שאלה 1:

הגדרנו זוג פונקציות עזר, אחת שמטרתה לייצר דגימות כפי שהוגדר בשם generate_points והשניה אשר מגדירה מסווג K-NN.

הפונקציות:

File - C:\Users\shado\PycharmProjects\Machine Learning 1\HW1\GenerateSamples.py

```python
import numpy as np
import matplotlib.pyplot as plt


def generate_samples(num, isplot):
    """
    Generating samples. A random number in [0,1] chooses which mu-parameter we'll use for the normal distribution.
    :param num: number of samples we want to generate.
    :param isplot: if isplot == 1, it will plot all the samples generated as dots.
    else it will not plot it.
    :return: a list of size 2, when index 0 contains all the samples, and index 1 contains all their labels.
    """
    mu1 = [-1, 1]
    mu2 = [-2.5, 2.5]
    mu3 = [-4.5, 4.5]
    sigma = [[1, 0], [0, 1]]
    test_group = []
    test_group_labels = []
    for i in range(num):
        pickmu = np.random.rand()
        if pickmu < 1 / 3:
            mean = mu1
            label = 1
        elif pickmu > 2 / 3:
            mean = mu3
            label = 3
        else:
            mean = mu2
            label = 2

        test_group.append(np.random.multivariate_normal(mean, sigma))
        test_group_labels.append(label)
    if isplot == 1:
        fig, ax = plt.subplots()
```

```python
from sklearn.neighbors import KNeighborsClassifier


def knnexec(k, trainset, testset, is_train):
    """
    Create a K-NN classifier, using the train set, and predicts the test set or the train set.
    :param k: parameter K in the K-NN classifier
    :param trainset: A list of size two. index 0 contain all the train samples, and index 1 contains all their labels.
    :param testset: A list of size two. index 0 contain all the test samples, and index 1 contains all their labels.
    :param is_train: If is_train == 1, the function will try to predict the train samples labels.
    else, it will predict the test samples labels.
    :return: the error rate of prediction as mentioned in the assignment.
    """
    test_size = len(testset[0])
    train_size = len(trainset[0])
    classifier = KNeighborsClassifier(n_neighbors=k)
    classifier.fit(trainset[0], trainset[1])
    error = 0
    if is_train == 1:
        pred = classifier.predict(trainset[0])
        for i in range(train_size):
            if pred[i] != trainset[1][i]:
                error += 1
        return error / train_size
    else:
        pred = classifier.predict(testset[0])
        for i in range(test_size):
            if pred[i] != testset[1][i]:
                error += 1
        return error / test_size
```

לאחר מכן, הגדרנו פונקציות אשר עונות על כל השאלות בנפרד.

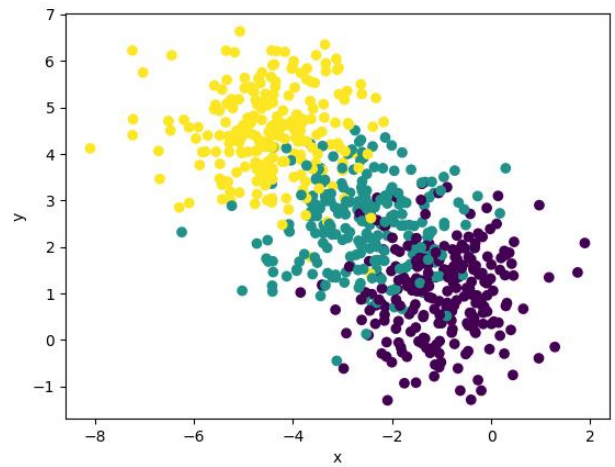
<u>סעיפים 1-4:</u>

הפונקציה:

