CS771: Introduction to Machine Learning Course Project

Satyam Kumar
220983

Deepak Chaurasia
220330

Amruth Raj
220642

Ujjwal Kumar
21123

Sameer
210912

1 Introduction

In this project, we address a binary classification task using three distinct datasets, each derived from the same raw data source but with unique feature representations. This setup enables a comparative analysis of model performance across different feature sets. Key aspects of the project are as follows:

- **Objective**:
  - To evaluate and compare the performance of machine learning models on three datasets, each designed for the same binary classification task but represented with different features.
- **Datasets Provided**:
  - Each of the three datasets represents the same binary classification task and is generated from the same raw data.
  - These datasets differ in terms of feature representations, allowing for a comparative analysis of the impact of feature engineering.
- **Dataset Structure**:
  - **Training Set**: Used for model training.
  - **Validation Set**: Labels given, allowing hyperparameter tuning and model selection.
  - **Test Set**: Labels are hidden; final predictions will be submitted for evaluation against the ground truth.
- **Evaluation and Submission**:
  - The validation set is provided to enable model optimization and performance analysis, while the test set's true labels are hidden to ensure objective evaluation.
  - Predictions for the test set will be submitted, and final model performance will be assessed based on these predictions.
- **Purpose of Analysis**:
  - This setup provides an opportunity to:
    - Experiment with various model selection and hyperparameter tuning strategies.
    - Analyze the impact of different feature engineering approaches on classification accuracy.

2 Data Exploration and Data Preprocessing

In this section, we explore each dataset and describe the preprocessing steps applied to prepare it for the corresponding model. The datasets include the NPZ, Text Sequence, and Emoticon datasets, each requiring specific transformations for optimal model performance.

2.1 NPZ Dataset Preprocessing

The NPZ dataset consists of high-dimensional numerical features. Preprocessing for this dataset involves flattening, dimensionality reduction, and standardization to prepare it for Logistic Regression modeling.

- **Data Flattening**:
  - Each instance in the NPZ dataset is a multidimensional array. Before any further processing, these arrays are flattened into one-dimensional vectors. Flattening simplifies the data structure and ensures uniform input shapes, allowing for seamless integration into the model pipeline.
- **Dimensionality Reduction using PCA**:
  - Given the high dimensionality of the NPZ dataset, Principal Component Analysis (PCA) is employed to reduce the feature space to 100 components. This step retains the most significant information while discarding redundant features, effectively lowering computational complexity and mitigating the risk of overfitting.
- **Standardization**:
  - After PCA transformation, the reduced features are standardized to have a mean of zero and a standard deviation of one. Standardization optimizes the data for Logistic Regression, ensuring balanced feature scaling and stable convergence during training.

2.2 Emoticon Dataset Preprocessing

The Emoticon dataset consists of sequences of emojis, each treated as a categorical feature. The preprocessing steps involve the following transformations:

- **Emoji Splitting**:
  - Each emoji sequence is split into individual emojis and assigned to separate columns, standardizing the data format. To maintain consistency, sequences shorter than the maximum length of 13 are padded.
- **One-Hot Encoding**:
  - Each split emoji is transformed into a binary vector using one-hot encoding, treating each unique emoji as a distinct categorical feature suitable for logistic regression modeling.
- **Hyperparameter Tuning**:
  - Hyperparameter tuning is performed on the regularization parameter to optimize the Logistic Regression model, employing cross-validation to select the best model configuration.

2.3 Text Sequence Dataset Preprocessing

The Text Sequence dataset consists of character-based sequences, which require specialized preprocessing to convert them into a suitable format for sequential modeling.

- **Character Tokenization**:
  - Each sequence undergoes character-level tokenization, where each character in the sequence is mapped to a unique integer based on its position in the vocabulary. This mapping enables the model to interpret characters as numeric data.
- **Truncation and Padding**:
  - Each sequence is truncated to remove the first three characters and then padded to a fixed length of 47. Padding ensures that all sequences have a uniform length, which is required for efficient processing in sequential neural networks.
- **Embedding Transformation**:

  - After tokenization, sequences are passed through an embedding layer, transforming characters into dense, low-dimensional representations. This step captures semantic relationships between characters in a way that the model can learn effectively.

