# Part 2.2 - Filtering target classes (4 points)

• 2.2.1. Print the name of classes in your training set along with selected_targets you can use target_names attribute of newsgroups_train.
Ans: **{1: 'comp.graphics', 7: 'rec.autos', 10: 'rec.sport.hockey', 13: 'sci.med', 15: 'soc.religion.christian', 16: 'talk.politics.guns', 17: 'talk.politics.mideast'}**

# Part 2.3 - Vectorizing documents (12 points)

• 2.3.1. What does TF-IDF stand for?
Ans: **Term Frequency - Inverse Document Frequency**
• 2.3.2. Why don't we only use term frequency of the words in a document as its feature vector? what is the benefit of adding inverse document frequency?
Ans: **When using only term frequency, it will only have values 1's and 0's. Only after adding the inverse document frequency do we only get the significance of the various values.**
• 2.3.3. Calculate the tf-idf vectors of the following two documents, assuming this is the entire corpus:
Ans:

| | this | is | a | another | sample | example |
|---|---|---|---|---|---|---|
| Sentence 1 | 0 | 0 | 0.4 * log(2) | 0 | 0.2 * log(2) | 0 |
| Sentence 2 | 0 | 0 | 0 | (2/7) * log(2) | 0 | (3/7) * log(2) |

# Part 3.1 - Sparsity (12 points)

In this section we will interpret the coefficients from the final model you trained on all of the training data.

• 3.1.1 Count the number of non-zeros in each row of the train_vec matrix.
Ans: **[89, 94, 217, 70, 190, ……. 345, 258, 342, 205, 124] (It is a list of 4081 elements, in the report we have just included the first five and the last five elements)**

• 3.1.2 What is the average number non zero elements in each row?
Ans: **The average number of non zero elements in each row is 170.56187209017398.**
• 3.1.3 On average what percentage of elements in each row have non-zero elements?

Ans: **On average, 0.30374489713848585 % of elements in each row have non-zero elements.**

# Part 3.2 - SVD (4 points)

Write out the result to a file called part_1.4_results.csv and submit this along with your assignment. (10 points) (You do not need to submit anything for your report for this part.)

• 3.2.1. What portion of the variance in your dataset is explained by each of the SVD dimensions?
Ans: **Portion of the variance in your dataset is explained by SVD training dimensions [0.01618638 0.00617073 0.00540306]**

# Part 3.4 - Visualization (8 points)

• 3.4.1. Based on your observation, what is the difference between SVD and UMAP embeddings? 1-2 sentences should suffice.
Ans: **The plot from UMAP had clusters evenly spaced out. Whereas in SVD, it was difficult to differentiate the clusters as they were cluttered.**

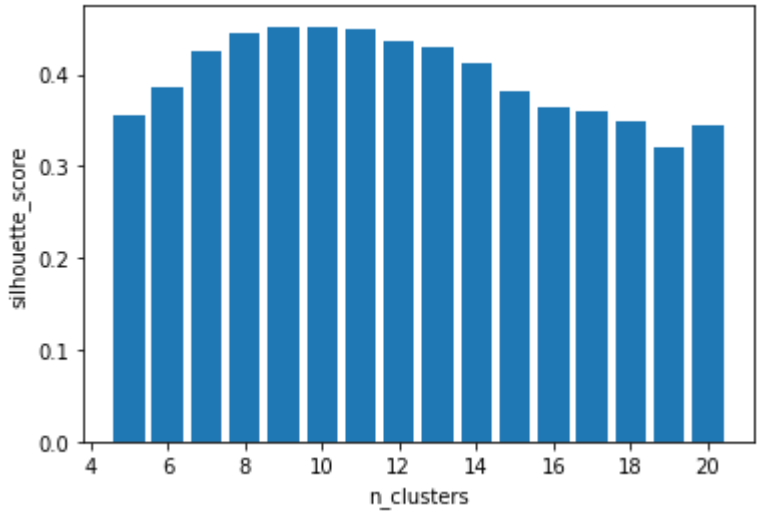
• 3.4.2. Which one do you prefer to use for a classification task? why? 1-2 sentences should suffice.
Ans: **We would prefer UMAP to SVD because from visualization we could see that UMAP clusters were more isolated and properly grouped when compared with SVD.**

# Part 4.1 - Clustering and evaluation (16 points)

• 4.1.1 What is the range of possible values of silhouette coefficients?
Ans: **Silhouette coefficients could be in the range -1 to 1.**
• 4.1.2 Describe what a silhouette score of -1 and 1 mean?
Ans: **Silhouette score's best value is 1 while it's worst value is -1. When the score is assigned to -1, it means that the sample has been assigned to the wrong cluster. However, when the score is assigned to 1, it means that the sample is within the right cluster.**
• 4.1.3. Use silhouette score and KMeans from sklearn library to find the optimum number of clusters in your train_umap. Don't forget to use SEED as your kmeans random_seed. In order to do this try different values of cluster numbers from 5 to 20. Choose the one that results in the best score.
Ans: **Selected number of clusters : 9**
**CPU times: user 19.2 s, sys: 3.07 s, total: 22.2 s**

**Wall time: 19.5 s**

• 4.1.4. Plot silhouette score for different values of n_clusters (a plot with n_clusters on the x-axis and silhouette score on the y-axis). Don't forget to put the plot in your report.