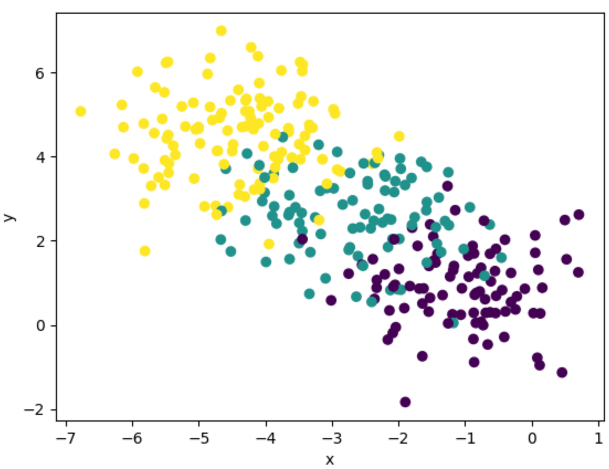
```python
from GenerateSamples import generate_samples
from knnexec import knnexec


def question1to4():
    train_set = generate_samples(700, 1)
    test_set = generate_samples(300, 1)
    train_error = knnexec(1, train_set, test_set, 1)
    test_error = knnexec(1, train_set, test_set, 0)
    print(train_error)
    print(test_error)
```

הגרפים שהתקבלו:

<u>Train Set</u>



<u>Test Set</u>



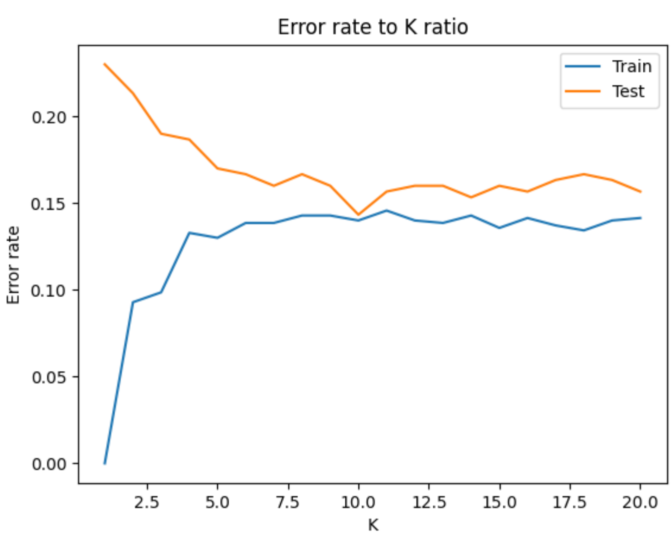
תשובה לסעיף 4: הError rate שהתקבל כשניסינו לצפות את הTrain Set היה 0, כפי שציפינו, משום שמסווג 1-NN יסווג נקודה מסט האימון לפי השכן הכי קרוב שלה – והיא תמיד תהייה הנקודה עצמה. לכן הוא תמיד יסווג נכון.
מנגד, קיבלנו שגיאה של 19.6% על הTest Set וזה בגלל שמסווג אשר בודק רק את השכן האחד הכי קרוב, לא מדוייק, היות והוא סובל מOverfitting.

סעיף 5:

הפונקציה שבה השתמשנו:

```python
from GenerateSamples import generate_samples
from knnexec import knnexec
import matplotlib.pyplot as plt


def question5():
    train_set = generate_samples(700, 0)
    test_set = generate_samples(300, 0)
    error_on_train_set = []
    error_on_test_set = []
    ax = [i for i in range(1, 21)]
    for i in range(1, 21):
        error_on_train_set.append(knnexec(i, train_set, test_set, 1))
        error_on_test_set.append(knnexec(i, train_set, test_set, 0))
    plt.plot(ax, error_on_train_set, label="Train")
    plt.plot(ax, error_on_test_set, label="Test")
    plt.xlabel("K")
    plt.ylabel("Error rate")
    plt.title("Error rate to K ratio")
    plt.legend()
    plt.show()
```

והגרף שהתקבל הינו:



תשובה לסעיף 5: ניתן לראות כי השגיאה קטנה ככל ש-K גדל. אנחנו גם רואים מהגרף, שזה לא תמיד מתקיים, כפי שניתן לראות על פי השיפועים החיוביים בגרף הכתום.
דוגמא נגדית נוספת תהייה:
נסווג נקודות על המישור הקרטזי, כך שהליבלים הם שחור או לבן.

