סיכום ודיווח ממצאים פרויקט אחזור מידע

1 תוכן

2 הקדמה

2.1 ספריות שהשתמשנו

```
import pandas as pd
import numpy as np
import os
import nltk
nltk.download("punkt")
nltk.download('stopwords')
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from gensim.models import Word2Vec
from sklearn.metrics import confusion_matrix
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
import zipfile
from sklearn.metrics import accuracy_score
import seaborn as sns

import matplotlib.pyplot as plt
```

```
# load libraries
import pandas as pd
import numpy as np
import os
import nltk
nltk.download("punkt")
nltk.download('stopwords')
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.neighbors import NearestCentroid
from sklearn.model_selection import train_test_split
from gensim.models import Word2Vec
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import KFold
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import BernoulliNB, MultinomialNB
import zipfile
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.feature_selection import RFE
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import GridSearchCV
import warnings
warnings.filterwarnings('ignore')
```

```
In [1]:
import pandas as pd
import glob
import os
import re
import nltk
nltk.download('wordnet') # Download WordNet if you haven't already
from nltk.corpus import wordnet
import numpy as np
from nltk.stem import PorterStemmer
```

2.2 קצת על הקוד והדוח

הדוח כולל הסבר מקיף על הקוד בכל השלבים שלו וכמו כן הוא מכיל דיון כך שיכסה את כלל הדרישות של המשימה. יישמנו את הדברים שלמדנו בקורס. אם התיעוד לא מובן בקוד הוא יהיה מובן פה.

2.3 הפניות – REFERENCES

- https://scikit-learn.org/stable/index.html
- https://numpy.org/
- https://pandas.pydata.org/
- https://www.wikipedia.org/
- https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf
- https://mw12.haifa.ac.il/course/view.php?id=2867
- https://chat.openai.com/auth/login

3 *Language modeling*

התחלנו את המשימה על ידי קריאת כל המסמכים של הנושא שבחרנו, הקריאה נעשתה על ידי שימוש בספריה פשוטה של קריאת מסמכים בפייתון:

```
corpus = "" # reading our text files

for doc in names:
    with open('documents/' + doc, encoding='utf-8') as f:
        content = f.read()
        corpus = corpus + str(content)
```

מה שקיבלנו זה טקסט אחד גדול שהוא מיזוג של כל הטקסטים שיש לנו:

```
corpus
```

```
'A Comparison of Explanations Given by\nExplainable Artificial Intelligence Methods on\nAnalysing Electronic Health Recor
ds\n\nJamie Duell*, Xiuyi Fan*, Bruce Burnett*, Gert Aarts*, Shang-Ming Zhou‘\n* Swansea University, Swansea, United King
dom, {853435, xiuyi.fan, 989563, g.aarts } @swansea.ac.uk\nt University of Plymouth, Plymouth, United Kingdom, smzhou @ie
ee.org; shangming.zhou @ plymouth.ac.uk\n\nAbstract—eXplainable Artificial Intelligence (XAD aims to\nprovide intelligibl
e explanations to users. XAI algorithms such\nas SHAP, LIME and Scoped Rules compute feature importance\nfor machine lear
ning predictions. Although XAI has attracted\nmuch research attention, applying XAI techniques in healthcare\nto inform c
linical decision making is challenging. In this paper,\nwe provide a comparison of explanations given by XAI methods\nas
a tertiary extension in analysing complex Electronic Health\nRecords (EHRs). With a large-scale EHR dataset, we compare\n
features of EHRs in terms of their prediction importance es-\ntimated by XAI models. Our experimental results show that\n
the studied XAI methods circumstantially generate different top\nfeatures; their aberrations in shared feature importance
merit\nfurther exploration from domain-experts to evaluate human trust\ntowards XAI.\n\nIndex Terms—Explainable AI, Black
-box, Glass-box, Machine\nLearning, Electronic Health Records\n\nI. INTRODUCTION\n\nThough machines outperform human expe
rts in some appli-\ncations, one question remains: how can we assure that the AI\nsolutions are trustworthy? Commonalitie
s arise in algorithm\napplications, to where a model can exhibit different behaviours\ncompromising the trust factor, thi
s is where “black-box”\nmodels become an adversary to human trust, as understanding\nthe internal mechanisms of such mode
ls is difficult, if not\nimpossible [1].\n\nThe medical and health sciences have witnessed a growing\ninterest of using M
achine Learning (ML). However, in light\nof ensuring trust in human-AI collaboration, Tonekaboni et\nal. [2] have looked
at the question “what clinicians want?”\nThey identify that merely having highly accurate ML models\nis not sufficient fo
r clinicians; notably a single metric such\nas classification accuracy does not provide insight to how\nthe solution was
```

הטקסט היה מלא במילים שאין להם אף משמעות בשפה האנגלית בכדי לטפל בזה ראשית אנחנו החלטנו לשמור רק מחרוזות שהם מורכבות אך ורק מאותיות באנגלית וזה נעשה על ידי שימוש ב regular expression:

```
corpus = re.sub('[^a-zA-Z]+', ' ', corpus) # i only want a alphabetacal words that good enough for our purpose
len(corpus)
```

ההחלטה הזו דיי מספקת את המטרה הסופית ולא פוגעת קשות במטרה הסופית. כי רוב המילים שיש להם משמעות בטקסטים הינם מילים שמורכבים אך ורק מאותיות, וגם במנועי חיפוש בכללי דיי נדיר שמישהו ירצה לחפש מספרים או כל משהו אחר למטרת צורך המידע שלו.

אחרי זה אנחנו שמנו לב שיש מלא צירופי אותיות שאין להם שום משמעות בשפה הטבעית שלנו, אז החלטנו להסיר צירופים אלו על ידי שימוש בספריית NLTK:

```
our_tokens = [word for word in tokens if wordnet.synsets(word)]
```

יש לשים לב להבדל בין האורך של רשימת ה- TOKENS שלנו לפני ואחרי פעולה זו:

לפני:

```
tokens = corpus.split(" ")
len(tokens)
```

```
482446
```

אחרי:

```
len(our_tokens)
```

```
337226
```

<u>בניית המילון על ידי הסתכלות על הטקסט שלנו כמודל שפה של UNIGRAM:</u>

עשינו זאת על ידי בניית פונקציה שמקבצת את כל המילים לרשימה של מפתחות ואת כל ה TF שלהם לרשימת הערכים על ידי הפונקציה הבאה:

creating our simple dictionary by looking at the corpus in a unigram sight

```
def create_unig(tokens):
    token_dict = {}

# Iterate over the list of tokens
    for token in tokens:
    # Check if the token is already in the dictionary
        if token in token_dict:
        # If it is, increment its value by 1
            token_dict[token] += 1
        else:
        # If it's not, add it to the dictionary with a value of 1
            token_dict[token] = 1
    return token_dict
```

