KNN based Sentiment Classifier

This section of the report provides a detailed analysis of the K-Nearest Neighbors (KNN) classifier applied to our sentiment classification task. We assess its performance using both Bag-of-Words (BoW) and TF-IDF feature representations, and examine how key hyperparameters—particularly the number of neighbors (k) and distance metrics—affect the model's accuracy and overall effectiveness.

Text Vectorization Methods and Distance Metrics

TF-IDF (Term Frequency–Inverse Document Frequency):

TF-IDF is a text vectorization technique that evaluates the importance of a word in a document relative to a collection of documents (corpus). It assigns higher weight to words that are frequent in a particular document but rare across the corpus, reducing the influence of common words and highlighting distinctive terms.
Mathematical Formula:

$$\textbf{TF-IDF}(t, d) = \textbf{TF}(t, d) \times \textbf{IDF}(t)$$

Where:
TF(t, d) = (Number of times term t appears in document d) / (Total number of terms in document d)
IDF(t) = $\log (N / n_t)$
N = total number of documents
n_t = number of documents containing term t

Bag of Words (BoW)

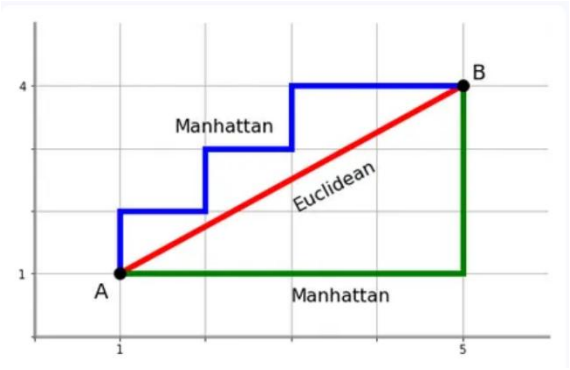
Bag of Words is a simple and widely used text vectorization method that converts text into numerical feature vectors based on word frequency. It disregards grammar, word order, and context, focusing only on the number of times each word appears in a document.
Mathematical Representation:
Given a vocabulary of size V, each document is represented as a vector:

$$X\_d = [f_1, f_2, f_3, \ldots, f\_V]$$

Where:
f_i is the frequency (count) of word i in document d

EUCLIDEAN DISTANCE

Euclidean Distance is a straight-line distance metric used to measure the direct distance between two points in multi-dimensional space. In text classification, it computes the root of squared differences between vectorized feature values, treating them as points in space.



Mathematical Formula:

$$d(p, q) = sqrt(\Sigma_i (p_i - q_i)^2)$$

Where:
p and q are two n-dimensional vectors (e.g. feature vectors of two text documents)
p_i and q_i are the ith components of the vectors

MANHATTAN DISTANCE

Manhattan Distance, also known as L1 distance or city-block distance, measures the absolute sum of differences between the coordinates of two points. It calculates the total distance traveled along axes at right angles.

Mathematical Formula:

$$d(p, q) = \Sigma_i |p_i - q_i|$$

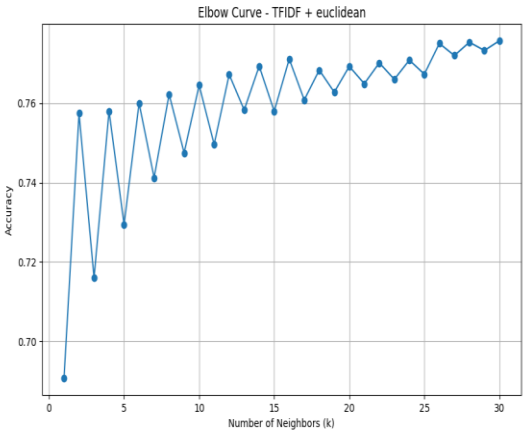
Where:
p and q are n-dimensional vectors
p_i and q_i are the ith components of the vectors

