Ravi Rajpurohit 1002079916

# —---------- ML Project 2 --------------

*PLEASE REFER TO THE IPYNB FILE for better comments*
*Comments are provided and explained with Markdowns*

1. **Softmax Regression with Bagging -**
   Implemented a bagging routine for a softmax regression classifier. I have created the softmax regressor again from scratch for the scope of improvement.
   Applied the bagging 10, 50, and 100 times to the training data to find out the following observations -
   **Observations -**
   The results indicate that the single classifier has an accuracy of 83.33% and an error rate of 16.67%. When using bagging with 10 iterations, the accuracy decreases slightly to 80.56% and the error rate increases to 19.44%. However, when using bagging with 50 or 100 iterations, the accuracy remains the same as the single classifier at 83.33%, and the error rate also remains the same at 16.67%.

   This suggests that bagging with a larger number of iterations can help improve the accuracy of the model while reducing the variance. However, if the number of iterations is too small, it may lead to overfitting and decrease the model's accuracy. Overall, the results indicate that bagging can be a useful technique to improve the accuracy and robustness of machine learning models.

   But for a different random state, I got the results as - the accuracy of a single classifier is 63.33%, while the accuracy of bagging with 10, 50, and 100 classifiers is 76.67%, 80.0%, and 80.0%, respectively. This increase in accuracy can be explained by the fact that bagging helps to reduce overfitting by training each classifier on a different subset of the data. By combining the predictions of multiple classifiers, we can reduce the variance of the model and improve its generalization performance on unseen data.

   However, it's important to note that the results obtained may vary depending on the dataset and the choice of hyperparameters. In practice, it's recommended to perform a thorough hyperparameter tuning to achieve the best performance.

2. **Softmax Regression Classifier with Boosting -**

Implemented a boosting routine for the same softmax regression classifier. I have reused the softmax function implementation from question 1.

Applied the boosting 10, 25, and 50 times to the training data to find out the following observations -

**Observations -**

The results of the evaluation show that the AdaBoost ensembles significantly outperformed the single classifier. The single classifier had an error rate of 22.22%, while the AdaBoost ensembles with 10, 25, and 50 boosting rounds all had an error rate of ~15%. This is a substantial improvement in accuracy, indicating that AdaBoost is a powerful technique for improving the performance of machine learning models.

Additionally, we can see that the performance of the AdaBoost ensembles does not seem to significantly improve beyond 25 boosting rounds, as the error rate remains consistent for 25 and 50 boosting rounds. This suggests that further boosting may not be necessary and could potentially lead to overfitting.

3. **K - Means Clustering -**

   Implemented the K-Means clustering algorithm for the values of k = 3,6,9. Used the unlabelled dataset to make it into an unsupervised clustering problem. Calculated the overall accuracy for each model using the weighted sum of each cluster as instructed.

   **Observations -**

   ***Before scaling the features -***

   The results of the K-Means clustering algorithm show that the accuracy increases with an increase in the number of clusters, as expected. With K=3, the overall accuracy of the algorithm is 43.33%, which is not very high. This is likely due to the fact that there are three distinct material types in the dataset, which may not be easily separable into just three clusters.

   With K=6, the overall accuracy increases to ~46%. This suggests that some of the overlap between the different material types is being captured by the algorithm, but there is still some confusion between the different clusters.

   Finally, with K=9, the overall accuracy increases further to ~55%. This suggests that the additional clusters are helping to better capture the different material types and reduce the confusion between them.

   Overall, the results suggest that K-Means clustering can be effective at identifying the different material types in the dataset, but that a larger number of clusters may be needed to achieve high accuracy. Additionally, it's worth noting that the accuracy of K-Means clustering is limited by the intrinsic separability of the data, which may not be perfect in all cases.

***After scaling the features -***

Scaling the data had a significant impact on the performance of K-Means clustering. The overall accuracy increased for all values of K, which suggests that scaling improved the clustering results.

In particular, the accuracy of the K-Means clustering (K=9) increased from 55% to 74% after scaling. This is a substantial improvement and indicates that the clusters are better aligned with the true labels.

Scaling is an important preprocessing step for many machine learning algorithms, as it can help improve the performance and stability of the models. In this case, scaling helped K-Means clustering to better capture the structure of the data and produce more accurate clusters.

***References -***

I have referred to the following articles in order to understand the nuances of softmax regression, bagging and boosting techniques.

https://towardsdatascience.com/ml-from-scratch-logistic-and-softmax-regression-9f09f49a852c

https://www.geeksforgeeks.org/bagging-vs-boosting-in-machine-learning/#

https://machinelearningmastery.com/implement-bagging-scratch-python/