$$train\ set = \{(1{,}0, Black), (0{,}1, black), (0{,}0, White)\}$$

$$test\ set = \{(0{,}0.1, White)\}$$

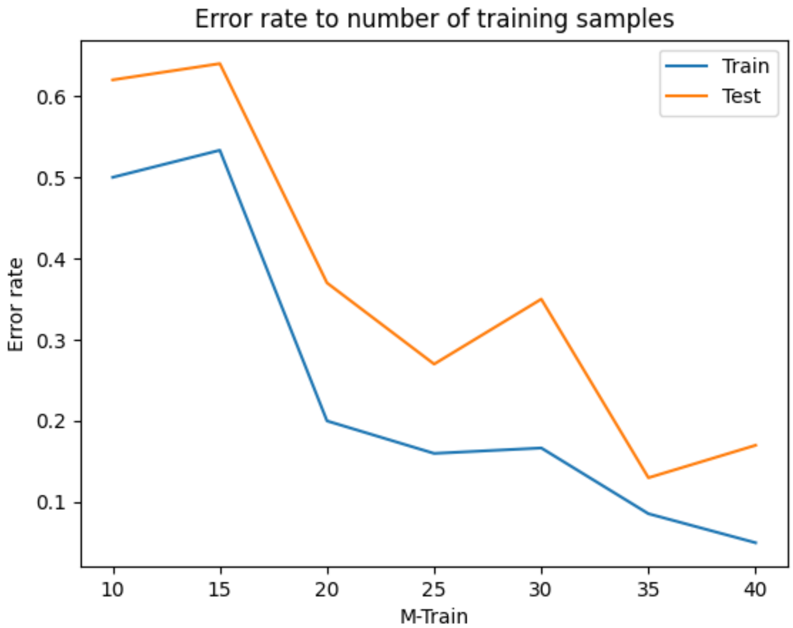
עבור K=1 המסווג יסווג נכון, היות והנקודה הכי קרובה לנק' הנבדקת היא הנקודה הלבנה בראשית הצירים.
כאשר K=3 המסווג יסווג כשחור, היות ואלו יהיו רוב השכנים של הנק' הנבדקת.
דבר כזה קורה בדרך כלל כאשר K גדול מאוד, ביחס לגודל של סט האימון.

סעיף 6:

הפונקציה:

```
from GenerateSamples import generate_samples
from knnexec import knnexec
import matplotlib.pyplot as plt


def question6():
    m_train = [i for i in range(10, 45, 5)]
    m_test = 100
    k = 10
    error_rate_train = []
    error_rate_test = []
    for m in m_train:
        train_set = generate_samples(m, 0)
        test_set = generate_samples(m_test, 0)
        error_rate_train.append(knnexec(k, train_set, test_set, 1))
        error_rate_test.append(knnexec(k, train_set, test_set, 0))
    plt.plot(m_train, error_rate_train, label="Train")
    plt.plot(m_train, error_rate_test, label="Test")
    plt.xlabel("M-Train")
    plt.ylabel("Error rate")
    plt.title("Error rate to number of training samples")
    plt.legend()
    plt.show()
```

הגרף המתקבל:



תשובה לסעיף 5: כאשר K=10 נוצרת אותה הבעיה שראינו בסעיף 5, כאשר אין לנו מספיק דוגמאות בסט האימון שלנו יחסית לגודל של K. אנחנו מאמינים שככל שסט האימון יכיל יותר דוגמאות, הטעות תרד.

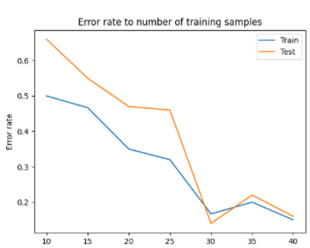
לאחר שהבטנו בגרף, אנו רואים שזה לא תמיד המקרה. באופן כללי, קיימת מגמת ירידה גם לסט האימון וגם לסט המבחן, אך לא לכל איטרציה בה כמות הדוגמאות שלנו גדלה.

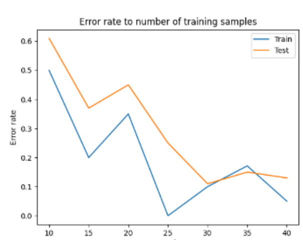
<u>סעיף 7:</u>

הפונקציה:

```
def question7():
    for i in range(6):
        question6()
```

הגרפים שהתקבלו:





תשובה לסעיף 7: הגרפים משתנים בצורה דרסטית בכל הרצה של הפונקציה מסעיף 6, אך באופן כללי השגיאה נהיית קטנה יותר בכולם ככל שסט האימון גדל.