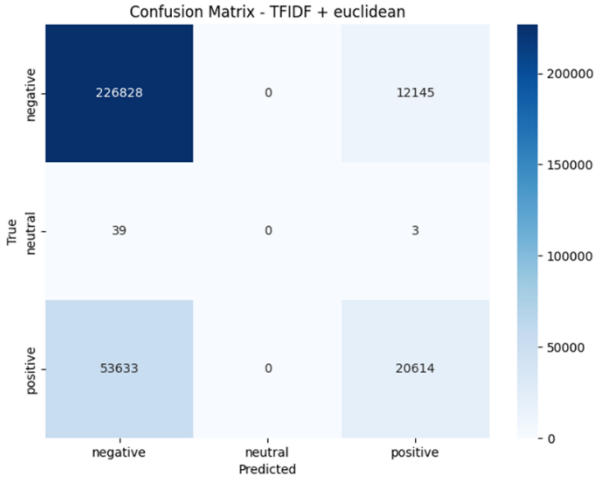
1. KNN WITH TF-IDF FEATURES

- **Using Euclidean Distance**

- **Accuracy :** 78.99%
- **Best_k :** 30

Elbow curve for best_K and confusion matrix:

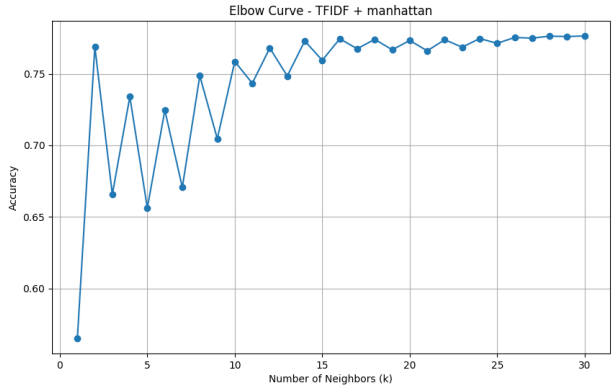


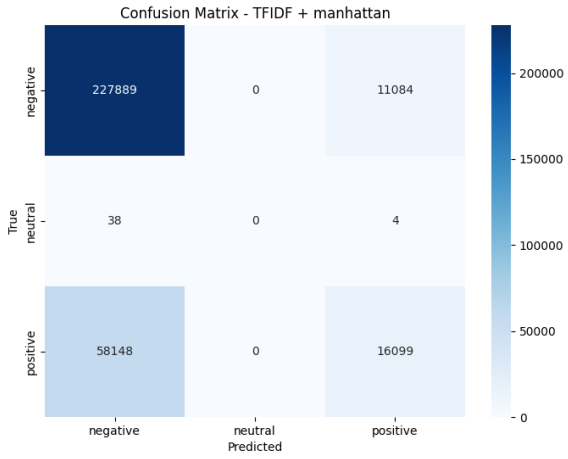


- **Using Manhattan Distance**

➢ **Accuracy :** 77.89%
➢ **Best_k :** 28

Elbow curve for best K and confusion matrix:



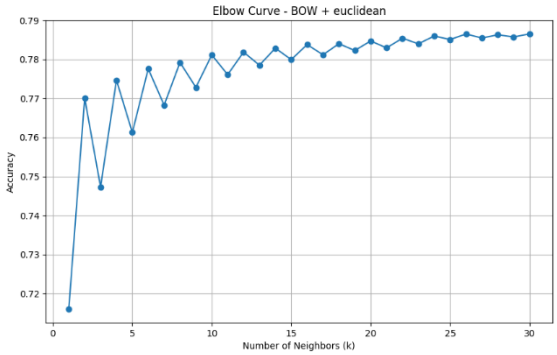


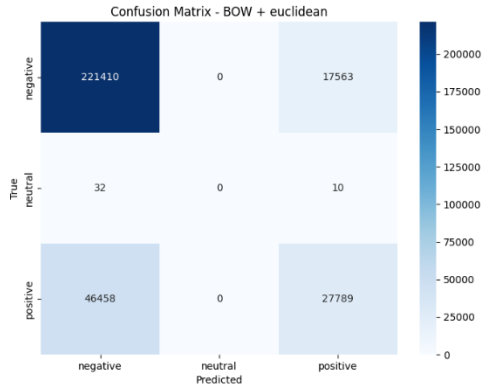
2. KNN WITH BOW FEATURES

- ### Using Euclidean Distance

- **Accuracy:** 79.54%
- **Best_k:** 30

Elbow Curve for best_k and confusion matrix:

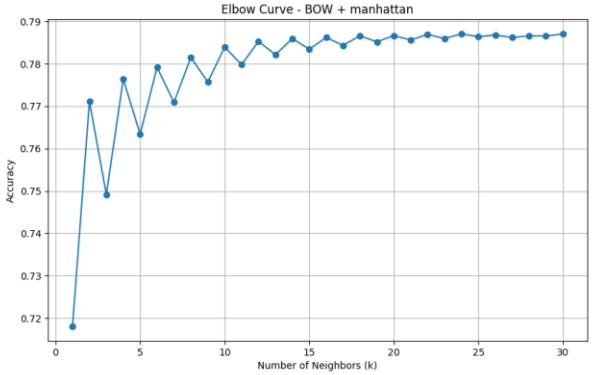


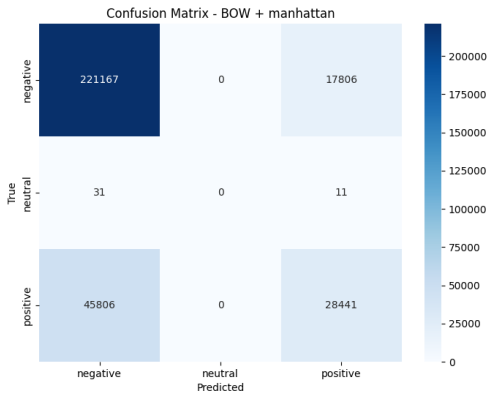


- ### Using Manhattan Distance

- **Accuracy:** 79.68%
- **Best_k:** 30

  Elbow Curve for best_k and confusion matrix:





INFERENCE FROM ELBOW CURVES :

To determine the optimal value of *k* for the K-Nearest Neighbors (KNN) classifier, the **Elbow Method** was employed. This method helps in identifying the point where increasing the number of neighbors no longer significantly improves model accuracy.
The procedure followed was:

- The dataset was initially split into **70% training data** and **30% testing data**.
- From the **70% training data**, a **10% stratified sample** was selected to efficiently perform hyperparameter tuning.
- For this sampled data, KNN models were trained by varying the value of *k* from **1 to 30**.
- This process was repeated separately for:
    - **Two vectorization techniques**:
        - TF-IDF (Term Frequency–Inverse Document Frequency)
        - BoW (Bag of Words)
    - **Two distance metrics**:
        - Euclidean Distance
        - Manhattan Distance
- For each combination, an **Elbow Curve** was plotted, displaying accuracy scores against the values of *k*.
- The point on the curve where the accuracy stabilized or started to decline was selected as the **optimal value of *k*** for that particular configuration.
- After k=26 in each of the above case for increasing K it was not increasing that it was ~0.78

Vectorizer	Distance Metric	Accuracy (%)	Best k
BoW	Euclidean	79.54	30
BoW	Manhattan	79.68	30
TF-IDF	Euclidean	78.99	30
TF-IDF	Manhattan	77.89	28

INFERENCE FROM CONFUSION MATRICES :

The **accuracy is nearly the same (~78%–80%) across all combinations** because the model's decision boundary isn't significantly impacted by small changes in vectorizer type (BoW, TF-IDF) or distance metric (Euclidean, Manhattan) when using **n-gram (1,2)** features.
 In the **confusion matrix**, the **'Neutral' column shows zero predictions**.
→ This is because the dataset is **highly imbalanced**:

- **Positive: ~77%**
- **Negative: ~23%**

- **Neutral: ~0.098%**

The model, driven by **majority class bias** in KNN, tends to favor Positive and Negative classes, leaving the **Neutral class entirely unpredicted**.

Misclassifications mainly occur between Positive and Negative classes, due to:

- **Semantic similarity in short or ambiguous tweets**
- The **minor influence of 2-grams** not being enough to capture contextual subtleties.

REFERENCES:

- https://www.ijcaonline.org/archives/volume182/number4/dhiman-2018-ijca-917513.pdf
- https://medium.com/@ekapradina02/sentiment-analysis-using-naive-bayes-and-k-nearest-neighbor-apps-reviews-23a6b290b04