

## Bachelor of Technology

in

## Computer Science & Artificial Intelligence

**By**

**Roll. No :** 2203A52096 **Name**: KOTHAPELLY KEERTHANA

**Batch No:** 31

**Submitted to**





**SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE SR UNIVERSITY, ANANTHASAGAR, WARANGAL**

**April ,2025.**

**CSV FILE PROJECT: PROTIEN DATASET**

**ABOUT DATASET:**

***Data reading:***

**A screenshot of a computer code

Description automatically generated**

The dataset has about 9 columns. Which are 'mcg', 'gvh', 'alm', 'mit', 'erl', 'pox', 'vac', 'nuc', 'name'

***Data evaluation:***

***Info:***

The shape of the dataset is (1484, 9) which mean it has 1484 rows and 9 columns. The dataset has no null values. The data type of 'mcg', 'gvh', 'alm', 'mit', 'erl', 'pox', 'vac', 'nuc' are float and of ‘name’ is object.

***Description:***

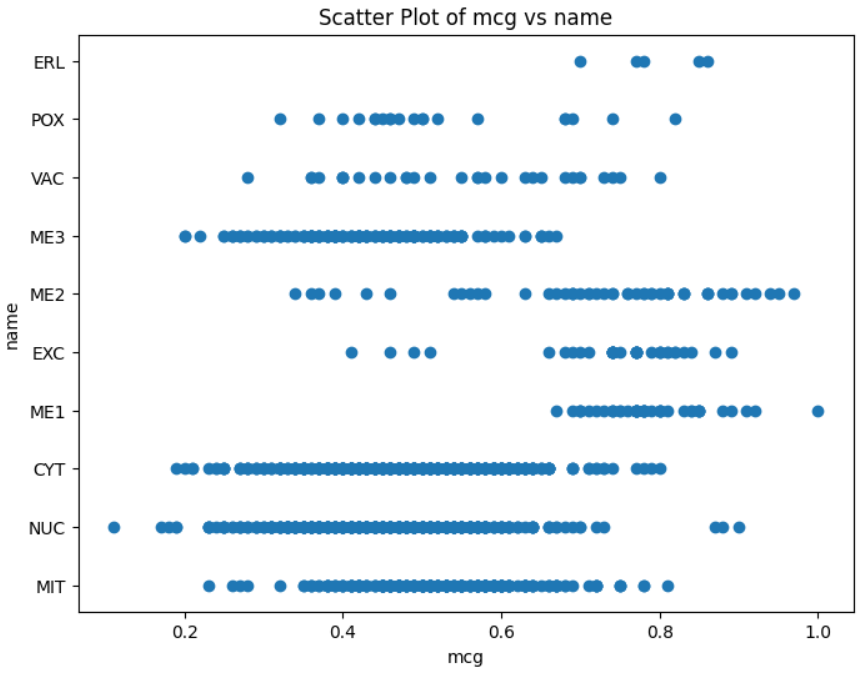
A screenshot of a computer code

Description automatically generated

The count, mean, standard deviation, minimum, 25%, 50%,75% and maximum values of each column are found and shown as above.

***Data visualization:***

***Scatter plot:***

A diagram of a scatter plot

Description automatically generatedA diagram of a scatter plot

Description automatically generatedA graph of blue dots

Description automatically generatedA graph with blue dots

Description automatically generatedA graph with blue dots

Description automatically generatedA graph of blue dots

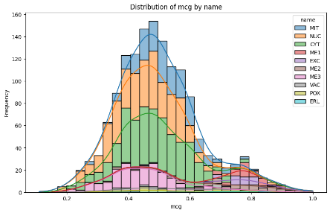
Description automatically generatedA diagram of a number of blue dots

Description automatically generated

The scatter plots are plotted between 'mcg', 'gvh', 'alm', 'mit', 'erl', 'pox', 'vac', 'nuc' and ‘name’, where 'mcg', 'gvh', 'alm', 'mit', 'erl', 'pox', 'vac', 'nuc' are features which are independent, so it they are taken on x-axis, ‘name’ is target variable, which is dependent so, it is taken on y-axis, The target variable has about 10 classes. The plot shows there are cluster in in different classes of the target variable “name”. For the scatter plots erl vs name and pox vs name there are very few data points to observe clusters and linearity. For remaining scatter plots, they show linearity class wise. i.e., for a particular class it has about one cluster and shows linearity. But when considered overall, there are a greater number of clusters and shows non- linearity.