<u>סיכום המודל הבסיסי:</u>

Unique words	Word counts
17558	316399

	Token	TF	df	idf
9	on	3503	50	0.000000
0	A	2461	50	0.000000
4	by	2271	50	0.000000
5	Explainable	626	50	0.000000
7	Intelligence	549	48	0.017729
6	Artificial	537	49	0.008774
2	Explanations	217	34	0.167491
8	Methods	116	36	0.142668
3	Given	39	21	0.376751
1	Comparison	32	15	0.522879

3.1 ממצאים אחרי מחיקת ה- *stop words*

פה נתמקד בהשפעה של הסרת ה- stop words שניתנה לנו על ידי צוות הקורס. אז הממצאים שמצאנו הינם:

<u>סיכום המודל אחרי הסרת STOP WORDS:</u>

Unique words	Word count
17025	226484

	Token	TF	df	idf
0	A	2461	50	0.000000
104	model	2329	49	0.020203
54	In	1811	50	0.000000
147	data	1741	50	0.000000
34	explanations	1494	45	0.105361
282	explanation	1487	47	0.061875
262	based	1373	50	0.000000
69	models	1271	48	0.040822
42	learning	1236	49	0.020203
85	AI	1222	50	0.000000

ניתן לראות שאחרי שהורדנו את המילות תפל אנחנו איבדנו קצת מידע אבל גודל ה VOCABULARY לא באמת השתנה לא הייתה ירידה דרסטית, הירידה הייתה פשוט מ- 17558 ל- 17025 אבל יש לשים לב שרשימת הפרסומים תרד משמעותית

3.2 *Case folding*

קיפול רישיות הוא תהליך המרת כל האותיות במחרוזת לאותיות נפוצות (בדרך כלל אותיות קטנות) על מנת להפוך את המחרוזת לקלה יותר להשוואה ולתפעול. בעיבוד טקסט ועיבוד שפה טבעית, קיפול רישיות הוא שלב נפוץ בנורמליזציה של טקסט, שהוא תהליך המרת נתוני טקסט לפורמט סטנדרטי שניתן להשתמש בו לניתוח נוסף.

אנחנו נעשה פעולה זו על ידי שורה אחת בפייתון:

```
case_folded_tokens = [token.lower() for token in our_tokens]
```

Unique words	Word counts
12791	316399

ניתן לראות את כמות המונחים שנשארה לנו אחרי ביצוע תהליך זה וזה נובע מכך שהרבה מילים יאוחדו למילה אחת.

	Token	TF	df	idf
0	a	10140	50	0.000000
50	in	9282	50	0.000000
56	is	4941	50	0.000000
9	on	3711	50	0.000000
34	as	3154	50	0.000000
106	are	2873	50	0.000000
112	model	2619	49	0.020203
208	be	2409	50	0.000000
4	by	2360	50	0.000000
118	an	2261	50	0.000000

3.3 *Stemming* - גזירה

במורפולוגיה הלשונית ובשליפת מידע, גזירה היא תהליך של הפחתת מילים מוטות (או לפעמים נגזרות) לצורת בסיס או שורש של המילה שלהן - בדרך כלל צורת מילה כתובה. הבסיס אינו חייב להיות זהה לשורש המורפולוגי של המילה; בדרך כלל מספיק שמילים קשורות ממפות לאותו בסיס, גם אם בסיס זה אינו שורש תקף בפני עצמו. אלגוריתמים ליישום STEMMING נחקרו במדעי המחשב מאז שנות ה-60. מנועי חיפוש רבים מתייחסים למילים עם אותו בסיס כמילים נרדפות כסוג של הרחבת שאילתה, תהליך הנקרא קונפלציה.

אנחנו השתמשנו ב אלגוריתם של פורטר:

```
stemmer = PorterStemmer()
```

```
stemmed_tok = [stemmer.stem(word) for word in our_tokens]
```

Unique words	Word counts
7619	316399

פה רואים ירידה משמעותית ב- unique words וזה קורה בגלל שהרבה מילים שהם בצורות שונות אבל יש להם את אותה תחילת למשל: $dog, doggy, dogs, dogging$.

	Token	TF	df	idf
0	a	10140	50	0.000000
49	in	9283	50	0.000000
55	is	4941	50	0.000000
66	model	4217	49	0.020203
9	on	3711	50	0.000000
2	explan	3531	48	0.040822
33	as	3154	50	0.000000
102	are	2874	50	0.000000
194	be	2617	50	0.000000
599	system	2553	49	0.020203

ניתן לראות שכל השיטות האלו גרמו להשפעה על כמות המילים ועל גודל המילון שיהיה לנו וכמו כן אם נרצה לעשות אינדקס למשל אנחנו נחסוך מידע שאולי הוא נחוץ ואולי לא, בעולם האחזור מידע אנחנו תמיד מנסים יוריסטיקות שונות כדי להחליט איך לעשות את התהליך הזה. כל בחירה הינה תלויה במשימה שלה. פה ניתן לראות למעלה איך כל שיטה השפיעה וכל אחת יש לה את הסיכונים שלה, בעולם שלנו סיכון זה משהו שיכול להיות קריטי ואולי לא. אצלנו לא היו ממצאים חריגים ביחס למשימה אפשר לעשות את כלל השיטות ועדיין לקבל מודלים טובים.

המילה הכי שכיחה שלא הושפעה מאף שיטה:

In [73]:
```
stem_10_l = list(stem_10["Token"])
caseFol_10_l = list(caseFol_10["Token"])
stop_words_l = list(stopW_10["Token"])
token_10_l = list(token_10["Token"])

print(stem_10_l)
print(caseFol_10_l)
print(stop_words_l)
print(token_10_l)

i1 = set(stem_10_l).intersection(set(caseFol_10_l))
i2 = i1.intersection(set(stop_words_l))
i2
```

```
['a', 'in', 'is', 'model', 'on', 'explan', 'as', 'are', 'be', 'system']
['a', 'in', 'is', 'on', 'as', 'are', 'model', 'be', 'by', 'an']
['A', 'model', 'In', 'data', 'explanations', 'explanation', 'based', 'models', 'learning', 'AI']
['on', 'A', 'by', 'Explainable', 'Intelligence', 'Artificial', 'Explanations', 'Methods', 'Given', 'Comparison']
```

Out[73]:
```
{'model'}
```

4 Text Classification

המטרה שלנו בחלק הזה הוא לבנות 4 מסווגים שונים כדי לבחון ביצועים על נתוני אימון שאנחנו בונים בעצמנו בהתייחס לדרישות.

