CS 6375

ASSIGNMENT ______3_______

Names of students in your group:

Ching-Yi Chang - CXC190002

Xiaokai Rong - XXR230000

Number of free late days used: ________0________

Note: You are allowed a **total** of 4 free late days for the **entire semester**. You can use at most 2 for each assignment. After that, there will be a penalty of 10% for each late day.

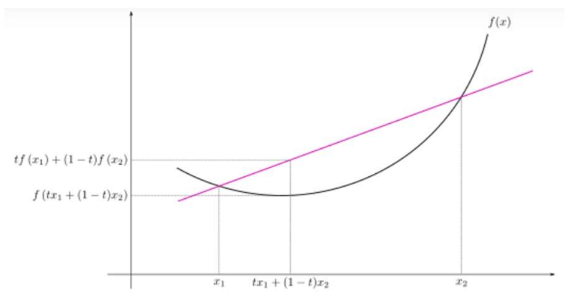
Please list clearly all the sources/references that you have used in this assignment.

PARTI. Theoretical Part (30 points)

1.

$$
\begin{aligned}
E_{\text{agg}} &= E\left[\left\{\frac{1}{M}\sum_{i=1}^{M} \epsilon_i(x)\right\}^2\right] \\
&= E\left[\frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M} \epsilon_i(x)\epsilon_j(x)\right] \\
&= \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M} E\big[\epsilon_i(x)\epsilon_j(x)\big]\big(E\big(\epsilon_i(x)\epsilon_j(x)\big) = 0 \text{ for all } i \neq j\big) \\
&= \frac{1}{M^2}\sum_{i=1}^{M} E[\epsilon_i(x)\epsilon_i(x)] \\
&= \frac{1}{M^2}\sum_{i=1}^{M} E[\epsilon_i(x)^2]\left(E_{\text{avg}} = \frac{1}{M}\sum_{i=1}^{M} E[\epsilon_i(x)^2]\right) \\
&= \frac{1}{M}\left(\frac{1}{M}\sum_{i=1}^{M} E[\epsilon_i(x)^2]\right) \\
&= \frac{1}{M}E_{\text{avg}}
\end{aligned}
$$

2.

	Jensen's inequality $tf(x_1)+(1-t)f(x_2) \geq f(tx_1+(1-t)x_2)$ $f\left(\sum_{i=1}^{M} \lambda_i x_i\right) \leq \sum_{i=1}^{M} \lambda_i f(x_i)$ $f(E[x]) \leq E[f(x)]$

In question 1, E_{avg} and E_{agg} are known,

$$E_{\text{avg}}(x) = \frac{1}{M}\sum_{i=1}^{M} E(\varepsilon_i(x)^2)$$

$$E_{agg}(x) = E\left[\left\{\frac{1}{M}\sum_{i=1}^{M} \epsilon_i(x)\right\}^2\right]$$

To prove that $E_{\text{agg}} \leq E_{\text{avg}}$, we can use Jensen's inequality and assumption the convex function. By Jensen's inequality, for any convex function $f(x)$:

$$f\left(\frac{1}{M}\sum_{i=1}^{M} x_i\right) \leq \frac{1}{M}\sum_{i=1}^{M} f(x_i)$$

Now, let's apply this to the definition of the error using the aggregated model $\left(E_{agg}\right)$:

$$f\left(\frac{1}{M^2}\sum_{i=1}^{M} [\epsilon_i(x)^2]\right) \leq \frac{1}{M^2}\sum_{i=1}^{M} f(\epsilon_i(x)^2) \leq \frac{1}{M}\sum_{i=1}^{M} \epsilon_i(x)^2$$

Now, the right-hand side of the inequality is exactly the definition of E_{avg} :

$$E_{agg}(x) \leq \frac{1}{M}\sum_{i=1}^{M} \epsilon_i(x)^2 = E_{avg}(x)$$

Thus, we have shown that $E_{\text{agg}} \leq E_{\text{avg}}$

3.

$$D_{T+1}(i) = D_1(i)\frac{e^{-\alpha_1 y_i h_1(x_i)}}{z_1} \cdot D_2(i)\frac{e^{-\alpha_2 y_i h_2(x_i)}}{z_2} \cdot D_t(i)\frac{e^{-\alpha_t y_i h_t(x_i)}}{z_t}$$
$$= \frac{1}{N}\frac{e^{-y_i \sum_{t=1}^{N} \alpha_t h_t(x_i)}}{\prod_t z_t} = \frac{1}{N}\frac{e^{-y_i f(x_i)}}{\prod_t z_t}$$

As $H(x) = \text{sign}\,(\sum_{t=1}^{T} \alpha_t h_t(x))$

$H(x_i) = \begin{cases} 0 & \text{if } y_i = H \\ 1 & \text{if } y_i \neq H \end{cases} \Rightarrow y_i$ is true label and belongs to $[-1,1]$

total training error

$$\underset{\text{at } t}{H(x_i)} = \frac{1}{N}\sum_{1}^{t} \left(\begin{cases} 0 & \text{if } y_i = H \\ 1 & \text{if } y_i \neq H \end{cases}\right) - (1)$$

this equation (1)