***Histograms:***

A graph of different colored bars

Description automatically generatedA graph of different colored bars

Description automatically generatedA graph of a distribution of mit by name

Description automatically generatedA graph of a distribution of erl by name

Description automatically generatedA graph of a distribution of pox by name

Description automatically generatedA diagram of a distribution of vac

Description automatically generatedA diagram of a distribution of nuc

Description automatically generated

1. **Mcg vs name**:
   * Symmetry: symmetric, bimodal
   * Skewness: None
   * Tail: moderate tail
2. **Gvh vs name**:
   * Symmetry: symmetric, unimodal
   * Skewness: None
   * Tail: moderate tail
3. **Alm vs name**:
   * Symmetry: Symmetric, bimodal
   * Skewness: None
   * Tail: moderate tail
4. **Mit vs name**:
   * Symmetry: Asymmetric
   * Skewness: Skew right
   * Tail: Long right tail
5. **Erl vs name**:
   * Symmetry: Asymmetric
   * Skewness: Skew right
   * Tail: Very long right tail
6. **Pox vs name**:
   * Symmetry: Asymmetric
   * Skewness: Skew right
   * Tail: Very sharp right tail
7. **Vac vs name**:
   * Symmetry: Asymmetric
   * Skewness: Skew Left
   * Tail: Long left tail
8. **Nuc vs name**:
   * Symmetry: Asymmetric
   * Skewness: Skew right
   * Tail: Long right tail

***Box plot:***

**A graph of a box

Description automatically generatedA blue rectangular object with black lines

Description automatically generatedA graph with a blue rectangular object

Description automatically generatedA graph with a blue rectangular object

Description automatically generatedA graph with numbers and lines

Description automatically generatedA graph with numbers and lines

Description automatically generated with medium confidenceA diagram of a box plot of vac

Description automatically generatedA graph with numbers and lines

Description automatically generated**

1. **mcg:**
   * Outliers: Few present
   * Skewness: Normal Distribution
   * Range: 0.2-0.8
2. **gvh:**
   * Outliers: Few present
   * Skewness: Normal Distribution
   * Range:0.2-0.8
3. **alm:**
   * Outliers: Few present
   * Skewness: Normal Distribution
   * Range:0.3-0.7
4. **mit:**
   * Outliers: Present
   * Skewness: Positive skew
   * Range: 0.0-0.55
5. **erl:**
   * Outliers: Many present
   * Skewness: positive skew
   * Range:0.5-0.52
6. **pox:**
   * Outliers: Many present
   * Skewness: positive skew
   * Range:0.0-0.2
7. **vac:**
   * Outliers: Moderate
   * Skewness: negative skew
   * Range:0.4-0.6
8. **nuc:**
   * Outliers: Moderate
   * Skewness: positive skew
   * Range:0.1-0.4

**PREPROCESSING:**

The target variable is ‘name’, it has 10 classes. This dataset is classification dataset because it has categorical data. In preprocessing first, the target values are converted into from string to integer data type using label encoder.