נתחיל בהקדמה קצרה של מה זה סיווג, סיווג טקסט הוא המשימה של סיווג או תיוג אוטומטי של מסמכי טקסט על סמך תוכנם. זה כרוך באימון אלגוריתם למידת מכונה על קבוצה של מסמכי טקסט מסומנים, כאשר לכל מסמך מוקצית קטגוריה או תוויות אחת או יותר. לאחר מכן האלגוריתם משתמש בנתוני האימון הללו כדי ללמוד דפוסים ויחסים בטקסט, ויכול לסווג מסמכי טקסט חדשים שלא נראים לקטגוריות המתאימות על סמך הדפוסים הנלמדים הללו.

לסיווג טקסט יש מגוון רחב של יישומים, כולל סינון דואר זבל, ניתוח סנטימנטים, מודלים של נושאים והמלצת תוכן, בין היתר. זה יכול לשמש בתחומים שונים, כגון שיווק, בריאות, חינוך וכספים, אם להזכיר כמה. מטרת סיווג הטקסט היא להפוך את תהליך הסיווג או התיוג של מסמכי טקסט לאוטומטיים, דבר שעלול לצרוך זמן רב ונוטה לשגיאות כאשר נעשה באופן ידני, ולאפשר ניתוח מהיר ומדויק של כמויות גדולות של נתוני טקסט.

המסווגים שהשתמשנו בהם:

1. Gaussian Naive Bayes
2. Bernoulli Naive Bayes
3. Rocchio
4. KNN

קודם כל התהליך שלנו התחיל על בחירת המסמכים משאר התקיות שהיו בדרייב:

```
my_docs_text = 'documents/'
other_docs_text = 'otherDocs/'
```

ואז הגרנו פונקציה שתקרא את המסמכים שיש בכל אחד מנתיבים הללו:

```
def read_text(dir):

    # variable for the text
    corpus = ""

    # names of all the documents in the directories
    names = os.listdir(dir)

    # loop through names
    for name in names:

        # create a file path with the curent name
        file_path = dir + os.sep + name
        if name.endswith(".txt"):

            # open the file and read text
            with open(file_path, "rb") as f:
                content = f.read()
                content = str(content)

                # add the text to corpus
                corpus = corpus + content

    # remove unwanted characters from the text
    corpus = corpus.replace("\\n", "")
    corpus = corpus.replace("\\", "")

    # case fold - lower the corpus
    corpus = corpus.lower()

    # remove stop words
    # list of stop words
    corpus = [word for word in corpus.split() if word.lower() not in stop_words]
    corpus = " ".join(corpus)


    # return the corpus
    return corpus
```

התהליכים דיי דומים כמו שהסברנו קודם בסעיף הקודם.

אחרי שקיבלנו את כל המסמכים תייגנו את המסמכים באופן בינארי באופן הבא:

$C : \{our\ documents, other\ student\ docments\} \rightarrow \{0,1\}$ בהתאמה.

מה שקיבלנו:

	Tokens	id
0	the hierarchically structured argumentation he...	0
1	uk/for-organisations/guide-to-data-protection/...	1
2	rr5 argumentation explainabilityrrexplainabili...	0
3	rrthe dnn decisions explained de-rscriptions r...	0
4	breusch-godfrey durbin-wat-son tests serial c...	0
...	...	...
18881	end, develop explanatory process built ontop ...	1
18882	by modelling argumentation solutionsand fulfil...	0
18883	explainability concept borrowedx0cxai: concep...	1
18884	evidence design choices affect algorith mic n...	0
18885	hand ai researchers developers, whoaim improv...	1

18886 rows × 2 columns

אחרי זה כדי להשתמש במסווגים של למידת מכונה, אנחנו צריכים לייצג כל מסמך\תת מסמך להיות ווקטור שערכים שלו הינם משקלי ה- tf-idf של כל מונח. זה נעשה תוך שימוש בספריה שמכילה את האובייקט המתאים לזה: TfidfVectorizer:

```
]:
tfidf = TfidfVectorizer()
tfidf.fit_transform(df.Tokens).todense().shape

:[16]: (18886, 66210)
```

אחרי זה בנינו embedding כדי שכל מילה תהיה תכונה שממנה המסווג ילמד לתת את הסיווגים שלו:

```
# Pre-process the text data
texts = [row.split() for row in df['Tokens']]

# Train the word2vec model on the pre-processed text data
model = Word2Vec(texts, window=5, min_count=1)

# Convert each row into an embedding by averaging the embeddings of the individual words
df['features'] = df['Tokens'].apply(lambda x: sum([model.wv[word] for word in x.split()]) / len(x.split()))
```

מה שקיבלנו:

	Tokens	id	features
0	the hierarchically structured argumentation he...	0	[-0.06265471, 0.14159884, 0.101124614, 0.00372...
1	uk/for-organisations/guide-to-data-protection/...	1	[-0.00464857, 0.0009504807, 0.0065278616, -0.0...
2	rr5 argumentation explainabilityrrexplainabili...	0	[-0.03910797, 0.10462184, 0.08724193, -0.00268...
3	rrthe dnn decisions explained de-rscriptions r...	0	[-0.055446625, 0.1463909, 0.1266201, -0.001698...
4	breusch-godfrey durbin-wat-son tests serial c...	0	[-0.038484395, 0.08553052, 0.058774166, 0.0098...
...	...	...	...
18881	end, develop explanatory process built ontop ...	1	[-0.071110375, 0.1967806, 0.16787355, -0.00603...
18882	by modelling argumentation solutionsand fulfil...	0	[-0.04562812, 0.101992786, 0.07584733, 0.00373...
18883	explainability concept borrowedx0cxai: concep...	1	[-0.05075425, 0.1339301, 0.12159431, 0.0005065...
18884	evidence design choices affect algorith mic n...	0	[-0.09377047, 0.24795686, 0.15885271, 0.010760...
18885	hand ai researchers developers, whoaim improv...	1	[-0.072184965, 0.25597805, 0.24685268, -0.0280...

