CS771 MINIPROJECT - 1

Palagiri Tousif Ahamad
220744

T.V.S Pawan Chanukya Reddy
221120

M.V Abhiram
220599

Daksh Kumar Singh
220322

Tanishq Maheshwari
221128

Binary Classification

1 Introduction

This mini-project involves binary classification using three distinct datasets. Each dataset represents the same classification task but differs in feature representation. The project is divided into two tasks: the first focuses on building optimal models for each dataset independently, and the second investigates whether combining the datasets can lead to improved performance. The objective is to optimizing the model with a focus on accuracy and efficiency with respect to training data.

2 Task 1: Choosing the best model

2.1 Emoticons Dataset

This dataset represents each input using 13 emoticons, with each emoticon serving as a categorical feature.

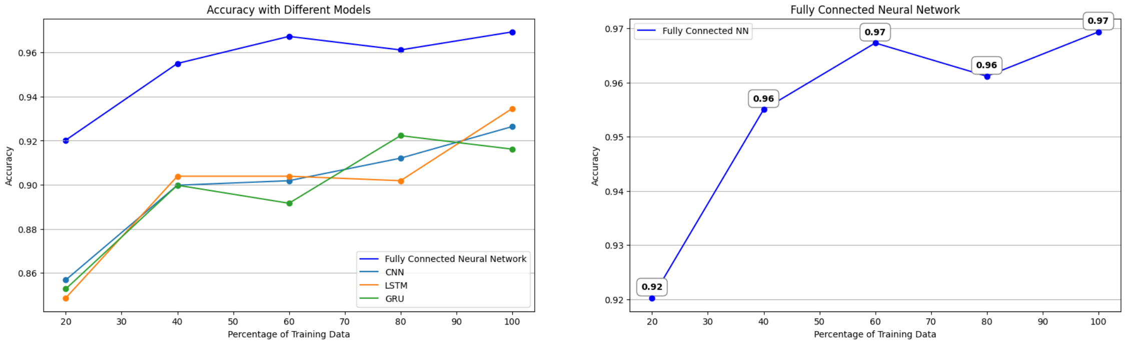
2.1.1 Feature Extraction

Since the dataset consisted of sequences of emojis, a tokenizer is used to transform these emojis into numerical tokens. Each emoji is assigned a unique integer, enabling the machine learning models to process them efficiently. There are a total of 214 unique emojis in training data.

2.1.2 Models

After tokenizing the data, we initially experimented with traditional machine learning models like SVM, Random Forests, and Logistic Regression. However, these models yielded lower-than-expected accuracies, suggesting that they struggled to capture the complex patterns in the emoticon-based dataset. Hence we shifted our approach towards using neural networks. We tried the following neural networks

- **Fully Connected Neural Network (Best Model)**
- CNN
- LSTM
- GRU



(a) Accuracy vs Training data

(b) Performance of Fully connected NN

Figure 1: Performance of various models with emoticon dataset

From Figure 1, a simple fully connected neural network with two dense layers is performing better than other models even with lower training data. All the other models couldn't give more than 94% accuracy even while using 100% of training data.

2.1.3 Results

The fully connected neural network trained on complete training data reached an accuracy of 97% on testing data. The confusion matrix of the same is below

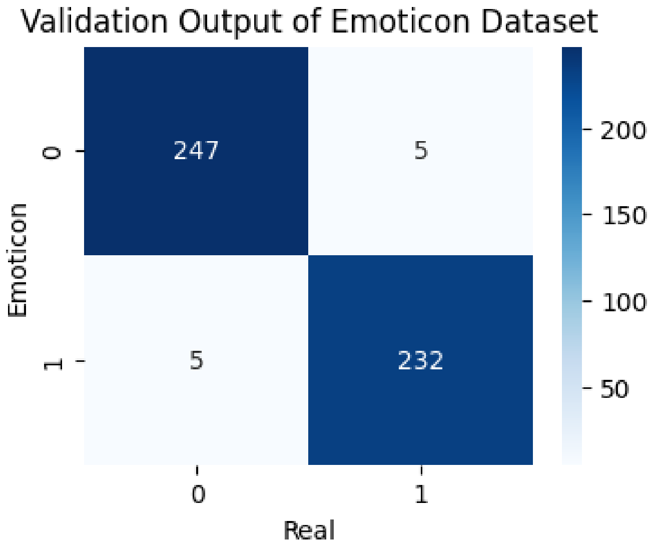


Figure 2: Confusion Matrix

2.2 Deep Features Dataset

In this dataset, the features of each input are extracted using a deep neural network, resulting in a 13x786 matrix. Each input is represented by a 786-dimensional embedding for each of the 13 emoticon features.

