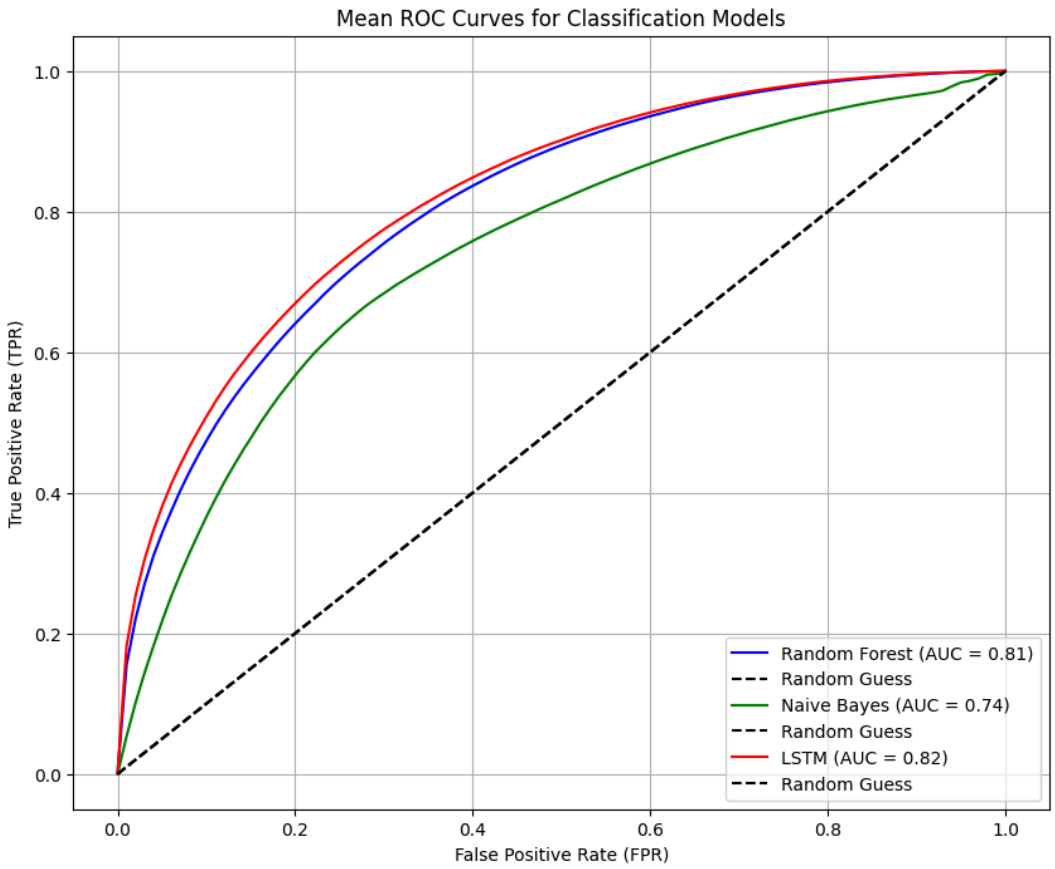
Which Model was better

Based on the evaluation metrics obtained from the 10-fold cross-validation, the LSTM (Long Short-Term Memory) model outperformed the Random Forest and Naive Bayes models.

Evaluation Metrics Comparison

1. **AUC (Area Under the ROC Curve)**:
    - **LSTM**: 0.82 (Highest)
    - Random Forest: 0.81
    - Naive Bayes: 0.74 The LSTM achieved the highest AUC, indicating it is better at distinguishing between classes.

Data Mining Final Project - Anish Panicker (ap2938)



2. **Accuracy**:
    - **LSTM**: 0.74 (Highest)
    - Random Forest: 0.73
    - Naive Bayes: 0.69 The LSTM had the highest overall accuracy, showing it made the most correct predictions overall.
3. **Precision**:
    - **LSTM**: 0.74 (Highest)
    - Random Forest: 0.72
    - Naive Bayes: 0.70 LSTM had the highest precision, meaning it was more accurate in predicting drinkers.
4. **F1 Score**:
    - **LSTM**: 0.74 (Highest)
    - Random Forest: 0.73
    - Naive Bayes: 0.69 The LSTM achieved the best balance between precision and recall, as reflected by its highest F1 score.
5. **Balanced Accuracy (BACC)**:
    - **LSTM**: 0.74 (Highest)
    - Random Forest: 0.73

    - Naive Bayes: 0.69 LSTM had the highest balanced accuracy, indicating a strong performance in correctly predicting both classes.
6. **Brier Score**:
    - **LSTM**: 0.17 (Lowest, better)
    - Random Forest: 0.18
    - Naive Bayes: 0.26 A lower Brier score for LSTM indicates better calibration of predicted probabilities.

Conclusion

The LSTM model performed better across most evaluation metrics, including AUC, accuracy, precision, F1 score, balanced accuracy, and Brier score. Its ability to capture complex patterns in the data likely contributed to its superior performance, especially given the sequential and numeric nature of the dataset. While Random Forest also showed strong performance, it fell slightly behind the LSTM, but the training time of Random Forest was much less compared to that of LSTM. Naive Bayes had the lowest performance across all metrics, as it assumes feature independence, which is not suitable for this dataset.

Overall, the LSTM model is the best choice for this problem, given its consistently higher performance across metrics.

References

Dataset: Link to dataset

Github link of project: Project link