אחרי זה אנחנו בנינו DATAFRAME חדש שמכיל את התכונות של הווקטור ייצג את המסמך:

```
features = []

for i in range(len(df)):
    features.append(list(df['features'][i]))
x = pd.DataFrame(features)
```

```
x
```

	0	1	2	3	4	5	6	7	8	9	...	90	91	92	
0	-0.062655	0.141599	0.101125	0.003729	-0.011048	-0.247232	0.063085	0.270171	-0.079529	-0.114763	...	0.165185	0.046771	0.037946	0.0517
1	-0.004649	0.000950	0.006528	-0.004460	-0.004877	0.006248	-0.008968	0.004221	0.008088	-0.009474	...	-0.009334	-0.007958	-0.000429	-0.0045
2	-0.039108	0.104622	0.087242	-0.002683	-0.008567	-0.194073	0.052504	0.205487	-0.052563	-0.102214	...	0.126187	0.026635	0.020649	0.0533
3	-0.055447	0.146391	0.126620	-0.001698	-0.008015	-0.288147	0.076697	0.297902	-0.080539	-0.148287	...	0.183947	0.037860	0.033685	0.0764
4	-0.038484	0.085531	0.058774	0.009809	-0.006212	-0.149358	0.030706	0.164239	-0.058903	-0.059754	...	0.105212	0.037676	0.031704	0.0172
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
18881	-0.071110	0.196781	0.167874	-0.006039	-0.019098	-0.384216	0.105792	0.396662	-0.107301	-0.197102	...	0.241944	0.046731	0.048010	0.1068
18882	-0.045628	0.101993	0.075847	0.003736	-0.011529	-0.186198	0.047936	0.203412	-0.061958	-0.088050	...	0.124889	0.032339	0.029753	0.0376
18883	-0.050754	0.133930	0.121594	0.000507	-0.019010	-0.280719	0.072892	0.295656	-0.078467	-0.151070	...	0.178505	0.039155	0.037383	0.0750
18884	-0.093770	0.247957	0.158853	0.010760	-0.021186	-0.387587	0.111079	0.400904	-0.113900	-0.184801	...	0.252820	0.075075	0.039089	0.1024
18885	-0.072185	0.255978	0.246853	-0.028028	-0.027372	-0.531435	0.156515	0.529483	-0.116660	-0.303792	...	0.321814	0.032256	0.043073	0.1775

18886 rows × 100 columns

החלטנו גם ליישם פונקציה של בחירת תכונות כדי לבדוק את התוצאות.

4.1 *Feature selection*

```
X_select = pd.DataFrame(x.copy())
y_select = df['id'].values

model = LogisticRegression()

rfe = RFE(model, n_features_to_select=15)

# Fit the RFE object to the data
rfe.fit(X_select, y_select)

best_features = X_select.columns[rfe.support_].tolist()

print(best_features)

X_select = X_select[best_features]
X_select
```

```
[13, 14, 22, 25, 35, 37, 40, 45, 59, 64, 68, 74, 77, 81, 96]
```

קוד זה מבצע בחירת תכונה באמצעות שיטת בחירת תכונות רקורסיבית (RFE) עם מודל רגרסיה לוגיסטית. להלן הסבר על כל שורה:

- X_select = pd.DataFrame(x.copy()): יוצר DataFrame X_select של פנדה חדש עם אותם נתונים כמו x, שהוא ככל הנראה מטריצה של תכונות לסיווג טקסט.

- y_select = df['id'].values: יוצר מערך numpy חדש y_select עם התגיות (LABELS) עבור משימת סיווג הטקסט.

- model = LogisticRegression(): יוצר מודל חדש של רגרסיה לוגיסטית.

- rfe = RFE(model, n_features_to_select=15): יוצר אובייקט RFE חדש עם מודל הרגרסיה הלוגיסטית בתור האומד, ומציין שעליו לבחור את 15 התכונות המובילות.

- rfe.fit(X_select, y_select): מתאים את אובייקט ה-RFE לנתונים ב-X_select ו-y_select, ובוחר את 15 התכונות הטובות ביותר.

- best_features = X_select.columns[rfe.support_].tolist(): מאחזר את שמות 15 התכונות המובילות מאובייקט RFE, וממיר אותם לרשימה.

- X_select = X_select[best_features]: יוצר DataFrame X_select חדש הכולל רק את 15 התכונות המובילות.

המטרה של קוד זה היא לבצע בחירת תכונות במטריצת הקלט x עבור משימת סיווג טקסט. שיטת RFE משמשת לזיהוי 15 התכונות המובילות החשובות ביותר עבור משימת הסיווג, ונוצר DataFrame X_select חדש הכולל רק את התכונות הללו. זה יכול לעזור לשפר את הדיוק והיעילות של מודל סיווג הטקסט, על ידי הפחתת הממדיות של תכונות הקלט והתמקדות באלה האינפורמטיבית ביותר.

4.2 חלקות הנתונים לנתוני אימון ובדיקה לפי הדרישות

```
X_train, X_test, y_train, y_test = train_test_split(x, df['id'].values, test_size=0.1, random_state=42)
X_train_select, X_test_select, y_train_select, y_test_select = train_test_split(X_select, df['id'].values, test_size=0.1,
```

קוד זה משמש לפיצול הנתונים לקבוצות הדרכה ובדיקות לסיווג טקסט. להלן הסבר על כל שורה:

X_train, X_test, y_train, y_test = train_test_split(x, df['id'].values, test_size=0.1, random_state=42): מפצל את מטריצת הקלט x ומתווית df['id'].values לקבוצות אימון ובדיקה . הפרמטר test_size מציין את הפרופורציה של הנתונים שישמשו לבדיקה (במקרה זה, 10%), והפרמטר random_state מגדיר את ה-SEEDS האקראי לשחזור. המשתנים המתקבלים הם X_train (תכונות אימון), X_test (תכונות בדיקה), y_train (תוויות אימון) ו-y_test (תוויות בדיקה).

מטרת קוד זה היא לפצל את הנתונים למערכות הדרכה ובדיקות להדרכה והערכה של מודלים, הן עבור התכונות המקוריות והן עבור התכונות הנבחרות. לאחר מכן, המשתנים המתקבלים יכולים לשמש כקלט לקוד האימון של המודל, כגון רגרסיה לוגיסטית או k-NN, כדי לאמן ולהעריך את ביצועי המודל. על ידי שימוש הן בתכונות המקוריות והן בתכונות שנבחרו, הקוד יכול להשוות את הביצועים של המודלים עם ערכות תכונות שונות, ולבחור את זו שמשיגה את הציון הטוב ביותר.

4.3 הסיווג

4.3.1 *Gaussian Naïve Bayes*

Gaussian Naive Bayes הוא אלגוריתם סיווג שמניח שהתכונות בנתונים מתפלגות נורמאלית , והוא מחשב את הסבירות של כל תכונה להיות שייכת למחלקה מסוימת. לאחר מכן הוא משלב את ההסתברויות כדי לחזות את המחלקה עם ההסתברות הגבוהה ביותר. האלגוריתם פשוט, מהיר ועובד היטב עם מערכי נתונים קטנים עד בינוניים, מה שהופך אותו לבחירה פופולרית למשימות סיווג טקסט. עם זאת, ייתכן שהוא לא מתפקד טוב עם מערכי נתונים גדולים ומורכבים מאוד או כאשר הנחת האי תלות של התכונות אינה מתקיימת.