2.2.1 Feature Extraction

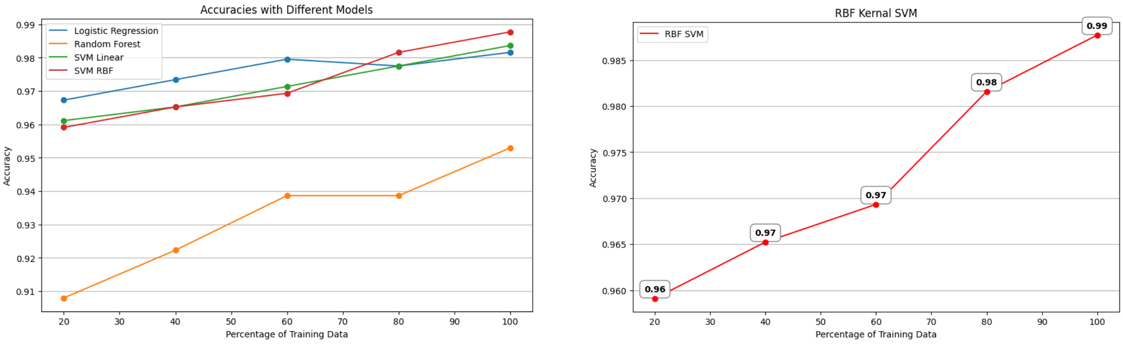
Each input is $(13, 768)$ dimensional matrix, hence the first step we did is to flatten the entire matrix to obtain $(9984,)$ sized array. As the number of features in this array is large, we chose to apply **PCA**(Principal Component Analysis) to reduce the dimensionality of the data to obtain 200 components. Hence after this step each input contains an array of 200 features extracted by PCA.

2.2.2 Models

The models we considered to train with this data are :

- SVM with linear Kernel
- **SVM with RBF kernel(Best model)**
- Logistic Regression
- Random Forest

As the input features are already extracted from a neural network, we are able to obtain decent accuracies even with simple machine learning models. These models are able to find patterns in the extracted features.



(a) Performance of various models

(b) Performance of SVM with RBF kernel

Figure 3: Performance of various models with feature dataset

From Figure 3, Logistic Regression gives good accuracy with lower training data whereas SVM with RBF kernel gives better accuarcy as we increase the amount of training data. This suggests that Logistic Regression is more effective with limited data due to its simplicity, while SVM, particularly with non-linear kernels like RBF, benefits from larger datasets, leveraging the increased data to better capture complex patterns and decision boundaries. Since, Logistic Regression, Linear and RBF Kernal Support Vector machines are working with very good validation accuracies $(96\% - 99\%)$, We choose RBF Kernal SVM.

2.2.3 Results

When trained with complete training data the RBF kernel based SVM gave an accuracy over 99%. The confusion matrix of the same is below

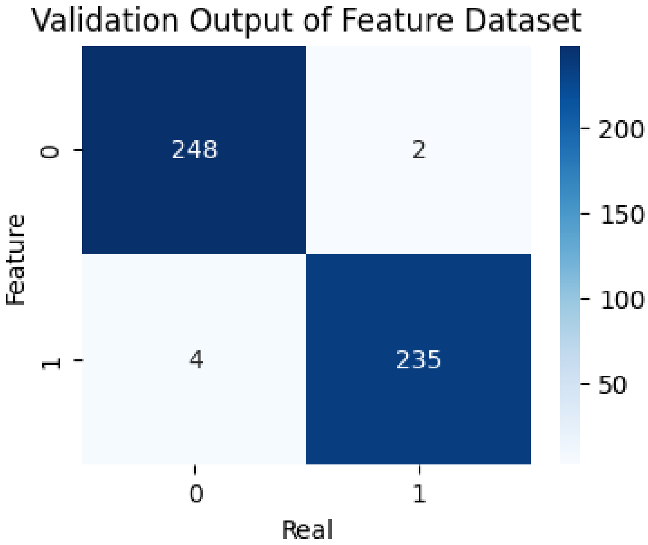


Figure 4: Confusion matrix

2.3 Text Sequence Dataset

The inputs in this dataset are represented by a string of 50 digits, where each string corresponds to a specific instance of the binary classification task.

