Assignment 1: Text Classification Task

Objective: Develop a simple text classification model using machine learning.

Dataset:- 20 Newsgroups

Dataset Description:-Dataset contains around 20,000 newsgroup documents categorized into 20 different classes.

Classes are as follows:-

['alt.atheism', 'comp.graphics', 'comp.os.ms-windows.misc', 'comp.sys.ibm.pc.hardware', 'comp.sys.mac.hardware', 'comp.windows.x', 'misc.forsale', 'rec.autos', 'rec.motorcycles', 'rec.sport.baseball', 'rec.sport.hockey', 'sci.crypt', 'sci.electronics', 'sci.med', 'sci.space', 'soc.religion.christian', 'talk.politics.guns', 'talk.politics.mideast', 'talk.politics.misc', 'talk.religion.misc']

Sample data of the newsgroup_data:-

```
Class name --->  7 alt.atheism
Data:-
 I was wondering if anyone out there could enlighten me on this car I saw
the other day. It was a 2-door sports car, looked to be from the late 60s/
early 70s. It was called a Bricklin. The doors were really small. In
addition, the front bumper was separate from the rest of the body. This is
all I know. If anyone can tellme a model name, engine specs, years
of production, where this car is made, history, or whatever info you
have on this funky looking car, please e-mail.
```

Preprocessing of Dataset:

Here I have chosen “spaCy” lib for text preprocessing for removal of Stopwords and punctuations. In spaCy there are many models with many sizes among which I have chosen “en_core_web_sm” which is a small size english model containing tagger, NER, parser but no vector component.

Example of preprocessed data:-

```
Newsgroup Data:-  I was wondering if anyone out there could enlighten me
on this car I saw the other day. It was a 2-door sports car, looked to be
from the late 60s/ early 70s. It was called a Bricklin. The doors were
really small. In addition, the front bumper was separate from the rest of
the body. This is all I know. If anyone can tellme a model name, engine
```

```
specs, years of production, where this car is made, history, or whatever
info you have on this funky looking car, please e-mail.
----------------------------------------
Preprocessed data:-  wondering enlighten car saw day 2 door sports car
looked late 60s/ early 70s called Bricklin doors small addition bumper
separate rest body know tellme model engine specs years production car
history info funky looking car e mail
```

Vectorization:

As ML models understand only the numerical representations of data, we convert our dataset into vectors representing the numerical values. TF-IDF vector is commonly used for this purpose. It represents documents as vectors by assigning weights to words based on their frequency within a document and uniqueness across the corpus.

Train a classifier:-

Naive Bayes is a family of simple yet effective probabilistic classifiers based on Bayes' Theorem. It assumes that the features (words in text classification) are independent of each other given the class label, hence the term "naive." Despite this strong assumption, it performs well in practice for various tasks like spam filtering, sentiment analysis, and document classification. It calculates the probability of each class given the features and predicts the class with the highest probability. Naive Bayes is fast, easy to implement, and works well with large datasets.

Support Vector Machine (SVM) is a powerful supervised learning algorithm used for classification and regression tasks. It works by finding the optimal hyperplane that best separates data into different classes. The goal is to maximize the margin (distance) between the closest points (support vectors) of each class to the hyperplane. SVMs are effective in high-dimensional spaces and are robust to overfitting, especially when using techniques like the "kernel trick" to handle non-linear data. SVMs are widely used for text classification, image recognition, and bioinformatics due to their accuracy and efficiency.