Ans:

# Part 4.2 - Making a Kmeans classifier (4 points)

• 4.2.1 show your mapping (resulted dictionary) inside your project report.

Ans: **{0: 7, 1: 1, 2: 10, 3: 13, 4: 17, 5: 16, 6: 15, 7: 15, 8: 17}**

# Part 4.3 - Analyzing clusters (12 points)

• 4.3.1. Are there any two clusters in your clustering output with the same original label (for example, are there two clusters which both have same training label)? Use your visualizations and describe why?

Ans: **Yes cluster's 4 and 8 both have training labels of 17. Cluster's 6 and 7 both have training labels of 15. For the label 17, there are outliers in the data that have been grouped together as a cluster. And for the label 15, points that lie on or near the cluster boundaries of two clusters have been grouped together to form a separate cluster.**

• 4.3.2. Write the function bellow that returns nearest samples to a cluster center. Use this function and explain why there are overlaps in your labels?

Ans:

```
def most_central_samples(clustering: KMeans, cluster_id, k=3):
    """returns the text of k most central samples in the specified cluster_id
    """
    # YOUR CODE
    centroid=clustering.cluster_centers_[cluster_id]
    print(centroid)
```

```
    centroid=np.reshape(centroid,(1,len(centroid)))
    points_on_cluster=train_umap[clustering.labels_==cluster_id]
    distances=cdist(centroid,points_on_cluster).squeeze()
    return train_umap[np.argpartition(distances,k)[:k]]
```

**Majority of the points in the data are very close to each other in 3 dimensional space, but even though umap does a great job of separating clusters out, the centroids of all the clusters are still close to each other. This is why there are overlapping labels.**

• 4.3.3. Can you infer the overlapping label(s) by checking out most central samples? check with original labels.
Ans: **Yes, we can take a look at most central samples from the outlier cluster, the original labels are different from the cluster id mapping. Thus, these points are general outliers from the data incorrectly clustered by k-means.**

# Part 4.4 - Evaluate your Kmeans model on test dataset (12 points)

• 4.4.2. Calculate the accuracy of model
Ans: **accuracy: 0.7152317880794702**
• 4.4.3. Calculate both micro and macro values of precision, recall and F1 score
Ans: **micro_precision_score: 0.7152317880794702**
**micro_recall_score: 0.7152317880794702**
**micro_f1_score: 0.7152317880794702**
**macro_precision_score: 0.7918735197238576**
**macro_recall_score: 0.7154683626200024**
**macro_f1_score: 0.7105467269613749**

# 574 ONLY Part 5.1 - KNN classification (16 points)

• 5.1.1. Train two seperate KNN models on both SVD and UMAP embeddings. Use n_neighbors=100.
• 5.1.2. Evaluate your model on test datas (test_umap and test_svd). Which model performs better? Why?
**Umap performs better because it increases variance between the clusters and decreases variance within the cluster, and this can be seen from the babyplot of umap. Since there is a clear separation of clusters it is able to classify better.**

• 5.1.3. Calculate macro and micro precision recall and fscore for test_umap. Which one of the two do you prefer for evaluating your model? Why?
Ans: **umap accuracy: 0.7759381898454746**
**umap micro_precision_score: 0.7759381898454746**
**umap micro_recall_score: 0.7759381898454746**
**umap micro_f1_score: 0.7759381898454747**
**umap macro_precision_score: 0.7842628754827806**
**umap macro_recall_score: 0.7754939245821235**
**umap macro_f1_score: 0.7767604825801012**
**We would prefer macro_f1_score as the metric over the micro score since the labels are not imbalanced and this is a multi-class classification problem.**

• 5.1.4. Shortly describe why the two sets of values (macro and micro) are so similar in this case.
Ans: **If there had been class imbalance in the dataset, macro score would not take this into account, and would have produced misleading results. But since this dataset has labels that are balanced both macro and micro are very similar.**

# Contribution Statement (Minus 15 points if you do not submit this)

Please describe what each group member contributed to this project. Note that the professor and TA reserve the right to challenge this statement, and falsification of effort will be considered a violation of Academic Integrity . That is, if we find reason to believe that a specific group member's claim about the work they contributed is not valid, then we reserve the right to take steps to ensure that the group member did, in fact, contribute as stated. We also reserve the right to adjust grades based on extreme differences in effort put into the assignment.

**Ann**

- **Made initial attempts**
- **Completed section 2.2.1, 3.1 - 3.3**
    - **There were some issues which Revanth fixed**
- **Also did section 4.1**
- **Handled all the theory questions - Some were answered by my teammates (those that involved coding from 3.4 and then from 4.2 onwards).**

**Revanth**

- **Fixed 2.2.1**
- **Fixed 3.1.3**
- **Fixed issues section 3.2 and 3.3**
    - **Issue 1 - Fit was done for both the test and train data which was wrong**
    - **Issue 2 - Explained_variance_ratio_ was called on numpy object instead of TruncatedSVD**
- **Plot graphs for 4.1 and find the best value for number of clusters**
- **Implement section 4.2.1**

**Sriram**

**Implemented section 4.3, 4.4 and 5.1**