<u>יישום:</u>

קוד זה מבצע hyperparameter tuning עבור מסווג גאוס נאיבי באמצעות GridSearchCV:

```
estimator_NB = GaussianNB()

param_grid_NB = {
    'var_smoothing': [1e-9, 1e-8, 1e-7]
}

kfold = KFold(n_splits=10, shuffle=True, random_state=42)

grid_search = GridSearchCV(estimator_NB, param_grid_NB, cv=kfold, n_jobs=-1)

grid_search.fit(X_train, y_train)

print('Best hyperparameters: {}'.format(grid_search.best_params_))
print('Best score: {:.2f}'.format(grid_search.best_score_))

y_pred = grid_search.predict(X_test)

print(classification_report(y_test, y_pred))


cm = confusion_matrix(y_test, y_pred)

# Plot the confusion matrix as a heatmap
sns.heatmap(cm, annot=True, cmap='Blues')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

misclassified_tokens = np.where(y_test != y_pred)[0:10]

filtered_df = df.loc[misclassified_tokens][['Tokens', 'id']]

print("documents classifed mine but they are not mine:\n")
print(filtered_df[filtered_df['id']== 1]['Tokens'].head(20))

print()

print("documents classifed not mine but they are  mine:\n")
print(filtered_df[filtered_df['id']== 0]['Tokens'].head(20))
```

- estimator_NB = GaussianNB(): יוצר אובייקט מסווג גאוס נאיבי חדש בתור האומד עבור ה-grid search .

- param_grid_NB = {'var_smoothing': [1e-9, 1e-8, 1e-7]}: מגדיר מילון של היפרפרמטרים לחיפוש. במקרה זה, אנו עושים tunning רק ל- פרמטר var_smoothing, שהוא פרמטר החלקה שקובע את עוצמת ה-regularization על הערכות השונות של התכונות .

- kfold = KFold(n_splits=10, shuffle=True, random_state=42): יוצר אובייקט חדש של של K-fold עם 10 פיצולים, ערבוב הנתונים לפני הפיצול והגדרת ה-seed האקראי לשחזור.

- grid_search = GridSearchCV(estimator_NB, param_grid_NB, cv=kfold, n_jobs=-1): יוצר אובייקט GridSearchCV חדש עם אומדן ה-Gaussian Naive Bayes, מילון ההיפרפרמטרים, אובייקט K-fold ומספר העבודות להפעלה במקביל (n_jobs=-1 פירושו להשתמש בכל ליבות המעבד הזמינות).

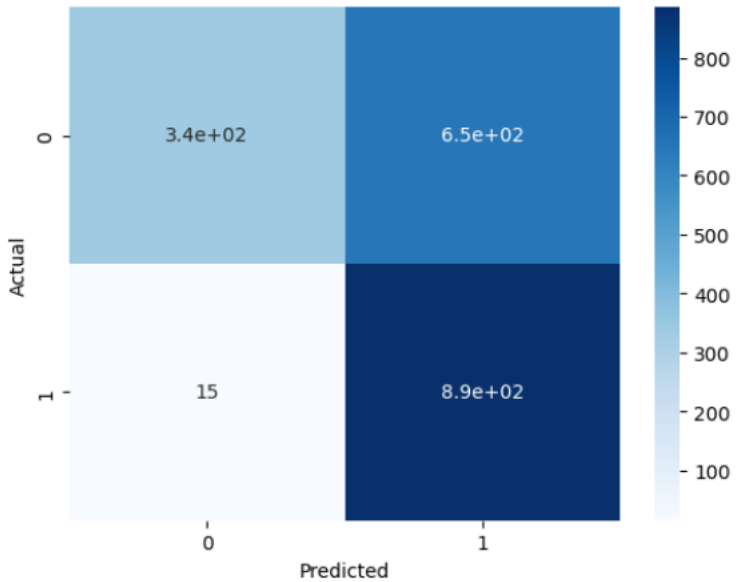
- grid_search.fit(X_train, y_train): מתאים את אובייקט GridSearchCV לנתוני האימון X_train ו-y_train, על ידי חיפוש במרחב ההיפרפרמטר שהוגדר ב-param_grid_NB והערכת ביצועי המודל עם אימות צולב.

- y_pred = grid_search.predict(X_test): משתמש במודל הטוב ביותר שנמצא כדי לחזות את התוויות עבור נתוני הבדיקה X_test.

מטרת קוד זה היא לבצע hyperparameter tuning עבור המסווג גאוס נאיבי, על ידי חיפוש על הפרמטר var_smoothing ובחירה באחד שמשיג את ציון ה-KFOLD הטוב ביותר. ניתן להשתמש בהיפרפרמטרים ובניקוד הטובים ביותר המתקבלים כדי לאמן מודל חדש על כל נתוני האימון, ולהעריך את הביצועים שלו על נתוני הבדיקה. ניתן להשתמש ב- y_pred גם לחישוב מדדי הערכה שונים.

<u>התוצאות שקיבלנו:</u>

```
Best hyperparameters: {'var_smoothing': 1e-09}
Best score: 0.63
              precision    recall  f1-score   support

           0       0.96      0.34      0.51       988
           1       0.58      0.98      0.73       901

    accuracy                           0.65      1889
   macro avg       0.77      0.66      0.62      1889
weighted avg       0.78      0.65      0.61      1889
```



ה- ACCURACY שווה ל 0.65 שכלומר המודל שלנו יצליח לסווג נכון בהסתברות של 0.65

עבור מחלקה 0, ה-PRECISION הוא 0.96, מה שאומר שכאשר המודל חוזה שמסמך נמצא במחלקה 0, הוא צודק ב-96% מהמקרים. עם זאת, ה-RECALL הוא רק 0.34, מה שאומר שהמודל מסוגל לזהות נכון רק 34% מהמסמכים האמיתיים של מחלקה 0. זה מצביע על כך שהמודל טוב מאוד בזיהוי ה-negative class כשהיא מופיעה, אבל הוא לא טוב מאוד בזיהוי כל ה-negative instances .