In target variable class wise data points count:

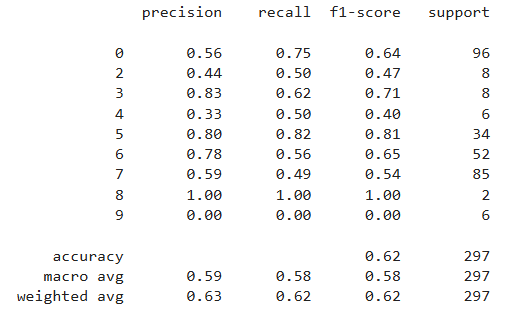
0->367 ,1->5 ,2->27 ,3->36 ,4->45 ,5->129 ,6->192 ,7->344 ,8->18, 9->24

Standard scalar is used to make all the feature in single range. So that all the features get equal priority. Then the dataset is split in train and test which are 80% and 20% respectively.

**MODEL TRANINIG:**

***1. SVM:***

Accuracy using SVM: 0.622895622895623



The model achieved an overall accuracy of 62%, indicating moderate performance. Precision and recall scores varied significantly across classes, with a macro average F1-score of 0.58, reflecting inconsistent results across all classes. Classes like 5 and 3 performed well with F1-scores above 0.70, while classes 4, 2, 7, and 9 had notably low performance. Class 8 showed perfect scores but had only 2 samples, which may not be reliable. Class 9 was not predicted at all, resulting in 0 precision and recall. The model performs better on frequent classes and struggles with rare or imbalanced ones.

A graph with numbers and a number in the center

Description automatically generated with medium confidence

The confusion matrix shows that the model correctly predicted many classes 0 (72), class 5 (28), class 6 (29), and class 7 (42) instances. However, there’s significant confusion between class 0 and class 6, with 19 class 0 instances misclassified as class 6 and vice versa. Class 4 was predicted poorly with just 1 correct prediction and many scattered misclassifications. Several classes (especially 0, 5, 6, 7) are often confused with each other, indicating overlapping features.

***2.LGBMClassifier:***

Accuracy using LGBMClassifier : 0.5791245791245792

A screenshot of a computer

Description automatically generated

The LGBMClassifier achieved an accuracy of 58%, showing moderate performance. Class 3 and Class 5 performed the best, with high F1-scores of 0.82 and 0.77, respectively. Class 0 had decent results due to its high support, while Classes 4, 6, and 7 showed weak performance. Classes 8 and 9 had zero precision and recall, indicating the model couldn't identify them at all. The macro average F1-score is 0.46, showing poor overall balance across classes.

A graph with numbers and a number in the center

Description automatically generated with medium confidence

Class 0 was often misclassified as Class 6 (30 times), indicating strong confusion between these two. Class 5 had 25 correct predictions but also frequent confusion with Classes 0 and 6. Class 4 was reasonably well predicted with 28 correct, though it had a few misclassifications. Classes 8 and 9 were poorly predicted, with very few or no correct classifications.

***3.Random Forest:***

Accuracy using Random Forest: 0.6026936026936027

A screenshot of a computer

Description automatically generated

The model achieved an overall accuracy of 58%, indicating moderate classification performance. Class 3 had the best results with an F1-score of 0.82, followed by Class 5 with 0.77. Class 0, the most frequent class, had a balanced performance (F1-score: 0.59). Classes 4, 6, and 7 showed weak performance, with F1-scores ranging from 0.31 to 0.55. Classes 8 and 9 had zero precision and recall, meaning the model failed to predict them correctly. The macro average F1-score is low (0.46), highlighting poor generalization across all classes. The model favors well-represented classes but struggles significantly with minority ones.

A graph with numbers and a number in the center

Description automatically generated with medium confidence

Class 0 was predicted correctly 57 times, but 30 instances were confused as class 6. Class 3, 4, 5, and 6 had decent true positive counts (28, 28, 25, and 49 respectively). Class 5 and 6 were often confused with each other, showing overlapping features. Class 1 and 2 had very low support and a few correct predictions. Class 8 had only 2 samples, both misclassified due to very low data. Class 9 does not appear in this matrix, because it had 0 correct or incorrect predictions.