2.3.1 Feature Extraction

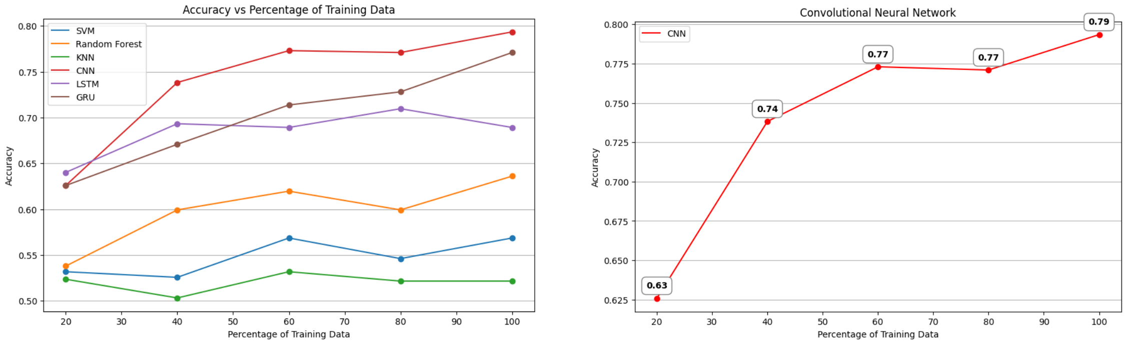
In this feature extraction step, the sequences are converted from character strings to numerical representations. Each character in the sequences is transformed into a list of integers. This conversion prepares the data for machine learning models.

2.3.2 Models

When this dataset is observed closely, there is a certain one-one correlation between emojis and the digits in this dataset. We can observe repetition of certain pattern of digits at places where emojis repeated in dataset-1. Moreover there is a variable length padding of zeros at the start of each input. We initially tried the models that we used for dataset-2 for this data but the accuracies obtained are low.

Hence we decided to experiment with neural networks to find the optimal model. The models we decided to work with include

- KNN
- **CNN(Best model)**
- LSTM
- GRU



(a) Accuracy vs Training data

(b) Performance of CNN

Figure 5: Performance of various NN models with feature dataset

From Figure 5, LSTM gave slightly more accuracy when training at 20% training data than CNN but CNN overtakes LSTM rapidly as we increase the training data. Also the increase in accuracy in the case of LSTM is not as prominent as in the case of CNN or GRU. Hence CNN is better model for this dataset.

2.3.3 Results

CNN model when trained with complete training data gave an accuracy of 78% on the validation data. The confusion matrix for the same is below

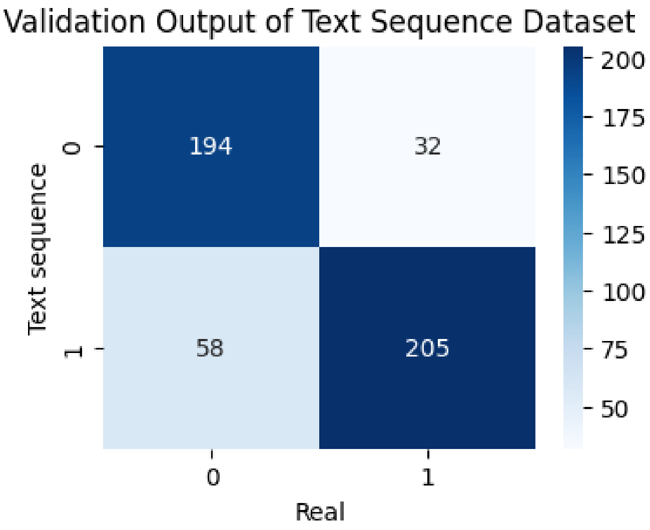


Figure 6: Confusion matrix

One possible reason as to why CNN performed better than other models might be its ability to detect local patterns. As there is repetition of certain patterns all throughout the dataset, the model might have captured it resulting in better performance.

3 Task 2 : Combined Dataset

For this task, we had to use all the three datasets together. The collective information obtained from the three datasets is used to train ML models so we can infer whether using more the combined data has any advantage over using individual datasets.

