MODEL IMPLEMENTATION REPORT

#Group: Learners

#Roll numbers: 220282, 221016, 220659, 220878

Introduction:- The goal of this mini-project was to work with three binary classification datasets, each representing different feature representations extracted from the same raw data. We developed models to classify these datasets individually and as a combined dataset, and we analyzed the performance with varying amounts of training data.

The datasets used were:
1. Emoticons as Features Dataset (categorical features)
2. Deep Features Dataset (dense embeddings)
3. Text Sequence Dataset (sequences of numbers)

In Task 1, we focused on developing separate models for each dataset, while in Task 2, we combined the datasets to train a unified model.

Description: This dataset consists of categorical features representing emoticons.

Modelling Approach: We first split the input emoticons to 13 different columns. Then we applied different encoding or vectorization techniques like one hot encode. We applied several machine learning models to this dataset, including Logistic Regression, Decision Tree, Random Forest, XDBoost, SVM Linear, SVM RBF, and Naive Bayes. Logistic Regression with one hot encode emerged as the best performing model due to the simplicity of the data.

Results:

1. Logistic Regression provided the highest validation accuracy.
2. The model's accuracy was stable across different training data sizes, showing generalization even with limited data.
3. We decided to use 90% training data with logistic regression to get the final prediction of test dataset.

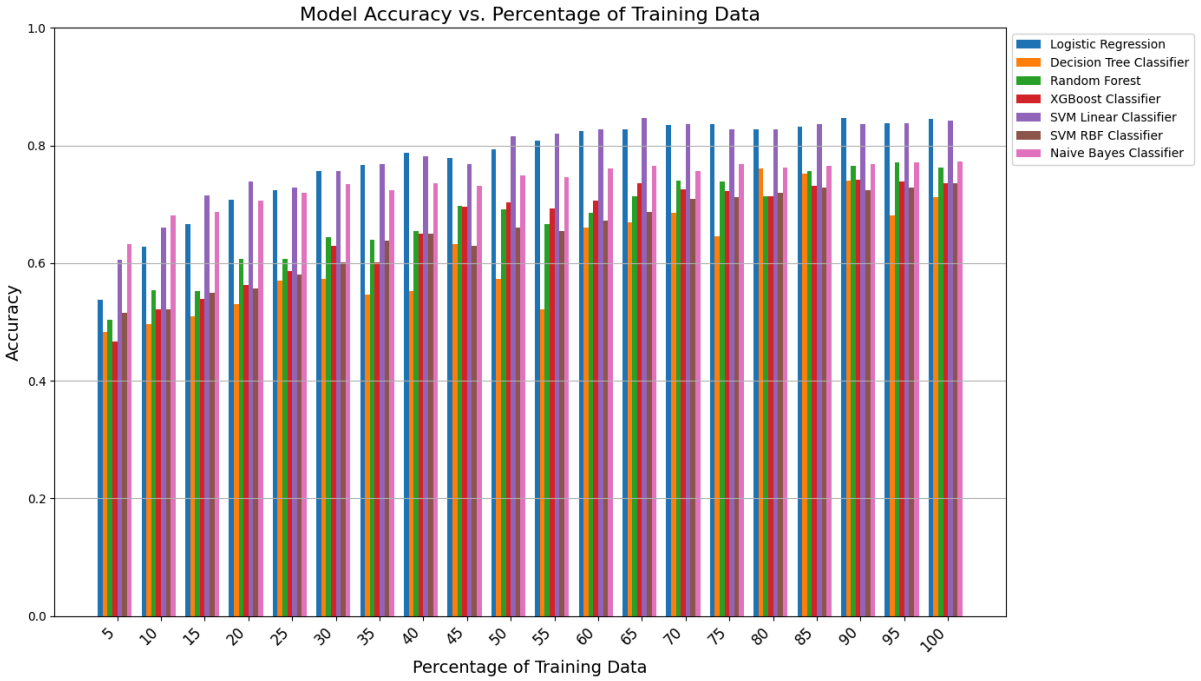
Observations:

1. Simpler models such as Logistic Regression and SVM Linear Classifications worked well for this dataset.
2. The dataset did not require much feature transformation, making it suitable for straightforward models.