**REMOVING OUTLIERS AND APPILED SMOTE:**

Classes with low data points are fixed and then outliers are removed because classes like class 1 have only 5 data points while removing outliers total classes data points are removed leading further complications.

New dataset shape after outlier removal is (1081, 8) (1081,)

Class distribution after outlier removal:

0->383 ,7->301, 6->175, 5->118, 4->30, 9->23, 2->19 3->19, 8->8, 1->5

Now for classes with few data points data points are oversampled, duplicate values are added (i.e. same data points are again added)

New class distribution(after oversampling minority classes):

0->383 ,7->301, 6->175, 5->118, 4->30, 9->23, 2->19, 3->19, 8->8, 1->30

Unique class labels: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

Next SMOTE is used and data is balanced.

Class distribution after ***SMOTE:***

0->383 ,7->383, 6->383, 5->383, 4->383, 9->383, 2->383, 3->383, 8->383, 1->383

Pre preprocessed with new data.

Then again Models are trained.

**MODELS TRAINED:**

***1.SVM***

Accuracy using SVM: 0.6840731070496083

A screenshot of a screen shot

Description automatically generated

This model achieved an overall accuracy of 68%, which is higher than all previous models. Class 1 achieved perfect performance with precision, recall, and F1-score of 1.00. Classes 2, 3, 4, and 5 also performed well, with F1-scores above 0.70. Class 0 had the worst performance, with a low F1-score of 0.27, indicating many misclassifications. Class 7 also struggled, with low precision (0.51) and recall (0.42). Both macro and weighted averages are around 0.68, showing consistent performance across most classes.

A graph with numbers and a graph

Description automatically generated with medium confidence

The model performs well for most classes, especially classes 1 to 5, with strong diagonal values. Class 0 is frequently misclassified as 7, 8, and 9, indicating confusion in edge categories. Classes 6 to 9 also show moderate misclassification into nearby classes. Overall, the model shows good performance with room for improvement in distinguishing similar classes.

***2. LGBMClassifier***

Accuracy using lLGBMClassifier 0.8616187989556136

A screenshot of a graph

Description automatically generated

The model achieved a high accuracy of 86% on 766 samples. Most classes have precision, recall, and F1-scores above 0.90, indicating excellent performance and consistency. Class 0 and Class 7 have relatively lower F1-scores of 0.60, suggesting room for improvement in those categories. The macro and weighted averages are both around 0.86–0.87, confirming balanced and strong performance across all classes.

A graph of a graph with numbers and a number

Description automatically generated with medium confidence

Class 0 and Class 7 had the most confusion, with several misclassifications between them. The confusion matrix confirms strong predictions for most classes with minimal errors.

***3.Random Forest:***

Accuracy using Random Forest: 0.8616187989556136

A screenshot of a table

Description automatically generated

The classification report shows an accuracy of 86%, with strong performance across most classes. Precision and recall are near-perfect for classes 1, 2, and 3, while class 0 and 7 have slightly lower scores (f1-scores of 0.62 and 0.64 respectively), indicating some misclassification. Macro and weighted averages are balanced at 0.86,

A graph of a graph with numbers

Description automatically generated with medium confidence

Most misclassifications occur in class 0 (notably predicted as class 7 and 8) and class 7 (confused with classes 0, 4, 5, and 9). However, classes like 1, 2, and 3 are predicted with near-perfect accuracy. The model handles the dataset quite consistently, though slight improvements could focus on better distinguishing between visually similar categories.