השינויים הבולטים יכולים להיות מוסברים על ידי הרנדומליות איתה יצרנו את סט האימון והמבחן שלנו. כיוון שבאו מפונקציה הסתברותית, ככל שיש פחות נקודות ההבדלים בינהם מאוד גדולים. אם היינו מייצרים כמות גדולה מאוד של נקודות, הגרפים היו "מתמצעים" לפחות או יותר אותם הערכים לפי פונקציות ההסתברותיות בהן השתמשנו.

סעיף 8:

נציע מסווג K-NN אשר עבור כל נקודה שהוא אמור לסווג, הוא לוקח את קבוצת השכנים שלו, ומחלק אותה למחלקות, כך שכל מחלקה שייכת ללייבל אחר.
לאחר מכן הוא לוקח את המרחק האוקלידי הממוצע של הנקודות במחלקה מסויימת מהנקודה הנבדקת.
המחלקה שהמרחק הממוצע ממנה, אל הנקודה הנבדקת, הוא הקטן ביותר, תהייה הלייבל של הנק' הנבדקת.

שאלה 3

1. נתון סט אימון $S=\{(x_i,y_i)\}_{i=1}^{m}$ עם תיוגים אפשריים $y_i\in\{1,\dots,k\}$.

$$p_w(Y_i=k\mid x_i)=\frac{e^{w_k^T x_i}}{\sum_{j=1}^{k} e^{w_j^T x_i}}=p^k$$

במקרה הבינארי כאשר $y_i\in\{0,1\}$:

$$p_w(Y_i=1/x_i)=\frac{e^{w_k^T x_i}}{1+e^{w^T x_i}}=p$$

נגדיר פונקציה $\phi(x,y)$ שעבורה פונקציית ההסתברות תהיה כך: שעבור המקרה הבינארי $y_i\in\{0,1\}$ וההגדרה $y_i=1$ נקבל שהמקרה הבינארי מהווה מקרה פרטי של המקרה הכללי.

$$\phi(x,y)=\begin{cases} x & ;\ y=1 \\ 0 & ;\ \text{אחרת} \end{cases}$$

$$p(y=1/x,w)=\frac{e^{w^T\phi(x,1)}}{e^{w^T\phi(x,1)}+e^{w^T\phi(x,0)}}=\frac{e^{w^Tx}}{e^{w^Tx}+e^{w^T0}}=\frac{e^{w^Tx}}{e^{w^Tx}+1}=p_1$$

$$p(y=0/x,w)=\frac{e^{w^T\phi(x,0)}}{e^{w^T\phi(x,1)}+e^{w^T\phi(x,0)}}=\frac{e^{w^Tx}}{e^{w^Tx}+e^{w^T0}}=\frac{1}{e^{w^Tx}+1}=p_2$$

$$p_1+p_2=\frac{e^{w^Tx}}{e^{w^Tx}+1}+\frac{1}{e^{w^Tx}+1}=1$$

.2

$$\boxed{W^* = \operatorname{argmax}_w \left\{ \sum_{i=1}^{m} \log\left(p_w\left(Y_i = k / X_i\right)\right) \right\}}$$

יש למצוא את ערכי Moment matching constraint
הפרמטרים את הבעיה הנ"ל.

לכל דגימה מתקיים כי:
$$p(Y_i = k / x_i) = \frac{e^{w_k^T x_i}}{\sum_{\hat{k}} e^{w_{\hat{k}}^T x_i}}$$

פונקציית ה- log likelihood של מדגם i נראית כך:

$$\log\left(p(Y_i = k/x_i)\right) = \log\left(\frac{e^{w_k^T x_i}}{\sum_{\hat{k}} e^{w_{\hat{k}}^T x_i}}\right) = W_k^T X_i - \log\left(\sum_{\hat{k}} e^{w_{\hat{k}}^T x_i}\right)$$

