Task 3.1:

Preprocessing data is very important when it comes to machine learning, as a model fails to get trained properly otherwise. Few methods followed for preprocessing text are:

1. Tokenization: By the method of tokenization, which itself has subparts like word tokenization and sentence tokenization, the text data is broken down into individual components as has been desired by the tokenizer method used, so as to allow further preprocessing steps to be conducted.
2. Stemming and Lemmatization: This is done in order to make sure that different versions of the same words that have been created by word tokenization do not add on to the number of words required for training the model. Using stemming, the process of shortening the word to root becomes faster, but is not very accurate, unlike lemmatization, which converts the word to its root, in such a way that it is grammatically correct.
3. Removing stop words: For models, especially ones which are used for various sorts of analysis based on context, removing stop words becomes a necessity, as the words like ‘and’, ‘the’, etc. occur multiple times, making it difficult for the model to predict the context or the main topic being discussed in the text.
4. Adjusting the case-sensitive problem: So as to ensure that same words are not considered to be different due to case-sensitivity, the text is converted into lower case (or upper case).
5. Removing punctuation marks: Punctuation marks usually add on to the length of the text, making model training unnecessarily time-consuming. Thus, they need to be removed.

Task 3.2: (Clustering)

Clustering is a way of dealing with unsupervised learning, wherein the dataset given is not at all labelled, i.e., the relationships between items in the data is not given explicitly. In this technique, the algorithm is made to find any sort of relation between the data and create groups/clusters accordingly. These clusters actually refer to group of data point having a large similarity with each other than with the data points outside the group. The evaluation of similarity is done based on a metric like Euclidean distance, Cosine similarity, etc.

In Clustering, the machine itself derives and extracts patterns hidden in the dataset, and thus this is a method whose complexity increases with increase in types of features in data, as finding similarity is convenient in a small group.

Clustering is particularly useful when the focus of the machine learning aspect we are using is to find out outliers, that is, data points that do not belong to a particular region.

Types of Clustering algorithms are:

1. K – Means clustering ->

In this method of clustering, grouping is done based on centroids. It tries to minimize the variance of data points within a cluster, and is a better option for small datasets. Here, data points of centroid of each cluster are used to find a data point at the centroid of newly formed cluster, and this process continues till a good cluster is obtained.

Working:

1. Randomly initialize ‘k’ centroids, for ‘k’ number of clusters
2. For each data point, find out the Euclidean distance from each centroid, and categorize each data type into the centroid which has the closest distance from it.
3. Move the centroid to the calculated mean found out by taking mean of coordinates of all newly categorized data points.
4. Repeat process till a good cluster is obtained.
5. DBSCAN ->

DBSCAN, short for Density-based spatial clustering of applications with noise, is a technique that is used to identify clusters by identifying areas of high density, and separating them from areas of low density.

This method involves three terms – core point (point lying in the region of high density), non-core points (points lying in the region of low density), and outliers (points that are not part of any region). There are two parameters needed for applying the DBSCAN algorithm, which are: epsilon (maximum distance between two data points so that they can be considered to be neighbors), and min\_samples (minimum number of neighbors a data point should have to be considered a core point.

Working:

1. Determine core points, based on how many neighbors a point has
2. Spreading the current cluster to other core points to form a new cluster. Here, if the point serves as a core point, it will be used for further propagation, else it will simply be marked as a core point. This method continues till all close points have been clustered.
3. All the non-clustered points are marked as outliers.

It is possible to tell at the beginning whether a point is a core point or no, however, we cannot accurately comment on whether it is a non-core point or an outlier unless the algorithm has been applied.