(1) $\leqslant \frac{1}{N}\sum_i e^{-y_i f(x_i)}$

Since $-y_i f(x_i) \leqslant 0, e^{-y_i f(x_i)} \geqslant 1$

(2) $= \frac{1}{N}\sum_i e^{-y_i f(x_i)} = \sum_i D_{t+1}(i) \cdot \prod_t z_t = \prod_t z_t$

So training error at final step is at most $\prod_t Z_t$

$$z_t = \sum_i D_t(i) \begin{cases} e^{-\alpha_t h_t y_i} \\ e^{\alpha_t h_t y_i} \end{cases}$$

we can simplify that to

$$\begin{aligned} &\text{if } h_t = y_i, h_t y_i = 1 \\ &\text{if } h_t \neq y_i, h_t y_i = -1 \end{aligned}$$

So

$$\begin{aligned} z_t &= \sum_i D_t(i) \cdot \begin{cases} e^{-\alpha t} & h_t = y_i \\ e^{\alpha t} & h_t \neq y_i \end{cases} \\ &= \sum_{h_t = y_i} e^{-\alpha_t} D_t(i) + \sum_{h_t \neq y_i} e^{\alpha_t} D_t(i) \\ &= e^{-\alpha_t} \sum_{h_t \neq y_i} D_t(i) + \sum_{h_t = y_i} D_t(i) + e^{\alpha_t} \\ &= e^{-\alpha_t}(1-\epsilon_t) + e^{\alpha_t}(\epsilon_t) \\ &= 2\sqrt{\epsilon_t(1-\epsilon_t)} \\ &= \sqrt{1-4\gamma_t^2} - (3) \end{aligned}$$

and (3) $= (1-4\gamma_t^2)^{\frac{1}{2}} \leq e^{-2\gamma_t^2} (1 + x \leq e^x$ for all Real $x)$

so we found our upper bound equal to

$$\exp(-2\gamma_t^2)$$

overall foundry $\underline{\underline{\text{is}}}$

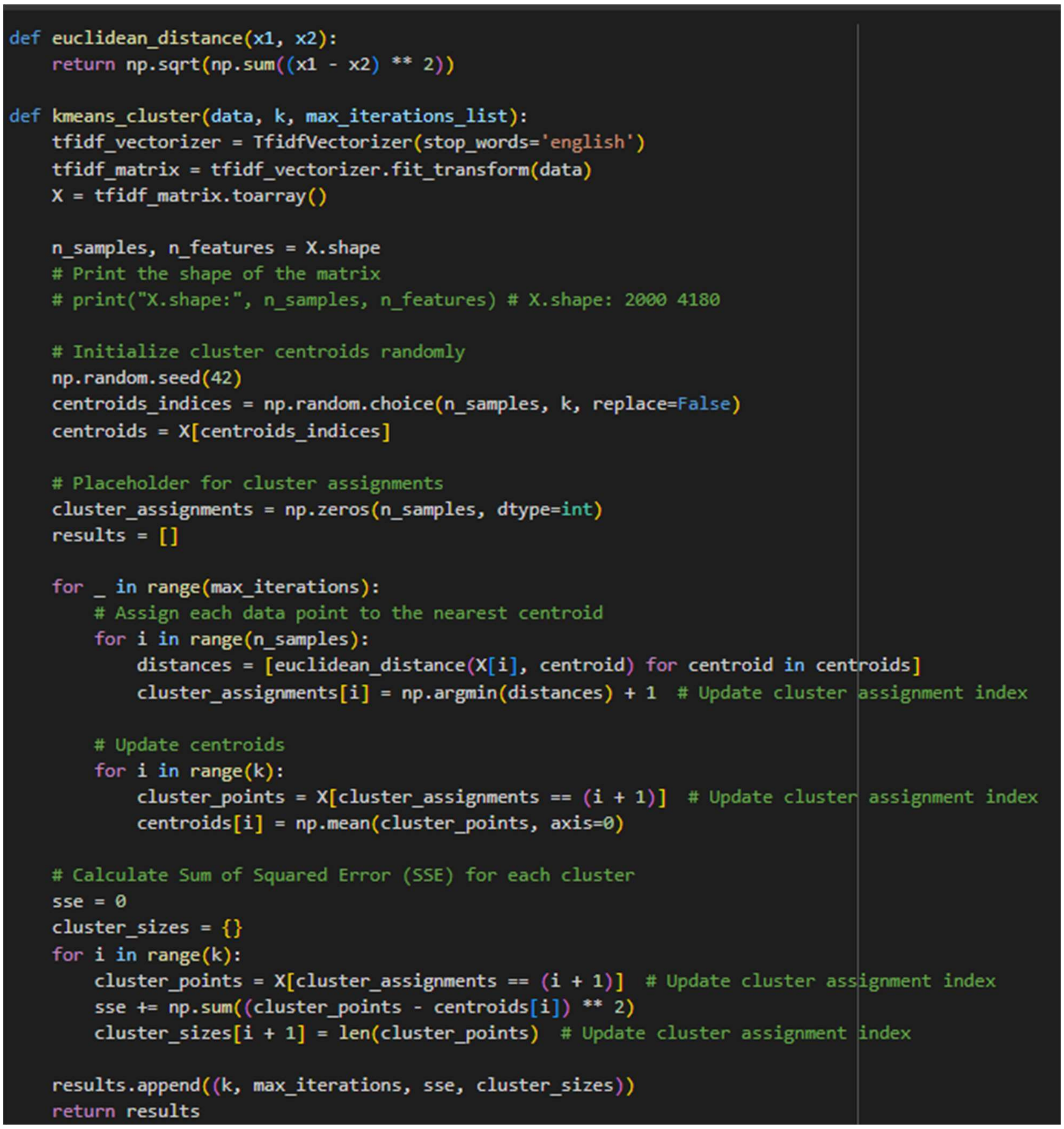
$$\exp\left(-2\sum_{t=1}^{t} (\gamma_t)^2\right)$$

