Major Project Report

Khushi Maheshwari (B21AI052)
Jiya Kumawat (B21CS036)
Riya (B21EE085)

Importing dataset:
We chose project 12 (Text Classification on Emails). First we mount the google drive and import the given dataset 'Text Classification on Emails'. We also import important libraries.

Data Preprocessing:
First, the text is transformed to lowercase and any non-Latin characters are removed using a regular expression. Next, we use the WordNetLemmatizer tool from the Natural Language Toolkit (nltk) to reduce the remaining words to their base or root form. We then remove stop words from the list of lemmatized words produced, which are commonly used terms in a language that lack significant meaning on their own (such as "the," "a," and "an"). Finally, we reassemble the remaining words into a single string. This process helps to simplify the text and make it easier to analyze by removing unnecessary words that don't contribute much to the overall meaning.

We got the top 12 most frequent words from all four categories and created a bar plot of the top 12 words with the words on the x-axis and the counts on the y-axis.

We then did the text vectorization using the TfidfVectorizer. The purpose of text vectorization is to convert raw text data into a numerical representation that can be used as input for machine learning models.

We Applied label encoder on "class". We selected the rows from data where the value of 'Class' is equal to 0, 1, or 2 (which indicates crime, politics, and science topics respectively). Then we took filtered text for each subject category from the data and then merged them into a single string using " ".join(). For each subject group, this results in a single string of preprocessed text data. After that we made a list named contents containing all the preprocessed text data .

We created a function named gen_wordcloud that generates a word cloud image of a fixed size from input text data and by using this function we generated word cloud image of crime content, politics content and science content.

Then we Splitted the data into a training and validation set.

Why we removed entertainment :
Because all the content was incorrectly classified as entertainment. Possible reason can be that entertainment content contains a significant amount of common words, and the model is not able to differentiate between the features that are specific to entertainment content and those that are specific to science,crime and politics content. Due to this the model may overfit on the entertainment features and not generalize well to the science,crime and politics content.

Logistic Regression:

Firstly, we made a list of various hyperparameter values to search over. We initialized an initial score of negative infinity to compare the performance of different hyperparameters .By using nested for loops we created an instance of logistic regression model using the hyperparameter and trained the model on training data and made predictions using validation data and calculated accuracy over each possible combinations of hyperparameters.The best score and best hyperparameters are updated if the current score surpasses the prior best score.

Then we created a logistic regression model with the best hyperparameters obtained and performed 5 fold cross validation then calculated the mean, standard deviation of cross validation score.
Finally, we fit the training data into this model and then made predictions using validation data and calculated accuracy.

At last, we plotted a heatmap of the confusion matrix for each class.

Multinomial Naive Bayes and Decision Tree Classifier:

Firstly, we made a list of various hyperparameters for respective models values to search over. Then we tuned the hyperparameters in the same way as we did in logistic regression for all models.

Then we created an instance of the respective model with the best hyperparameters obtained and performed 5 fold cross validation then calculated the mean, standard deviation of cross validation score.
Finally, we fit the training data into this model and then made predictions using validation data and calculated accuracy.

At last, we plotted a heatmap of the confusion matrix for each class.

SGDClassifier with Squared Hinge Loss :

We created an instance of SGD classifier with parameters like loss, penalty,alpha,max_iter etc. Trained the model using training data and made predictions using validation data then calculated accuracy.
At last, we plotted a heatmap of the confusion matrix for each class.

XGBoost :

We created an instance of XGBoost classifier with parameters like subsample etc.
Trained the model using training data and made predictions using validation data then calculated accuracy.

KNeighborsClassifier:

Firstly, we made a list of various hyperparameter values to search over. Then we tuned the hyperparameters in the same way as we did in logistic regression.

Then we created an instance of the KNN model with the best hyperparameters obtained and performed 5 fold cross validation then calculated the mean, standard deviation of cross validation score.
Finally, we fit the training data into this model and then made predictions using validation data and calculated accuracy.

At last, we plotted a heatmap of the confusion matrix for each class.

Voting Classifier:

Hard voting:
We created a voting classifier using hard voting(final prediction is determined by majority voting) with 5 different base models: logistic regression, Multinomial Naive Bayes, decision tree, k-nearest neighbors, and XGBoost. Then, it fits the voting classifier on the training set and calculates the accuracy on the validation set.

Soft voting:
We created a voting classifier using Soft voting (considers the predicted probabilities of each classifier and averages them to make the final prediction) with the same 5 models that we used in hard voting and then fit the data and calculated accuracy in the same way as we did in hard voting.

Comparison:
We compared the accuracy of each model on validation set and the result we got is:

Classifier	**Accuracy**
Logistic Regression	96.5831
Multinomial Naive Bayes	96.5072
Decision Tree Classifier	70.3113
SGD Classifier	93.6978

Xgboost	92.027335
K Neighbours Classifier	92.5588
Hard Voting	96.355
Soft Voting	96.051632

After observing the above result we conclude that logistic regression has best accuracy on validation dataset .

Testing on Unseen Dataset:
We take 5 emails as unseen dataset and preprocess the emails as we had done in our training and validation dataset.After that we evaluate our unseen data on our previously trained model and compare the result.We observe that all aur models give approximately 80% accuracy.

	Text	**Prediction**
0	Science_1	Science
1	Politics_1	Politics
2	Politics_2	Politics
3	Crime_1	Crime
4	Crime_2	Science

<u>Contribution of each individual:</u>

Khushi Maheshwari (B21AI052):

- Did the hyperparameter tuning for Logistic Regression and finally applied the Logistic Regression and calculated the accuracy score.
- Did the hyperparameter tuning for MultiNomial Naive Bayes and finally applied MultiNomial Naive Bayes and calculated the accuracy score.
- Applied XGBoost with Decision tree and calculated the accuracy.
- Contributed in Testing on unseen data

Riya (B21EE085):

- Did the appropriate pre-processing of the dataset.
- Did the hyperparameter tuning for KNN and finally applied the KNN classifier and calculated the accuracy score.

- Applied SGDClassifier with Squared Hinge Loss and calculated the accuracy.
- Contributed in Testing on unseen data

Jiya Kumawat (B21CS036):

- Did the appropriate pre-processing of the dataset.
- Did the hyperparameter tuning for Decision Tree Classifier and finally applied the Decision Tree Classifier classifier and calculated the accuracy score.
- Applied ensemble learning using soft voting and hard voting classifiers.
- Contributed in Testing on unseen data