כמו כן, כדי למצוא את w המביא למקסימום את הפונקציה הנ"ל,
עלינו למזער את $L_S(w)$:

$$L_S(w) = \frac{1}{m}\sum_{i=1}^{m} -\log\left(p\left(Y_i = y_i / x_i\right)\right) = \frac{1}{m}\sum_{i=1}^{m}\left(\log\left(\sum_{\hat{k}} e^{w_{\hat{k}}^T x_i}\right) - w_{y_i}^T x_i\right)$$

נגזור את הפונק' $L_S(w)$ לפי כל w_i ונשווה ל-0:

$$\forall k = 1, \dots, k \quad : \quad \frac{\partial L_S(w)}{\partial W_k} = \sum_{i=1}^{m} I[y_i = k] x_i - \sum_{i=1}^{m}\left(\frac{e^{w_k^T x_i}}{\sum_{\hat{k}} e^{w_{\hat{k}}^T x_i}}\right) x_i$$

$$\sum_{i=1}^{m} I[y_i = k] x_i = \sum_{i=1}^{m}\left(\frac{e^{w_k^T x_i}}{\sum_{\hat{k}} e^{w_{\hat{k}}^T x_i}}\right) x_i$$

$$\Downarrow$$

קיבלנו כי ה- moment matching constraint הוא:

$$\forall k = 1, \dots, k \qquad \sum_{i=1}^{m} I[y_i = k] x_i = \sum_{i=1}^{m}\left(\frac{e^{w_k^T x_i}}{\sum_{\hat{k}} e^{w_{\hat{k}}^T x_i}}\right) x_i$$

מההגדרה, מתקיים כי w^* מהווה w המקסם את $L_S(w)$, כלומר:

$$\forall k = 1, \dots, k \quad \sum_{i=1}^{m} I[y_i = k] x_i = \sum_{i=1}^{m}\left(\frac{e^{w_k^T x_i}}{\sum_{\hat{k}} e^{w_{\hat{k}}^T x_i}}\right) x_i = \sum_{i=1}^{m}\left(p_w\left(Y_i = k / x_i\right)\right) x_i$$

רגרסיה לוגיסטית

3. א. כל אחד מהמקדמים מייצג את ההשפעה באלמנט המתאים, אבל אנו משיגים את הסתברות הסיווג שמתאימה ל- X_i אבל אנו בוחנים.

ב.

$$h_s(X_i) = W_1 X_{i1} + W_2 X_{i2} + \dots + W_d X_{id} + b = W^T X_i + b$$

כאשר b מייצג את ה- bias.

כמו כן, אפשר להגדיר שלכל דגימה במדגם הכניסה ה-0 תוגדר כ- 1:

$$\tilde{X}_i = (1, X_i)$$

נקבל:

$$h_s(X_i) = W_0 X'_{i0} + W_1 X'_{i1} + \dots + W_d X'_{id} = W^T X'_i$$

ומכאן כי $b = W_0$.

$$\Downarrow$$

הדגימות המקוריות בגודל $X_i \in R^d$, $W \in R^{d+1}$ ולכן זה מתקיים

<u>עבור המקרה שלנו: $X' = (1, X) \Longleftarrow X \in R^2$ בגודל $W \in R^3$.</u>

ג. $\underline{X_1 = (X_{i0}, X_{i1}, X_{i2}) = (1, 1, 8)}$

$$P_w(Y=0|X_1) = \frac{e^{W_1^T X_1}}{e^{W_1^T X_1} + e^{W_2^T X_1} + e^{W_3^T X_1}} = \frac{e^{21.5}}{e^{21.5} + e^{-9.5} + e^{-12}} \cong 1$$

$$P_w(Y=1|X_1) = \frac{e^{W_2^T X_1}}{e^{W_1^T X_1} + e^{W_2^T X_1} + e^{W_3^T X_1}} = \frac{e^{-9.5}}{e^{21.5} + e^{-9.5} + e^{-12}} = 3.44 \times 10^{-14} \cong 0$$

$$P_w(Y=2|X_1) = \frac{e^{W_3^T X_1}}{e^{W_1^T X_1} + e^{W_2^T X_1} + e^{W_3^T X_1}} = \frac{e^{-12}}{e^{21.5} + e^{-9.5} + e^{-12}} = 2.826 \times 10^{-15} \cong 0$$

