Logo, company name

Description automatically generated

**CSE523 Machine Learning**

**Prof. Mehul Raval**

**Group number: 17**

**Group name: The Mandelbrot set**

| **Name** | **Enrolment Number** |
| --- | --- |
| Aastha Gaudani | AU2040032 |
| Khushi Patel | AU2040068 |
| Devyash Shah | AU2040152 |
| Simran Khoja | AU1910606 |

This week we trained a CatBoostClassifier on our dataset using TF-IDF features.

First, we split the dataset into train and test sets. The train set will be used to train the model, while the test set will be used to evaluate the model's performance. Then, we preprocessed the text data by removing stop words and performing lemmatization.

Later, we extracted features from the text data using the TF-IDF vectorization technique. This creates a matrix of numerical features that can be used to train the model. After that we defined a CatBoostClassifier with a set of hyperparameters.

*CatBoost is a gradient boosting algorithm that is commonly used for classification tasks. CatBoost's name, which stands for "category," emphasizes the algorithm's aptitude at handling categorical features. It employs "ordered boosting," a kind of gradient boosting that may use both category and numerical variables. While using the gradient boosting method, ordered boosting takes into account the categorical variables' order. It also has the benefit of handling missing values in the data without the need for imputation. Moreover, it includes regularization strategies built in to avoid overfitting.*

*The text classification work being carried out is a multi-class classification task, where each instance might belong to one of multiple categories. As a result, the "MultiClass" loss function is employed in this code. For classification jobs, the CatBoostClassifier offers a number of loss functions, with the 'MultiClass' loss function being created especially for multi-class classification problems.*

Then we trained the CatBoostClassifier using the training data and made predictions on the test data using the trained model. Finally, we evaluated the model's performance using the accuracy score and the confusion matrix. The snapshot of that is given below. We got 94.62% accuracy, a little lower than last week.