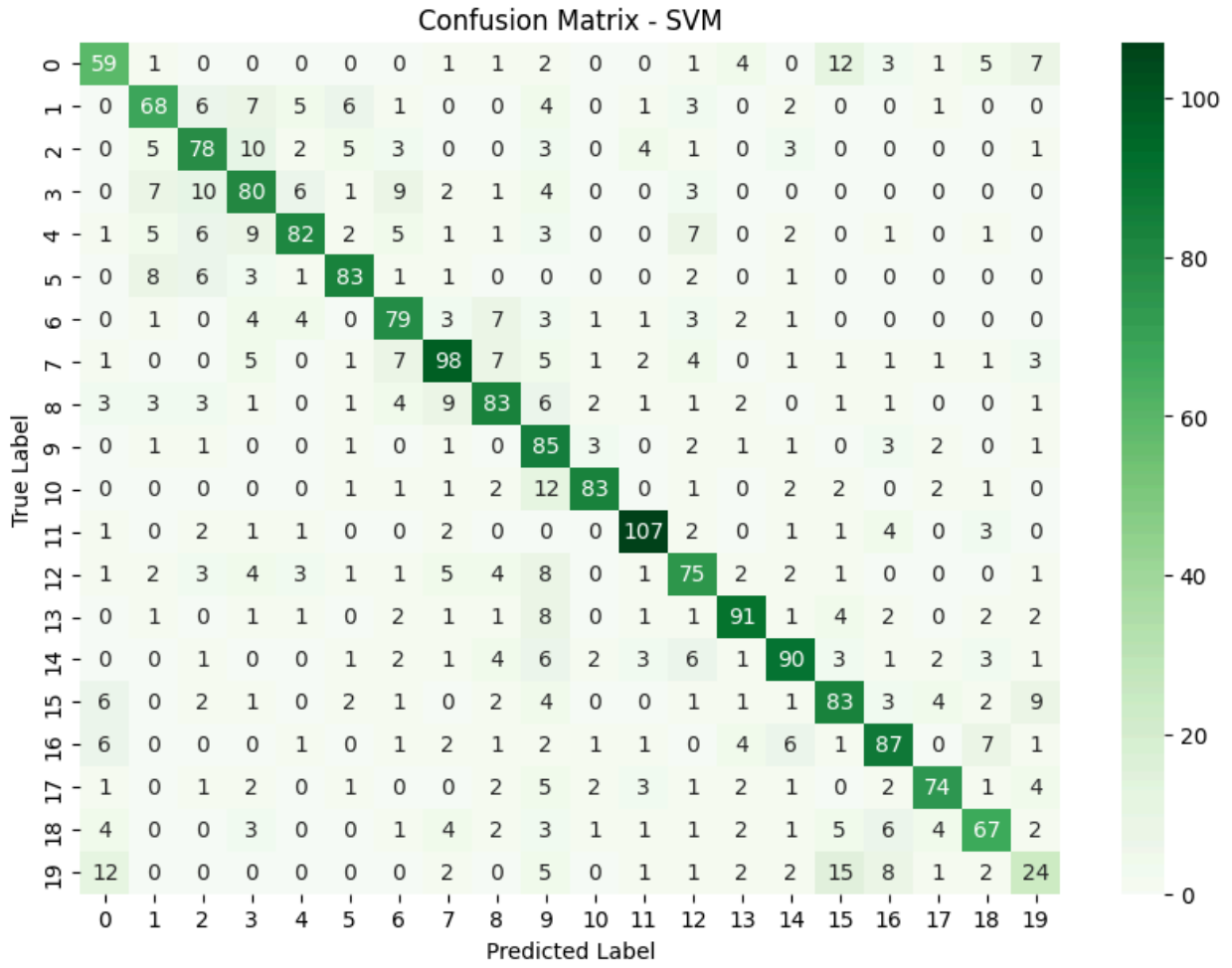
Evaluation:-

Model Name	**Precision**	**Recall**	**F1**
Navies Bayes	0.70	0.68	0.67
SVM	0.70	0.69	0.69

Precision is slightly the same for both the models. It tells that both models are good at minimizing false positives across all the classes.

Recall for SVM is higher than Navies, SVM is better in capturing all the true instances compared to Navies. SVM is better in minimizing the false negative and ensuring to capture most of the classes correctly.

F1 score is better in SVM, which says that there is a balance between Precision and Recall which makes it better for Classification for all the classes across.



- The Diagonal elements represent the number of correctly predicted instances across each class.
- For many classes, the diagonal elements are high indicating the classes are correctly classified.
- Class 11 has value 107, indicating that most of the values in the class have been predicted correctly.
- Class 10 also performs well with value 83.
- There are also many misclassifications, for instance class 19 is misclassified as 15, 18 etc. This might be due to similar features .
- The classifier performs well for a majority of classes, with strong diagonal dominance for many rows and relatively low off-diagonal values.
- Certain classes are challenging for the classifier, as indicated by their widespread misclassifications across multiple predicted labels.