**SKEWNESS AND KURTOSIS:**

***Skewness:***

mcg 0.603680

gvh 0.416218

alm -0.220772

mit 1.443315

erl 10.149361

pox 10.266493

vac -1.789829

nuc 2.410591

***Kurtosis:***

mcg 0.453474

gvh 0.550199

alm 1.599862

mit 2.278222

erl 101.009524

pox 105.378745

vac 9.465336

nuc 7.747540

**STATISTICAL TESTS:**

***Model: LGBMClassifier:***

Mean Residual: 0.09090909090909091

Standard Deviation of Residuals: 3.7903439971469144

Z-Test Statistic: 0.4133395029001524, P-Value: 0.6796571765973735

T-Test Statistic: 0.4133395029001524, P-Value: 0.6796571765973735

***Model: Random Forest:***

Mean Residual: 0.08417508417508418

Standard Deviation of Residuals: 3.695731478128673

Z-Test Statistic: 0.39251962474792157, P-Value: 0.694956813410887

T-Test Statistic: 0.39251962474792157, P-Value: 0.694956813410887

F-Test Statistic: 0.9507001550269725, P-Value: 0.33195772401468654

***Model: SVM:***

Mean Residual: 0.5286195286195287

Standard Deviation of Residuals: 3.564088305966253

Z-Test Statistic: 2.556071346426705, P-Value: 0.011085549314284857

T-Test Statistic: 2.556071346426705, P-Value: 0.011085549314284857

F-Test Statistic: 0.8841779057264912, P-Value: 0.14509773756333502

***Summary***

ANOVA F-Statistic: 1.418960873271315

ANOVA P-Value: 0.24251340446842928

***Type I Error (False Positive Rate):***

'LGBMClassifier': (0.32034282340262654),

'Random Forest' (0.305043186589113),

'SVM': (0.9889144506857152)

***Type II Error (False Negative Rate):***

'LGBMClassifier':(0.6796571765973735),

'Random Forest': (0.694956813410887),

'SVM': (0.011085549314284857)}

***Best Model Based on Mean Residual: Random Forest***

**IMAGE FILE PROJECT: FASHION DATASET**

**ABOUT DATASET:**

***Data directories:***

The dataset has the following directories. Each directory can be considered a class.

['hoodies', 'skirts', 'shorts', 'Jeans', 'Dress']

So, here we have 5 classes.

***Data exploration:***

A screenshot of a computer

Description automatically generated

***Data visualization:***

The data visualization is done to know more clearly about this dataset. For each class randomly 5 sample images are shown in gray scale and rgb. i.e., for 'hoodies' 5 random sample images, for 'skirts' 5 random sample images, for 'shorts' 5 random sample images, for 'Jeans' 5 random sample images, for 'Dress' 5 random sample images.

***Gray scale:***



***Rgb:***

A collage of different clothes

Description automatically generated

**PRE PREOCESSING:**

Hyperparameters are taken as IMG\_SIZE = (200, 200), BATCH\_SIZE = 16, EPOCHS = 10.

Image Data Generator is used to rescale the images. Rescale is taken as 1. /255, validation split is taken as 0.2 Which categorized 1178 images belonging to 5 classes and 293 images belonging to 5 classes.

**MODEL TRAINING:**

***1.CNN:***

A screenshot of a computer program

Description automatically generated

A screenshot of a computer screen

Description automatically generated

The classification report shows that the model is underperforming with an overall accuracy of just 23%. Precision, recall, and F1-scores across all classes (hoodies, skirts, shorts, jeans, dress) are low, ranging mostly between 0.15 and 0.30. The model struggles the most with distinguishing dress, which has the lowest F1-score of 0.16.

A diagram of a variety of blue squares

Description automatically generated with medium confidence

The confusion matrix shows that the CNN model struggles to distinguish between clothing categories, with frequent misclassifications across all classes. For instance, many hoodies are misclassified as skirts, and shorts are often confused with dress and jeans. The overall performance is low, which aligns with the 23% accuracy reported earlier, indicating poor classification effectiveness.