So all classifier will not have error greater than

$$\exp\left(-2\sum_{t=1}^{t} (\gamma_t)^2\right)$$

PARTII. Programming Part (70 points): K-means unsupervised learning algorithm

A:



```python
def euclidean_distance(x1, x2):
    return np.sqrt(np.sum((x1 - x2) ** 2))

def kmeans_cluster(data, k, max_iterations_list):
    tfidf_vectorizer = TfidfVectorizer(stop_words='english')
    tfidf_matrix = tfidf_vectorizer.fit_transform(data)
    X = tfidf_matrix.toarray()

    n_samples, n_features = X.shape
    # Print the shape of the matrix
    # print("X.shape:", n_samples, n_features) # X.shape: 2000 4180

    # Initialize cluster centroids randomly
    np.random.seed(42)
    centroids_indices = np.random.choice(n_samples, k, replace=False)
    centroids = X[centroids_indices]

    # Placeholder for cluster assignments
    cluster_assignments = np.zeros(n_samples, dtype=int)
    results = []

    for _ in range(max_iterations):
        # Assign each data point to the nearest centroid
        for i in range(n_samples):
            distances = [euclidean_distance(X[i], centroid) for centroid in centroids]
            cluster_assignments[i] = np.argmin(distances) + 1  # Update cluster assignment index

        # Update centroids
        for i in range(k):
            cluster_points = X[cluster_assignments == (i + 1)]  # Update cluster assignment index
            centroids[i] = np.mean(cluster_points, axis=0)

    # Calculate Sum of Squared Error (SSE) for each cluster
    sse = 0
    cluster_sizes = {}
    for i in range(k):
        cluster_points = X[cluster_assignments == (i + 1)]  # Update cluster assignment index
        sse += np.sum((cluster_points - centroids[i]) ** 2)
        cluster_sizes[i + 1] = len(cluster_points)  # Update cluster assignment index

    results.append((k, max_iterations, sse, cluster_sizes))
    return results
```

(1) Table of results

A. Refer the website https://archive.ics.uci.edu/ml/datasets/Health+News+in+Twitter and the picked dataset “foxnewshealth.txt” and the following procedures.

B. Perform the following pre-processing steps:

- Remove the tweet id and timestamp

- Remove any URL
- Remove any hashtag symbols (#)
- Remove any word that starts with the symbol (@)
- Convert every word to lowercase

C. Perform K-means clustering on the resulting tweets using at least 5 different values of K and Note that the sum of squared error is defined as:

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} \mathrm{dist}^2(m_i, x)$$

The SSE is calculated as follows:

1. For each data point, compute the squared distance from the data point to its assigned cluster centroid (the cluster mean).
2. Sum up all these squared distances for all data points.

Value of K	SSE Sum of Squared Errors	Size of each cluster
1	1988.39	{1: 2000}
2	1982.58	{1: 1880, 2: 120}
3	1971.55	{1: 1683, 2: 118, 3: 199}
4	1962.20	{1: 155, 2: 74, 3: 1455, 4: 316}
5	1956.55	{1: 154, 2: 74, 3: 1402, 4: 308, 5: 62}

In K-means, the algorithm aims to partition the data into K clusters by finding the centroids that minimize the SSE. At each iteration, the algorithm assigns data points to the nearest cluster centroid and then updates the centroids based on the new assignments. This process is repeated until convergence or until a specified number of iterations is reached.

As K increases from 1 to 5 (i.e., the number of clusters grows), the SSE generally becomes lower. This is because with more clusters, the algorithm can capture finer details and smaller patterns in the data. As a result, data points are likely to be closer to their respective centroids in smaller, more focused clusters, leading to a lower overall SSE.

(2) Please refer.

- “kmeans.ipynb” or refer to the Colab notebook can be accessed at https://colab.research.google.com/drive/1d5eb3PrrCj8Dn224Am7cQhYetjVsVMOJ?usp=sharing
- “README” file to run the code.