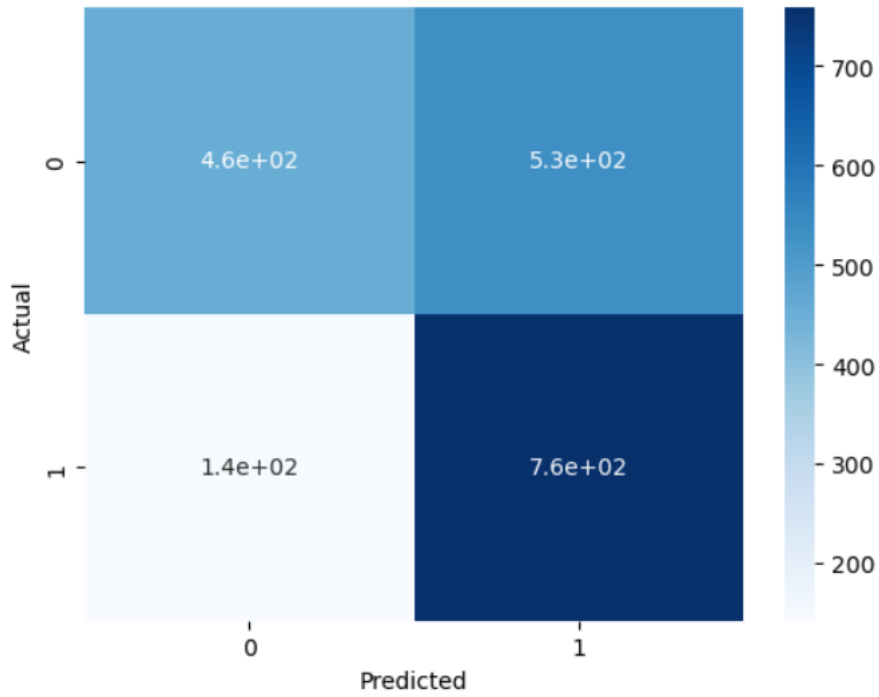
עבור מחלקה 1, ה-PRECISION הוא רק 0.58, מה שאומר שכאשר המודל חוזה שמסמך נמצא במחלקה 1, הוא צודק רק 58% מהמקרים. עם זאת, ה-RECALL הוא 0.98, מה שאומר שהמודל מסוגל לזהות נכון 98% מהמסמכים במחלקה 1 בפועל. זה מצביע על כך שהמודל אינו מדויק מאוד בזיהוי כל ה-positive instances , אך הוא טוב מאוד בזיהוי ה-positive instances כאשר הוא כן מזהה אותם.

```
documents classifed mine but they are not mine:

8      havingestablished explanatory relevance mathe...
11     level ofunderstanding sufficient effective xa...
17     hand, works adopt-ing deep learning solutions...
19     specifically, seek answer-how attacker deceiv...
29     role fibronectin-binding proteins bin vitro c...
38     compute metrics takencarefully tasks applicat...
39     lee: vibration signals analysis xal approach:...
46     7, july 2020pointhop: explainable machine lea...
48     number published articles (y axis) xai biblio...
50    xc2xae companies associated, hand, witha colle...
56     snapshot overview customer churninformation c...
57     propose (1) xe2x80x98black boxxe2x80x99 be-co...
58     tablex0cexplainable artificial intelligence, ...
61     2xe2x80x94based probabilitiesto churn colors:...
64     work providedcomprehensive xai categorization...
66    arun [82] proposed concerns role xai ininfluen...
75     computation kurtosis skewnesscan represented ...
77    markus langer, daniel oster, timo speith, holg...
81     level, heat-mapto explain classified animal i...
91     held multiple leadership positions campusorga...
Name: Tokens, dtype: object
```

```
documents classifed not mine but they are  mine:

2      rr5 argumentation explainabilityrrexplainabili...
10     16a decisions can, fact, fairer resultof remov...
25     x00 x00jx00ox00ux00rx00nx00ax00lx00 x00ox00fx0...
27     , xxe2x88x97 (0) = xxe2x88x97xcexb1 , period l...
28     x00 x00ux00nx00dx00ex00rx00 x00ax00lx00lx00 x0...
33      decision-makers happen belong overwhelmingly ...
36      respondents national news reportsrrwere biase...
43     x00 x00tx00hx00ex00 x00dx00ax00tx00ax00 x00cx0...
49      large regional national mainstream media inte...
52      specifically, argumentation play role opposin...
70     x00 x00rx00x00wx00hx00ex00nx00 x00ox00rx00gx00...
78     1 general purpose argumentation systemsrras ge...
80      16 cumulative distribution function (cdf) ent...
107     ethodswe trained neural network models predic...
110     usually, abd performed strict set rules, prot...
113    r porale processing accent variation spoken la...
123    rruser: preconditions running clinic here?rdoc...
127     rppd prediction rcomparison predictive models...
131     foremost, extend theoreticalknowledge drivers...
133     work alsoneeded extend variety types explanat...
```

<u>אחרי שימוש רק בתכונות שנבחרו על ידי ה-RFE:</u>

```
Best hyperparameters: {'var_smoothing': 1e-09}
Best score: 0.65
              precision    recall  f1-score   support

           0       0.76      0.46      0.57       988
           1       0.59      0.84      0.69       901

    accuracy                           0.64      1889
   macro avg       0.67      0.65      0.63      1889
weighted avg       0.68      0.64      0.63      1889
```




```
documents classifed mine but they are not mine:

11     level ofunderstanding sufficient effective xa...
17     hand, works adopt-ing deep learning solutions...
19     specifically, seek answer-how attacker deceiv...
38     compute metrics takencarefully tasks applicat...
46     7, july 2020pointhop: explainable machine lea...
48     number published articles (y axis) xai biblio...
50    xc2xae companies associated, hand, witha colle...
56     snapshot overview customer churninformation c...
57     propose (1) xe2x80x98black boxxe2x80x99 be-co...
58     tablex0cexplainable artificial intelligence, ...
64     work providedcomprehensive xai categorization...
66    arun [82] proposed concerns role xai ininfluen...
67    050contribution cc ceteris paribus09 pfn [mg/l...
75     computation kurtosis skewnesscan represented ...
77    markus langer, daniel oster, timo speith, holg...
81     level, heat-mapto explain classified animal i...
92     usesdata grad-cam effort extend theevaluation...
93     besides,this technique extends grad-cam and, ...
95     understanding productionparameters process qu...
96    information communication technology advanceme...
Name: Tokens, dtype: object
```

```
documents classifed not mine but they are  mine:

4       breusch-godfrey durbin-wat-son tests serial c...
9       formally defne deviation, xf0x9dx9bxa5pa, fac...
25     x00 x00jx00ox00ux00rx00nx00ax00lx00 x00ox00fx0...
27     , xxe2x88x97 (0) = xxe2x88x97xcexb1 , period l...
28     x00 x00ux00nx00dx00ex00rx00 x00ax00lx00lx00 x0...
33      decision-makers happen belong overwhelmingly ...
36      respondents national news reportsrrwere biase...
37      fusing atdd assurance caseswe propose combine...
49      large regional national mainstream media inte...
52      specifically, argumentation play role opposin...
70     x00 x00rx00x00wx00hx00ex00nx00 x00ox00rx00gx00...
73     x00 x00lx00ex00tx00 x00(x00x16x00xx00;x00x16x0...
80      16 cumulative distribution function (cdf) ent...
100    x00rx00x00[x006x009x00]x00 x00lx00ix00lx00yx00...
107     ethodswe trained neural network models predic...
110     usually, abd performed strict set rules, prot...
113    r porale processing accent variation spoken la...
123    rruser: preconditions running clinic here?rdoc...
127     rppd prediction rcomparison predictive models...
131     foremost, extend theoreticalknowledge drivers...
Name: Tokens, dtype: object
```