***2. MobileNetV2:***

**Total params:** 2,422,597 (9.24 MB)

**Trainable params:** 164,613 (643.02 KB)

**Non-trainable params:** 2,257,984 (8.61 MB)

A screenshot of a computer screen

Description automatically generated

This classification report shows strong model performance, with an overall accuracy of 89%. Most classes like shorts and skirts have high precision, recall, and F1-scores (above 0.9), indicating consistent predictions. Dress has the lowest performance with an F1-score of 0.75, but it's still fairly reliable. Overall, the model performs well across all clothing categories.

A graph of clothing items

Description automatically generated with medium confidence

The MobileNetV2 confusion matrix shows strong performance, especially for skirts, shorts, and hoodies, with minimal misclassifications. Jeans and dress have a few confusions, particularly with each other, but still maintain good prediction accuracy.

***3. EfficientNetB0***

**Total params:** 4,214,184 (16.08 MB)

**Trainable params:** 164,613 (643.02 KB)

**Non-trainable params:** 4,049,571 (15.45 MB)

A screenshot of a graph

Description automatically generated

This classification report shows strong model performance, with an overall accuracy of 91%. Most classes like shorts skirts, hoodies and jeans have high precision, recall, and F1-scores (above 0.9), indicating consistent predictions. Dress has the lowest performance with an F1-score of 0.82. So, the model performs well across all clothing categories.

A diagram of clothing items

Description automatically generated with medium confidence

The EffiecentNetB0 confusion matrix shows strong performance, especially for skirts, shorts, hoodies and jeans, with minimal misclassifications. Dress has a few confusions, but still maintain good prediction accuracy. Overall, the model is effective with clear diagonal dominance indicating accurate class predictions.

**STATITICAL TESTS:**

***Val accuracies of each model in different epochs are taken manually in a list and statistical tests are performed.***

***F-test (Levene’s Test for Equal Variance):***

F = 3.2035, p = 0.0564 → Equal variances

***Independent t-tests (pairwise):***

CNN vs MobileNetV2: t = -9.8198, p = 0.0000

CNN vs EfficientNetB0: t = -13.6705, p = 0.0000

MobileNetV2 vs EfficientNetB0: t = -2.1450, p = 0.0458

***One-way ANOVA Test:***

F = 107.5901, p = 0.0000 → Significant difference

***Considering all factors efficientNetB0 is the best model for this dataset.***

**AUDIO FILE PROJECT: INDIAN LANGUAGES DATASET**

**ABOUT DATASET:**

***Data directories:***

The dataset has the following directories. Each directory can be considered a class.

['Telugu', 'Bengali', 'Tamil', 'Kannada', 'Punjabi', 'Malayalam', 'Hindi']

So, here we have 7 classes.

***Data summary:***

A screenshot of a computer code

Description automatically generated  
This table shows some basic audio features for different Indian languages. It includes the average duration of the audio clips, how often the sound changes suddenly (called zero crossing rate), and how loudor energetic the sound is (measured by RMS energy). All the languages have almost the same duration, around 5 seconds. Hindi and Kannada have the highest rate of sudden sound changes, which might mean they have sharper or more complex sounds. Tamil has the highest energy, meaning it might sound louder or more powerful, while Kannada has the lowest energy, so it might sound softer.

***Data audio duration distribution:***

A graph with text and numbers

Description automatically generated with medium confidence

***Waveform:***

The following is a waveform of a sample audio file.

A blue sound waveform with numbers

Description automatically generated  
The waveform shows varying amplitude over time, indicating changes in loudness and the presence of multiple sound segments, likely corresponding to syllables or words. There are clear peaks and gaps, suggesting pauses between spoken parts. The energy appears higher at the beginning and becomes more evenly distributed later, which may reflect a dynamic speaking pattern.

Sample Rate: 22050

Audio Duration: 4.98 seconds

***Spectrogram:***

A blue and white spectrogram

Description automatically generated

The spectrogram shows how the frequency content of the audio changes over time. Brighter areas (orange/red) indicate higher energy at specific frequencies and moments. The presence of vertical patterns suggests clear speech segments, with energy concentrated in the lower frequency range (below 4000 Hz), which is typical for human speech. The variations in intensity and frequency bands reflect changes in pronunciation, pitch, and articulation across time.