3.1 Constructing optimal features

To construct features, our initial approach was to

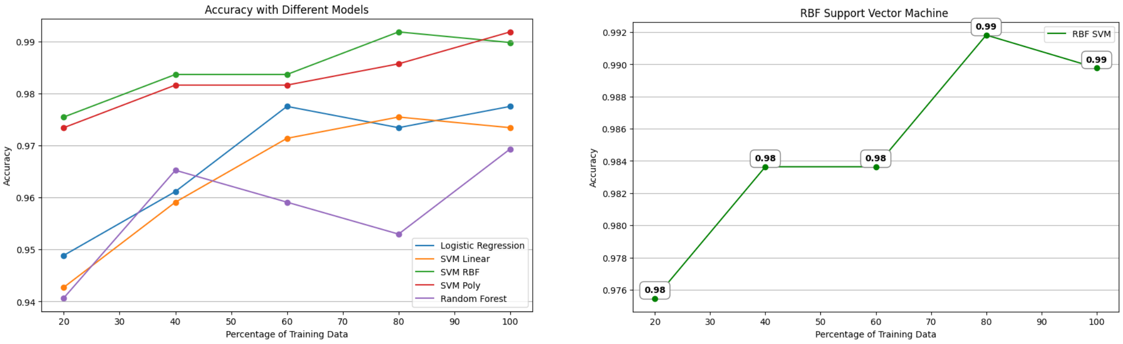
- Tokenize the emojis from dataset 1
- Flatten the inputs from dataset 2
- Split the text sequences from dataset 3 into an array of 50 digits.
- Concatenate the above three array for each input to get (10047,) sized array of features

But we later realized that most of the features in the input are from dataset 2 and this might cause model to give more importance to features from dataset 2. Hence we considered to use PCA to reduce the dimensionality of dataset 2 to 150 features. Now the concatenated combined input contains 213 features.

3.2 Models

We tried to fit the following models for this combined data

- SVM with Linear Kernel
- **SVM with RBF kernel(Best Model)**
- SVM with degree 2 kernel
- Logistic Regression
- Random Forest



(a) Accuracy vs Training data

(b) Performance of SVM with RBF kernel

Figure 7: Performance of various models with Text sequence dataset

From Figure 7, when we are training at 100% training data, SVM with degree 2 polynomial kernel gives more accuracy but this is weighed down by the fact that accuracies of RBF SVM are better at low training data meaning that RBF kernel SVM trains better for this data even when small amount of data is given.

3.3 Results

The RBF kernel based SVM model gave accuracies upto 99% on validation data when trained with complete training data. The confusion matrix for the same is below

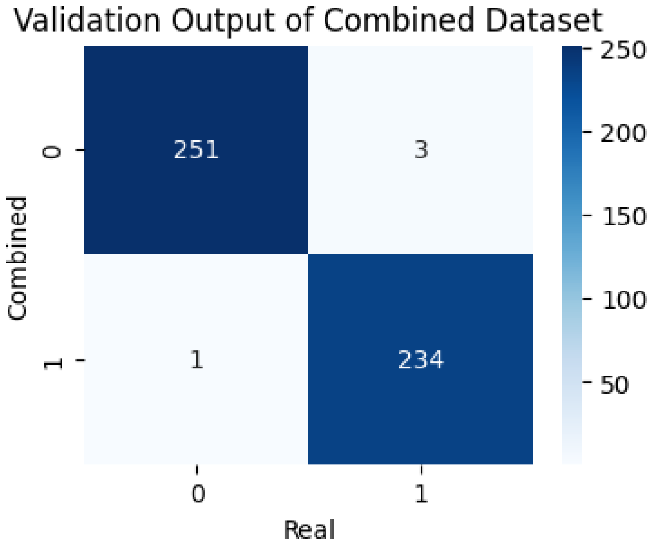


Figure 8: Confusion Matrix

4 Insights

In process of training various models for each of the datasets, these are some insights that we pondered upon

- Capturing patterns from the text sequences is difficult hence the low accuracy.
- For the emoticon data or the text sequence data, simpler models like SVMs or Random Forests failed to capture the relations among the inputs where as they were successful for Feature Dataset.
- As the feature dataset is extracted from a neural network, the features might be easier for even simple machine learning models to capture.
- However, when we used these models on the combined dataset, high accuracies were obtained. Considering the fact that the same model *failed* on emoticon and text sequence model, the difference comes with the prescence of feature dataset's contribution. This contributes to the most of the accuracy but more importantly the accuracy obtained is more than that obtained using feature dataset marking the importance of other two datasets.

Libraries:

We have used the following standard python libraries:

- numpy - 1.24.3
- Pandas - 2.2.3
- Tensorflow - 2.16.2
- Sklearn - 1.3.2