אין הבדל משמעותי בין הממצאים של שימוש במודל אחרי ולפני בחירת תכונות, הביצועים כמעט זהים לכן אותו הסבר כמו קודם.

4.4 *Bernoulli Naive Bayes*

Bernoulli Naive Bayes הוא גרסה של Naive Bayes המשמשת לבעיות סיווג בינארי עם תכונות בעלות ערך בינארי (0 או 1), כגון סיווג טקסט. זה מחשב את ההסתברות שכל תכונה שייכת למחלקה מסוימת ומשלבת אותם כדי לקבוע את ההסתברות שנקודת נתונים שייכת למחלקה. המודל הזה מניח שתכונות אינן תלויות זו בזו, מה שהופך אותו לפשוט ומהיר, אך מודל זה עשוי שלא לעבוד טוב עם מערכי נתונים גדולים או מורכבים או כאשר הנחת האי תלות מופרת.

```python
clf = BernoulliNB()

param_grid = {'alpha': [0.1, 1.0, 10.0], 'binarize': [0.0, 0.5, 1.0]}


grid_search = GridSearchCV(clf, param_grid, cv=kfold, n_jobs=-1)

grid_search.fit(X_train, y_train)

print('Best hyperparameters: {}'.format(grid_search.best_params_))
print('Best score: {:.2f}'.format(grid_search.best_score_))

y_pred = grid_search.predict(X_test)

print(classification_report(y_test, y_pred))


cm = confusion_matrix(y_test, y_pred)

# Plot the confusion matrix as a heatmap
sns.heatmap(cm, annot=True, cmap='Blues')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

misclassified_tokens = np.where(y_test != y_pred)[0:10]

filtered_df = df.loc[misclassified_tokens][['Tokens', 'id']]

print("documents classifed mine but they are not mine:\n")
print(filtered_df[filtered_df['id']== 1]['Tokens'].head(20))

print()

print("documents classifed not mine but they are  mine:\n")
print(filtered_df[filtered_df['id']== 0]['Tokens'].head(20))
```

למסווג BernoulliNB יש שני היפרפרמטרים שניתן לעשות להם TUNING:

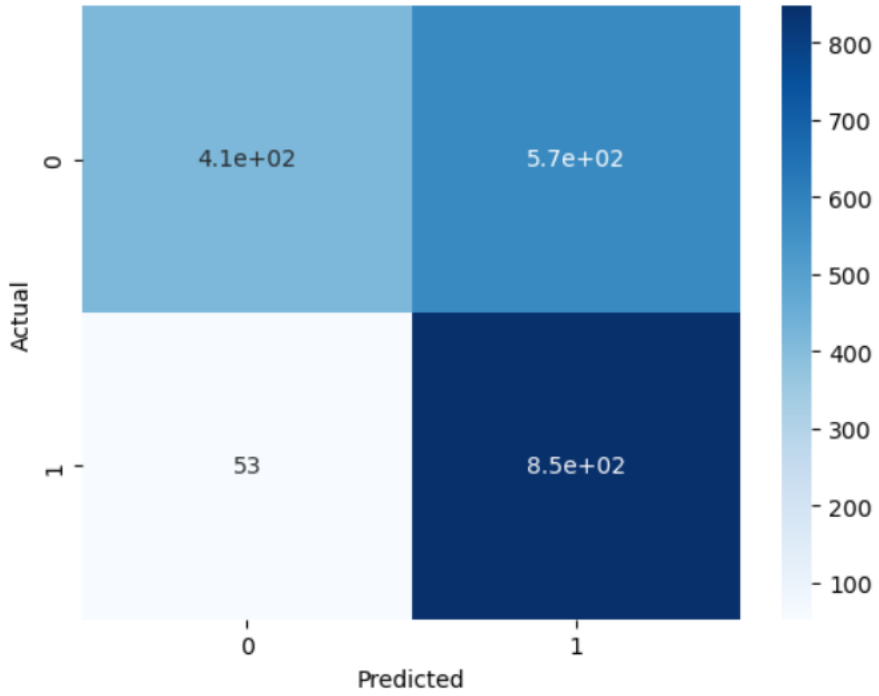
- alpha: זהו פרמטר החלקה, השולט ביכולת ההכללה של המודל. ערך קטן יותר של אלפא פירושו פחות החלקה והמודל רגיש יותר לנתוני האימון, בעוד שערך גדול יותר של אלפא מביא ליותר החלקה והמודל ניתן להכללה יותר.

- binarize: פרמטר זה משמש לבינאריזציה של התכונות, כלומר המרת ערכי התכונה ל-0 או 1 על סמך ערך סף. אם בינאריות מוגדרת ל-0, התכונות אינן בינאריות, בעוד שערך שאינו אפס ישמש כערך הסף. סף נמוך יותר יביא ליותר תכונות בינאריות.

מילון param_grid מכיל את הערכים של ההיפרפרמטרים הללו שינוסו על ידי GridSearchCV.
cv=kfold מגדיר את שיטת ה- CROSS VALIDATION לשימוש, ו-n_jobs=-1 אומר לפונקציה להשתמש בכל המעבדים הזמינים כדי להאיץ את החישוב. המשתנה y_pred מכיל את הערכים החזויים עבור נתוני הבדיקה.