החישובים בהם השתמשנו:

$$\begin{bmatrix} w_1^T x_1 = (8,-2.5,2)\begin{pmatrix}1\\1\\8\end{pmatrix} = 8-2.5+16=21.5 \\ w_2^T x_1 = (2\ \ 0.5\ \ -1.5)\begin{pmatrix}1\\1\\8\end{pmatrix} = 2+0.5-12=-9.5 \\ w_3^T x_1 = (-10\ \ 2\ \ -0.5)\begin{pmatrix}1\\1\\8\end{pmatrix} = -10+2-4=-12 \end{bmatrix}$$

$\boxed{\text{מסקנה: המודל יסווג את הלייבל של } x_1 \text{ כ- } 0.}$

$\underline{x_2 = (x_{i0}, x_{i1}, x_{i2}) = (1,6,-2)}$

$$p_w(Y=0/x_2) = \frac{e^{w_1^T x_2}}{e^{w_1^T x}+e^{w_2^T x}+e^{w_3^T x}} = \frac{e^{-11}}{e^{-11}+e^{8}+e^{3}} = 5.56\times10^{-9} \cong 0$$

$$p_w(Y=1/x_2) = \frac{e^{w_2^T x_2}}{e^{w_1^T x_2}+e^{w_2^T x_2}+e^{w_3^T x_2}} = \frac{e^{8}}{e^{-11}+e^{8}+e^{3}} = 0.993$$

$$p_w(Y=2/x_2) = \frac{e^{w_3^T x_2}}{e^{w_1^T x_2}+e^{w_2^T x_2}+e^{w_3^T x_2}} = \frac{e^{3}}{e^{-11}+e^{8}+e^{3}} = 6.69\times10^{-3}$$

החישובים בהם השתמשנו:

$$\begin{bmatrix} w_1^T x_2 = (8\ \ -2.5\ \ 2)\begin{pmatrix}1\\6\\-2\end{pmatrix} = 8-15-4=-11 \\ w_2^T x_2 = (2\ \ 0.5\ \ -1.5)\begin{pmatrix}1\\6\\-2\end{pmatrix} = 2+3+3=8 \\ w_3^T x_2 = (-10\ \ 2\ \ -0.5)\begin{pmatrix}1\\6\\-2\end{pmatrix} = -10+12+1=3 \end{bmatrix}$$

$\boxed{\text{מסקנה: המודל יסווג את הלייבל של } x_2 \text{ כ- } 1.}$

$$\underline{X_3 = (X_{i0}, X_{i1}, X_{i2}) = (1\ \ 12\ \ 4)}$$

$$P_w(Y=0/X_3) = \frac{e^{w_1^T X_3}}{e^{w_1^T X_3}+e^{w_2^T X_3}+e^{w_3^T X_3}} = \frac{e^{-14}}{e^{-14}+e^{2}+e^{12}} = 5.1\times 10^{-12}$$

$$P_w(Y=1/X_3) = \frac{e^{w_2^T X}}{e^{w_1^T X_3}+e^{w_2^T X_3}+e^{w_3^T X_3}} = \frac{e^{2}}{e^{-14}+e^{2}+e^{12}} = 4.54\times 10^{-5}$$

$$P_w(Y=2/X_3) = \frac{e^{w_3^T X_3}}{e^{w_1^T X_3}+e^{w_2^T X_3}+e^{w_3^T X_3}} = \frac{e^{12}}{e^{-14}+e^{2}+e^{12}} = 0.99$$

החישובים בהם השתמשנו:

$$W_1^T X_3 = (8\ \ -2.5\ \ 2)\begin{pmatrix}1\\12\\4\end{pmatrix} = 8-30+8 = -14$$

$$W_2^T X_3 = (2\ \ 0.5\ \ -1.5)\begin{pmatrix}1\\12\\4\end{pmatrix} = 2+6-6 = 2$$

$$W_3^T X = (-10\ \ 2\ \ -0.5)\begin{pmatrix}1\\12\\4\end{pmatrix} = -10+24-2 = 12$$

<u>מסקנה:</u> המודל יסווג את הלייבל של X_3 כ-2.