2.4 Combined Feature Model Preprocessing

The Combined Model integrates features from the Emoticon, Text Sequence, and NPZ datasets. Each dataset undergoes specific preprocessing steps—one-hot encoding or scaling—before being concatenated to create a unified feature set for model training.

- **Emoticon Dataset Preprocessing**:
  - The Emoticon dataset, which consists of categorical data in the form of emoji sequences, is preprocessed using one-hot encoding. Each emoji sequence is transformed into a binary vector representation, enabling the model to interpret emojis as distinct categorical features.
- **Text Sequence Dataset Preprocessing**:
  - Similar to the Emoticon dataset, the Text Sequence dataset comprises character-based sequences, treated as categorical data. This dataset is also preprocessed using one-hot encoding, converting each sequence into a binary vector format to capture individual character representations.
- **NPZ Dataset Preprocessing**:
  - The NPZ dataset contains high-dimensional numerical features. To standardize these features, a scaler is applied to ensure each feature has a mean of zero and a standard deviation of one. This scaling step improves model stability and optimizes numerical feature integration in the combined dataset.
- **Concatenation of Encoded and Scaled Features**:
  - After one-hot encoding the Emoticon and Text Sequence datasets, and scaling the NPZ dataset, all three processed datasets are concatenated along their feature dimensions. This combined feature matrix incorporates both categorical (Emoticon and Text Sequence) and numerical (NPZ) information, allowing the model to leverage a diverse set of features for improved classification performance.

3 Model Description and Performance

This section outlines each model applied to the datasets, detailing the specific parameters chosen and the achieved validation accuracy.

3.1 NPZ Dataset Model: FeatureModel

- **Model Type**: Logistic Regression with Principal Component Analysis (PCA) for dimensionality reduction.
- **PCA Components**: 100 components, reducing the initial high-dimensional feature space of 9984 to improve computational efficiency and retain essential variance.
- **Regularization**: L2 regularization with a regularization strength parameter $C = 100$, helping prevent overfitting by penalizing large coefficients.
- **Solver**: `lbfgs`, optimized for handling small datasets with high dimensionality.
- **Maximum Iterations**: Set to 700 to ensure convergence.
- **Validation Accuracy**: 98.57%

3.2 Emoticon Dataset Model: EmoticonModel

- **Model Type**: Logistic Regression with one-hot encoded categorical data.
- **Regularization**: L1 regularization to encourage sparsity in the model by reducing the impact of less relevant features.

- **Regularization Strength**: Tuned to $C = 10$ using cross-validation to find the optimal configuration.
- **Solver**: `liblinear`, effective for sparse data.
- **Hyperparameter Tuning**: GridSearchCV was applied to select the best regularization strength.
- **Validation Accuracy**: 92.84%

3.3 Text Sequence Dataset Model: TextSeqModel

- **Model Type**: Long Short-Term Memory (LSTM) network designed to capture temporal dependencies in character-based sequences.
- **Embedding Layer Dimension**: 16-dimensional space, providing a dense representation for each character token.
- **LSTM Units**: 32 units in the LSTM layer to learn sequential dependencies.
- **Regularization**: Dropout layer with a dropout rate of 0.3, reducing the risk of overfitting by randomly omitting certain features during training.
- **Optimizer**: Adam optimizer, which adapts the learning rate based on model performance.
- **Batch Size**: 32, allowing efficient use of memory during training.
- **Maximum Epochs**: 200 epochs, providing sufficient learning time for improved generalization.
- **Validation Accuracy**: 84.00%

3.4 Combined Model: Integration of Emoticon, Text Sequence, and NPZ Features

The Combined Model integrates features from the Emoticon, Text Sequence, and NPZ datasets. Three classifiers were evaluated on this combined feature set to leverage different model strengths:

- **Random Forest Classifier**:
  - **Number of Estimators**: 100 trees, providing a robust ensemble approach.
  - **Criterion**: Gini impurity to evaluate splits in nodes.
  - **Maximum Depth**: No depth restriction, allowing trees to grow fully.
  - **Validation Accuracy**: 97.96%
- **XGBoost Classifier**:
  - **Evaluation Metric**: Multiclass log loss (`mlogloss`) to assess the performance.
  - **Learning Rate**: 0.1, controlling the step size for each iteration.
  - **Maximum Depth**: 6, balancing model complexity with computational efficiency.
  - **Validation Accuracy**: 98.36% (highest among classifiers in the Combined Model)
- **Logistic Regression**:
  - **Regularization**: L2 regularization to maintain stability and avoid overfitting.
  - **Solver**: `lbfgs`, suitable for large datasets.
  - **Maximum Iterations**: 1000, ensuring model convergence with a complex feature set.
  - **Validation Accuracy**: 98.16%

4 Experiment Analysis

This section provides an analysis of the experiments conducted on each dataset. For each dataset, we present the best accuracy achieved by the models and include plots that illustrate the performance of different models.

4.1 NPZ Dataset

For the NPZ dataset, we tested three models: Logistic Regression, Random Forest, and XGBoost. The best validation accuracy achieved by each model is as follows:

- **Logistic Regression**: 98.57%
- **Random Forest**: 97.96%
- **XGBoost**: 98.36%

Figure 1 presents a comparison of these models on the validation set, highlighting that Logistic Regression achieved the highest accuracy, closely followed by XGBoost.

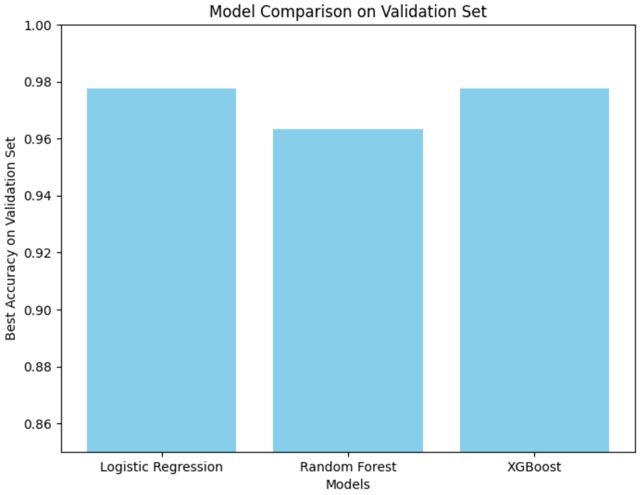

Figure 1: Model Comparison on Validation Set for NPZ Dataset

Additionally, Figure 2 illustrates how the training size affects accuracy across these models. Both Logistic Regression and XGBoost demonstrate improved performance with increased training sizes, whereas Random Forest peaks at a moderate training size before slightly declining.

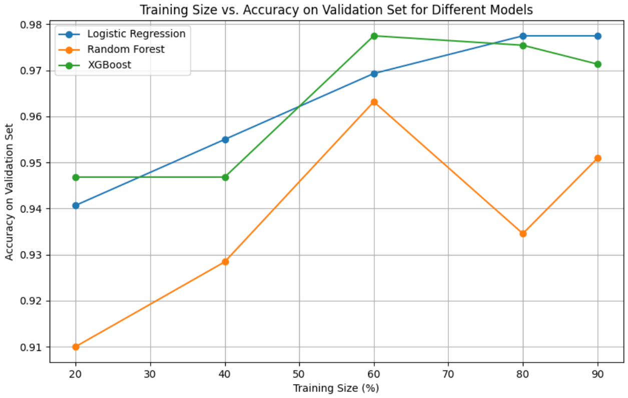

Figure 2: Training Size vs. Accuracy on Validation Set for NPZ Dataset

4.2 Emoticon Dataset

For the Emoticon dataset, we tested three models: Logistic Regression, Random Forest, and XGBoost. The best validation accuracy achieved by each model is as follows:

- **Logistic Regression**: 92.84%
- **Random Forest**: 86.50%
- **XGBoost**: 82.30%

Figure 3 presents a comparison of these models on the validation set, indicating that Logistic Regression achieved the highest accuracy.