***MFCC:***

A red and blue graph

Description automatically generated

The MFCC (Mel-Frequency Cepstral Coefficients) plot shows the audio's timbral characteristics over time. In this visualization, the lower bands (in blue) capture more detailed variations, which are crucial for distinguishing different speech sounds. Most of the coefficients remain in the warmer color range (red to light orange), indicating stable features. The visible changes in the lower MFCCs suggest the presence of distinct phonetic events or syllables, useful for tasks like speech recognition or speaker identification.

**PRE-PROCESSING**

Extracted 13 MFCC mean features per audio file. Encoded each language label. Taken language name as target variable. Prepared the data for classification.

For training random forest model should be X is 2-D which is (7000 ,13) for this dataset, And for training LSTM model X should be 3-D which is (7000,100,13). Pre-Processing and training are done separately.

**MODEL TRAINING**

***Random forest***:

A screenshot of a graph

Description automatically generated

The classification report shows the performance of a Random Forest model in identifying seven Indian languages based on audio features. The model achieved high accuracy (95%) overall. Languages like Hindi, Kannada, Tamil, and Punjabi have particularly high precision and recall scores, indicating they were classified very accurately. Bengali has slightly lower precision (0.88), meaning the model was more likely to confuse it with other languages. However, all languages have f1-scores above 0.90, suggesting balanced and strong performance. The macro and weighted averages also confirm consistent accuracy across all classes.

A graph of a number and a number

Description automatically generated with medium confidence

***LSTM:***

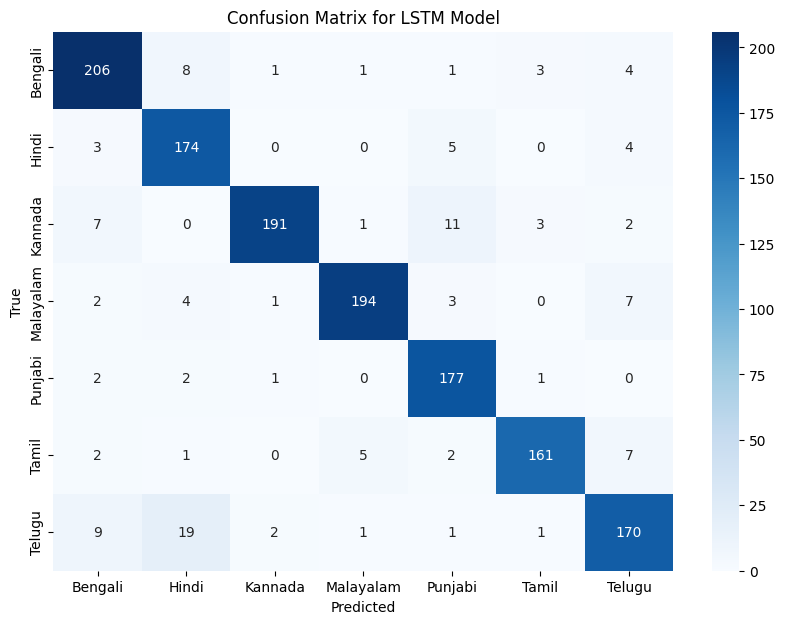
***A screenshot of a computer

Description automatically generated***

***A screenshot of a graph

Description automatically generated***

The model achieved an overall accuracy of 91% in classifying seven Indian languages. It performed best on Kannada, Malayalam, and Tamil, with high precision and f1-scores. Hindi and Telugu showed slightly lower performance, indicating more misclassifications. Overall, the model is effective but can be improved for certain languages.



**STATISTICAL TESTS:**

T-statistic: 29.79914286297416

P-value (T-Test): 2.6370559834455445e-10

Z-score: 34.23087998670484

P-value: 8.398267592510873e-257