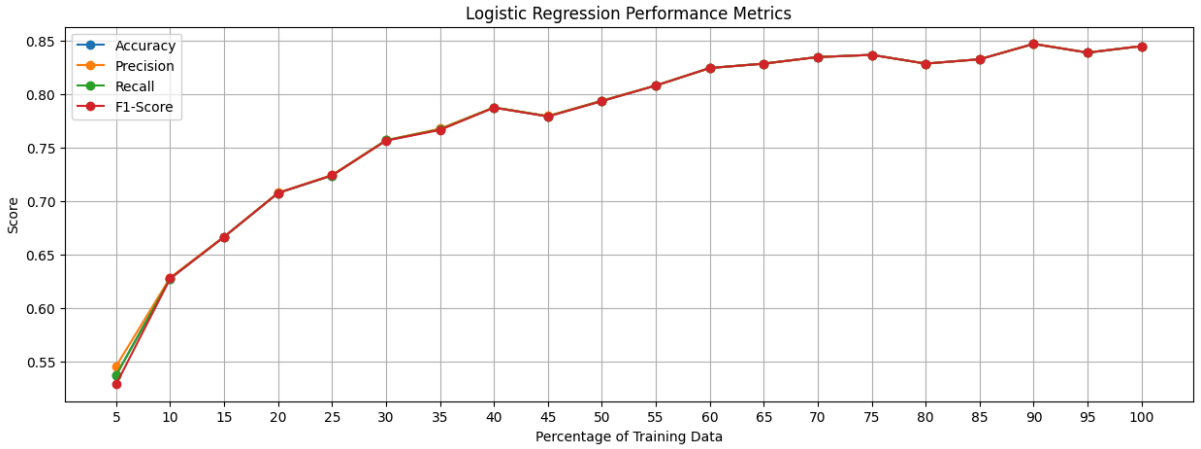
MODEL IMPLEMENTATION REPORT

Plot: Accuracy vs Training Size

1. We can clearly see that, with lower training size data, XGDBoost gives accuracy of about 70%, but it might be a fluke. As data size increases, Logistic regression gives best accuracy of more than 80%.



2. For 90% of training data, we get 84.66% accuracy which is pretty good.



MODEL IMPLEMENTATION REPORT

Deep Features Dataset

Description:
The dataset includes dense features, a matrix of 13x786 for each sample. These deep embeddings represent more abstract features compared to the Emoticons dataset.

Modeling Approach:
We applied several machine learning models to this dataset, including Logistic Regression, Decision Tree, Random Forest, XDBoost, SVM Linear, SVM RBF, and Naive Bayes.

Results:

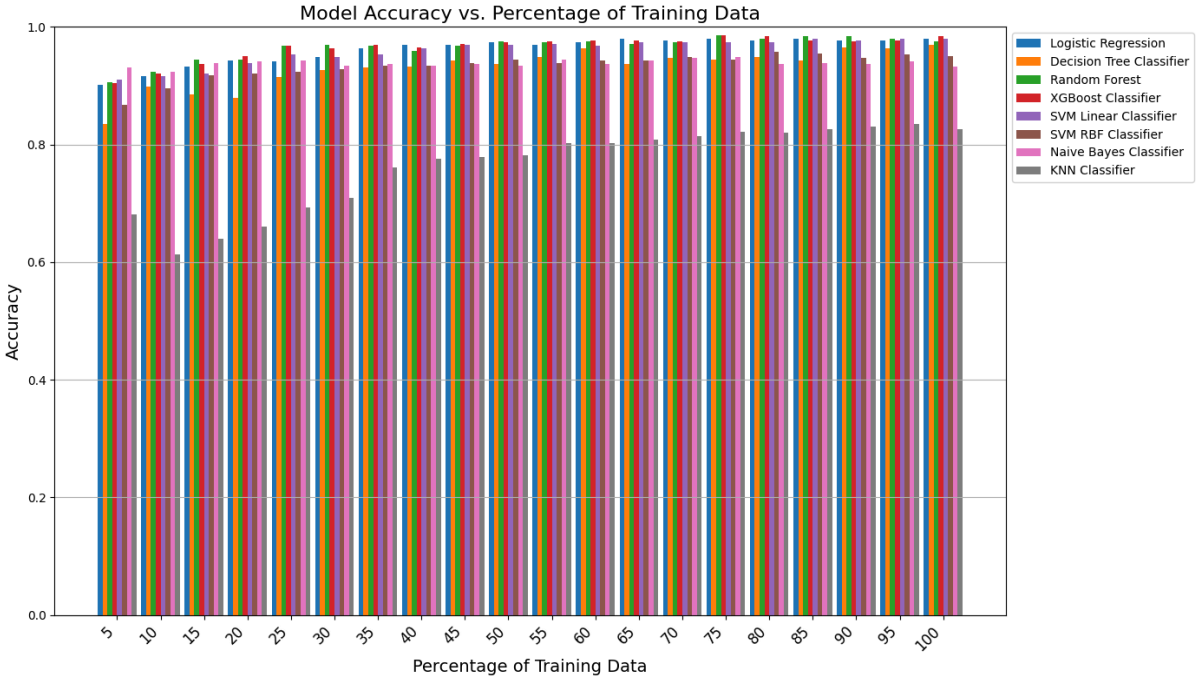
1. Random Forest with 75% of training data size gave highest accuracy of 98.57%.