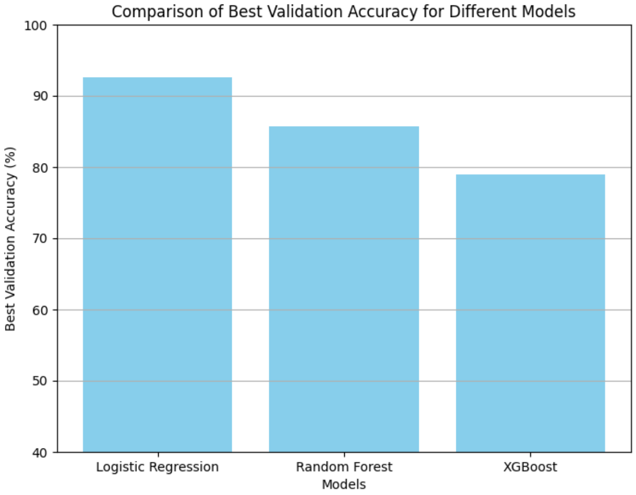

Figure 3: Model Comparison on Validation Set for Emoticon Dataset

Additionally, Figure 4 illustrates how the training size affects accuracy across these models. Logistic Regression consistently improves with increased training size, while Random Forest and XGBoost show variable trends as the training size increases.

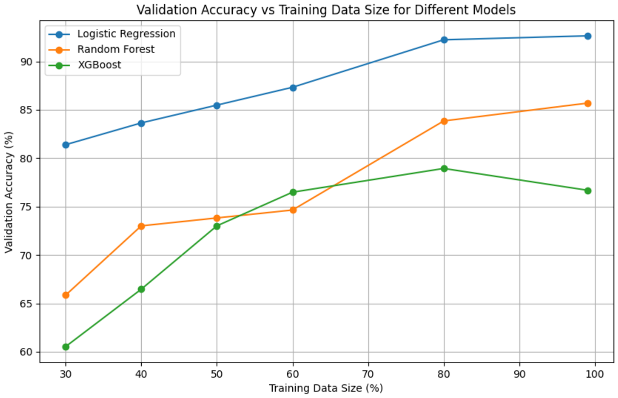

Figure 4: Training Size vs. Accuracy on Validation Set for Emoticon Dataset

4.3 Combined Dataset

For the Combined dataset, we tested three models: Random Forest, XGBoost, and Logistic Regression. The best validation accuracy achieved by each model is as follows:

- **Random Forest**: 98.50%
- **XGBoost**: 98.30%
- **Logistic Regression**: 98.45%

Figure 5 presents a comparison of these models on the validation set, showing that Random Forest achieved the highest accuracy, closely followed by Logistic Regression.

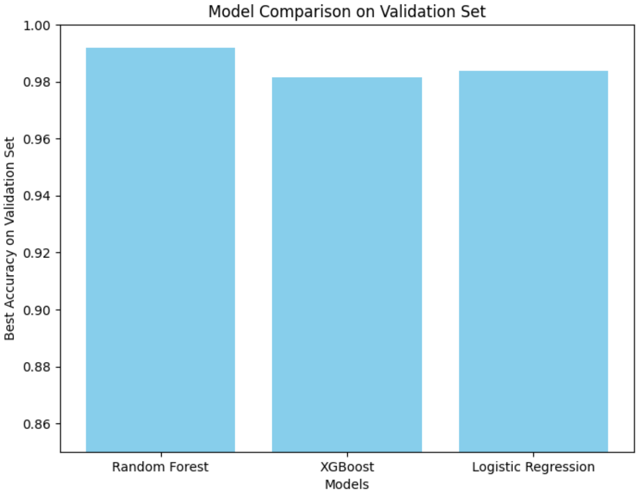

Figure 5: Model Comparison on Validation Set for Combined Dataset

Additionally, Figure 6 illustrates how the training size affects accuracy across these models. Random Forest shows an initial increase and reaches a peak before a slight decline, while XGBoost and Logistic Regression demonstrate stable performance across varying training sizes.

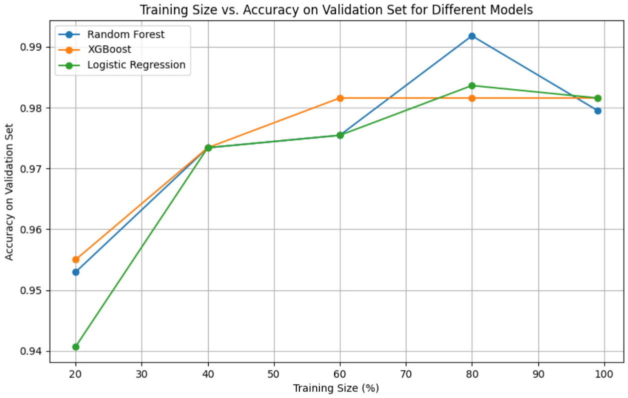

Figure 6: Training Size vs. Accuracy on Validation Set for Combined Dataset

4.4 Text Sequence Dataset

For the Text Sequence dataset, we used an LSTM-based model. The best validation accuracy achieved by this model is as follows:

- **LSTM Model**: 84.00%

Figure 7 shows how the training size affects accuracy for the LSTM model. The plot illustrates that increasing the training size leads to gradual improvements in accuracy, demonstrating the model's ability to benefit from larger datasets.

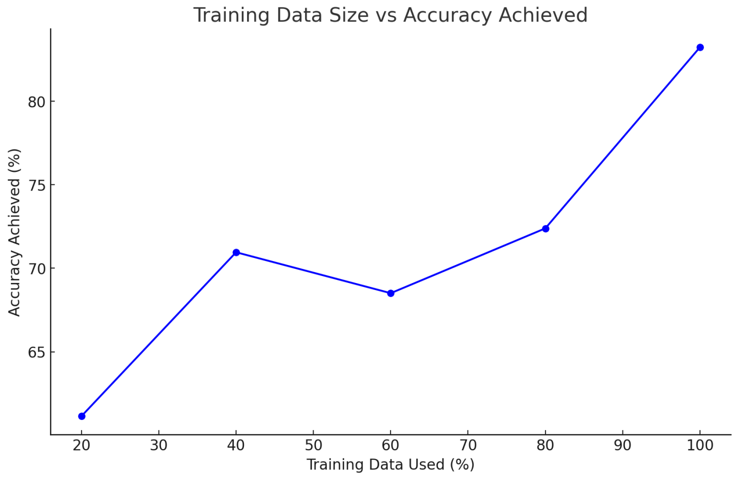

Figure 7: Training Size vs. Accuracy on Validation Set for Text Sequence Dataset

5 Conclusion

In this mini-project, we explored various binary classifiers across four distinct datasets, each with unique feature representations. Through extensive experimentation and analysis, we derived several insights:

- **Best-performing models per dataset:** The Logistic Regression model performed best on both the NPZ and Emoticon datasets, achieving high accuracy and stability. For the Text Sequence dataset, the LSTM-based model was most suitable, capturing the sequential nature of the data. For the Combined dataset, Random Forest achieved the highest accuracy, leveraging features from all datasets.
- **Training size impact:** Across all datasets, larger training sizes generally improved model accuracy. Models like Logistic Regression and LSTM showed a strong correlation between increased training data and accuracy improvements, emphasizing the benefit of more data in training robust classifiers.

Overall, this project demonstrated that model selection and feature integration strategies should be tailored to the specific characteristics of the dataset to maximize classification performance.