Observations:

1. Almost all models performed very well for this dataset.
2. This shows that detailed data like this performs very well.

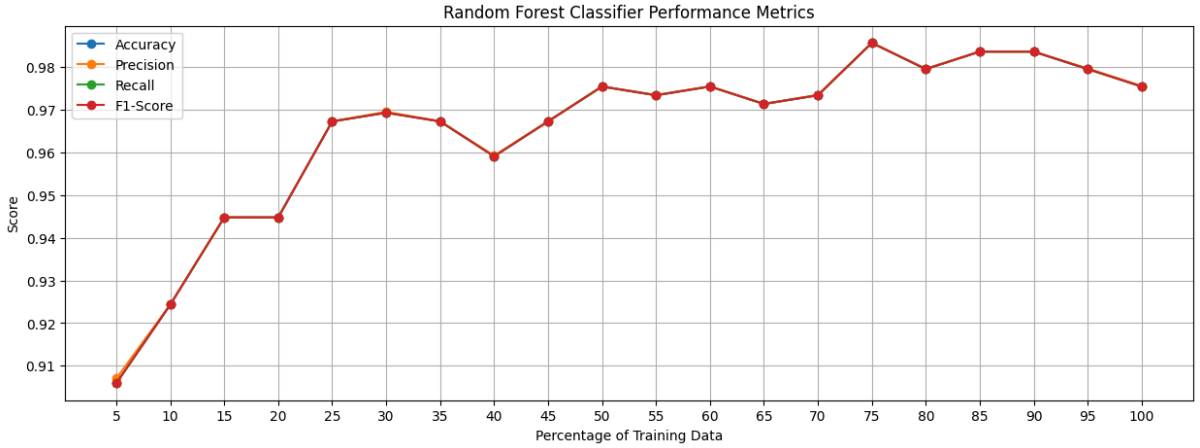
Plot: Accuracy vs Training Size

1. The accuracy increased steadily with larger training sizes. All models except KNNs achieved pretty high accuracy even with as low as 5% of training data size.



MODEL IMPLEMENTATION REPORT

2. Random Forest being the best has over 98% accuracy with higher training data size. While, 5% of training data also achieved about 90% accuracy.



Text Sequence Dataset

Description: This dataset contains sequences of 50 digits. These sequences were tokenized and vectorized for input into machine learning models.

Modeling Approach: Similar to emoticons data set approach, we first separated input string to 50 columns and then applied one hot encoding. We applied several machine learning models to this dataset, including Logistic Regression, Decision Tree, Random Forest, XDBoost, SVM Linear, SVM RBF, and Naive Bayes.

Results:

1. XG Boost got highest accuracy of 66.67% with 85% of data.
2. The sequence data was more complex, and thus, the models needed more data to generalize well.
3. We can use deep learning approaches to get better outcomes.

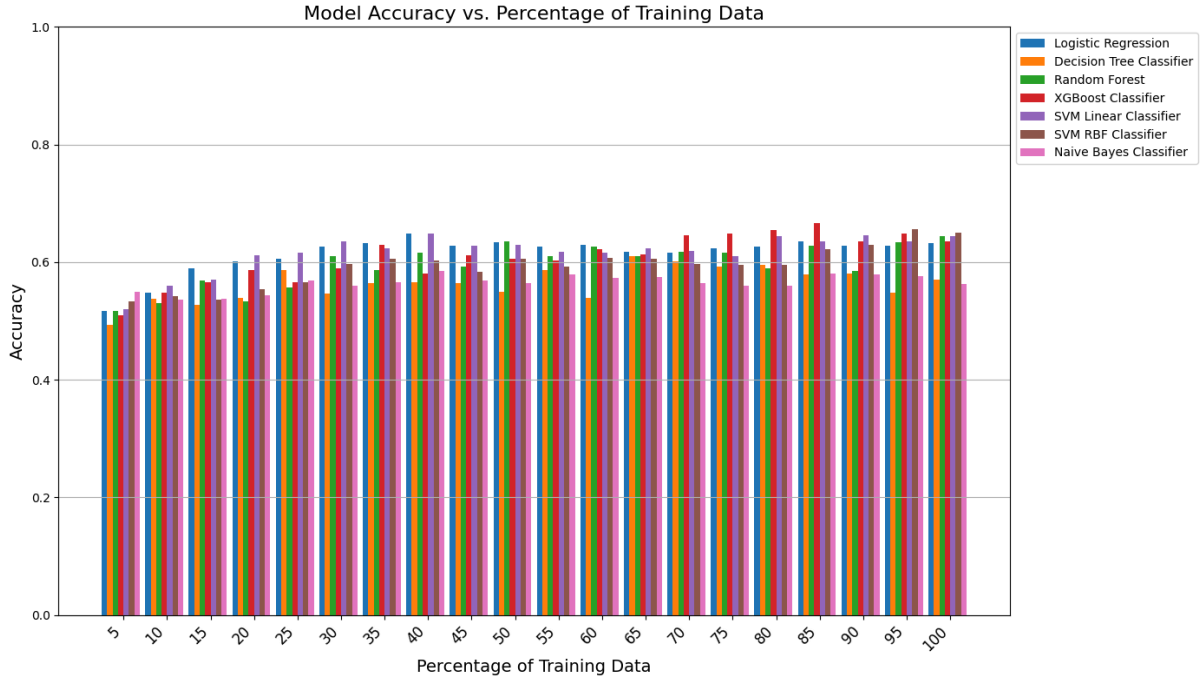
Observations:

1. Small training sizes resulted in poor generalization.
2. Even larger training sizes and many complex algorithms failed to achieve 70% of accuracy

MODEL IMPLEMENTATION REPORT

Plot: Accuracy vs Training Size

1. The model's accuracy improved sharply as the training data increased from 40% to 100%, highlighting the importance of data size for sequence models.



Task 2: Combined Dataset Analysis

In this task, we combined all three datasets. We used voting approach to get the result. All three previous models, voted and the label with most votes got the win.

Modeling Approach: We combined the features from the Emoticons, Deep Features, and Text Sequence datasets and voted using all three models as discussed above.

Results:

1. The combined dataset performed better than the individual datasets, as the model could learn from multiple feature representations.
2. Accuracy increased from previous models

Observations:

1. Combining datasets enriched the feature space, providing the model with more comprehensive information.

MODEL IMPLEMENTATION REPORT

Plot: Accuracy vs Training Size

1. The combined model achieved the highest accuracy when 100% of the data was used, and it showed improved performance even with smaller training sizes compared to the individual datasets.

Conclusion:

1. Emoticons Dataset: Logistic Regression
2. Deep Features Dataset: Random Forest
3. Text Sequence Dataset: XD Boost

Impact of Training Size: Increasing the training size consistently improved the models' performance across all datasets.The Emoticons dataset reached saturation earlier, while the Text Sequence dataset required more data for significant improvements.

Unified Model Performance: Combining the datasets resulted in better overall accuracy, demonstrating the advantage of leveraging multiple feature representations.

Deliverables:-

1. **Predicted Labels**: Predictions for each dataset (Emoticons, Deep Features, Text Sequence, and Combined) will be provided.

2. **Visualizations**: Plots showing the accuracy vs training size for each dataset, along with confusion matrices for the best models.