<u>התוצאות שקיבלנו:</u>

```
Best hyperparameters: {'alpha': 0.1, 'binarize': 0.0}
Best score: 0.66
              precision    recall  f1-score   support

           0       0.89      0.42      0.57       988
           1       0.60      0.94      0.73       901

    accuracy                           0.67      1889
   macro avg       0.74      0.68      0.65      1889
weighted avg       0.75      0.67      0.65      1889
```

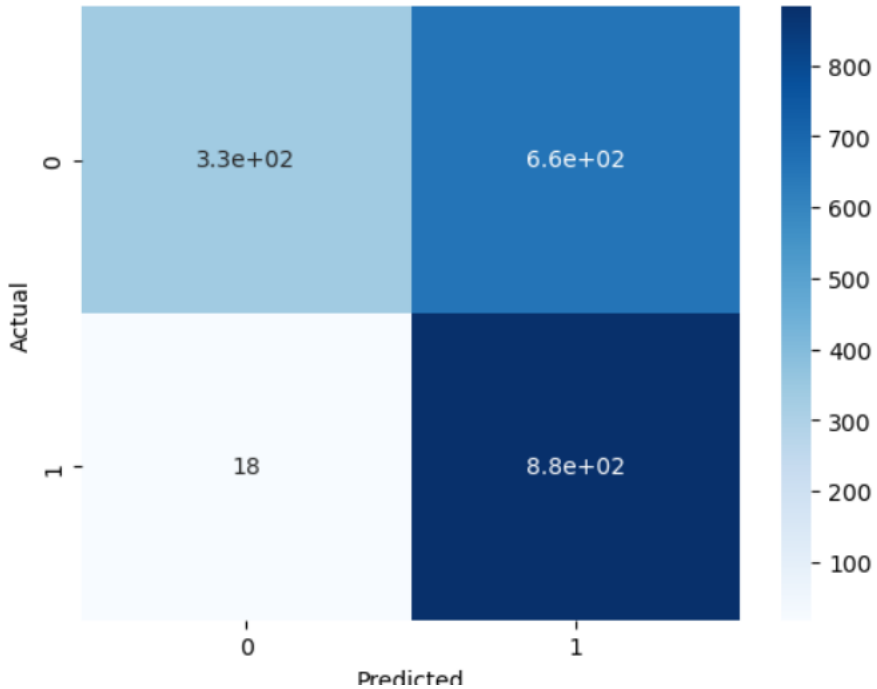


תוצאות דיי דומות למודל הקודם לכן ההסבר יהיה זהה.

```
documents classifed mine but they are not mine:

11     level ofunderstanding sufficient effective xa...
19     specifically, seek answer-how attacker deceiv...
29     role fibronectin-binding proteins bin vitro c...
38     compute metrics takencarefully tasks applicat...
39     lee: vibration signals analysis xal approach:...
48     number published articles (y axis) xai biblio...
50    xc2xae companies associated, hand, witha colle...
56     snapshot overview customer churninformation c...
57     propose (1) xe2x80x98black boxxe2x80x99 be-co...
58     tablex0cexplainable artificial intelligence, ...
64     work providedcomprehensive xai categorization...
66    arun [82] proposed concerns role xai ininfluen...
75     computation kurtosis skewnesscan represented ...
77    markus langer, daniel oster, timo speith, holg...
81     level, heat-mapto explain classified animal i...
92     usesdata grad-cam effort extend theevaluation...
93     besides,this technique extends grad-cam and, ...
95     understanding productionparameters process qu...
96    information communication technology advanceme...
99     addition renowned algorithms, aresome algorit...
Name: Tokens, dtype: object
```

ממצאים אחרי שימוש במודל על התכונות שנבחרו על ידי RFE:

```
Best hyperparameters: {'alpha': 0.1, 'binarize': 0.0}
Best score: 0.64
              precision    recall  f1-score   support

           0       0.95      0.34      0.50       988
           1       0.57      0.98      0.72       901

    accuracy                           0.64      1889
   macro avg       0.76      0.66      0.61      1889
weighted avg       0.77      0.64      0.61      1889
```



היה שיפור קטן בערכים של ה- RECALL וה- PRECISION עבור מחלקה 0 ומחלקה 1. וההסבר הכולל זהה למודל הקודם מבחינת הערכים של כל המדדים.

4.5 *Rocchio*

סיווג Rocchio הוא אלגוריתם השייך לאלגוריתמי ה- SUPERVISED המשמש למשימות סיווג טקסט, הפועל על ידי הקצאת מסמך למחלקה בהתבסס על הדמיון שלו למרכז המחלקה (CENTROID). המרכז מחושב כממוצע של וקטורי התכונה של כל מסמכי האימון השייכים לאותה מחלקה.

באלגוריתם זה, כל מחלקה מיוצגת על ידי וקטור מרכז במרחב התכונה. האלגוריתם מחשב תחילה את הסנטרואידים של כל המחלקות בהתבסס על נתוני האימון, ולאחר מכן משתמש במרכזים אלו כדי לסווג מסמכים חדשים.

כאשר מוצג מסמך חדש, האלגוריתם מחשב את וקטור התכונה שלו ומשווה אותו למרכזים של כל המחלקות. לאחר מכן, המסמך מסווג למחלקה עם המרכז הקרוב ביותר. מדד הדמיון המשמש להשוואת וקטור התכונה והסנטרואידים הוא בדרך כלל הדמיון הקוסינוס.

אלגוריתם Rocchio הוא פשוט, מהיר ויכול להיות יעיל עבור משימות סיווג טקסט, במיוחד עבור משימות שבהן מספר התכונות קטן יחסית. עם זאת, הוא יכול לסבול מבעיה של התפלגויות תכונות שהן חופפות, כאשר למחלקות שונות יש התפלגות תכונות דומות, מה שמקשה על הפרדתן על ידי הגבולות שהוא יוצר.

יישום:

```python
rocc = NearestCentroid()

param_grid = {'metric': ['euclidean', 'manhattan', 'minkowski'], 'shrink_threshold': [None, 0.1, 0.5, 1.0]}

grid_search = GridSearchCV(rocc, param_grid, cv=kfold, n_jobs=-1)

grid_search.fit(X_train, y_train)


print('Best hyperparameters: {}'.format(grid_search.best_params_))
print('Best score: {:.2f}'.format(grid_search.best_score_))

y_pred= grid_search.predict(X_test)


print(classification_report(y_test, y_pred))


cm = confusion_matrix(y_test, y_pred)

# Plot the confusion matrix as a heatmap
sns.heatmap(cm, annot=True, cmap='Blues')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()


misclassified_tokens = np.where(y_test != y_pred)[0:10]

filtered_df = df.loc[misclassified_tokens][['Tokens', 'id']]

print("documents classifed mine but they are not mine:\n")
print(filtered_df[filtered_df['id']== 1]['Tokens'].head(20))

print()

print("documents classifed not mine but they are  mine:\n")
print(filtered_df[filtered_df['id']== 0]['Tokens'].head(20))
```

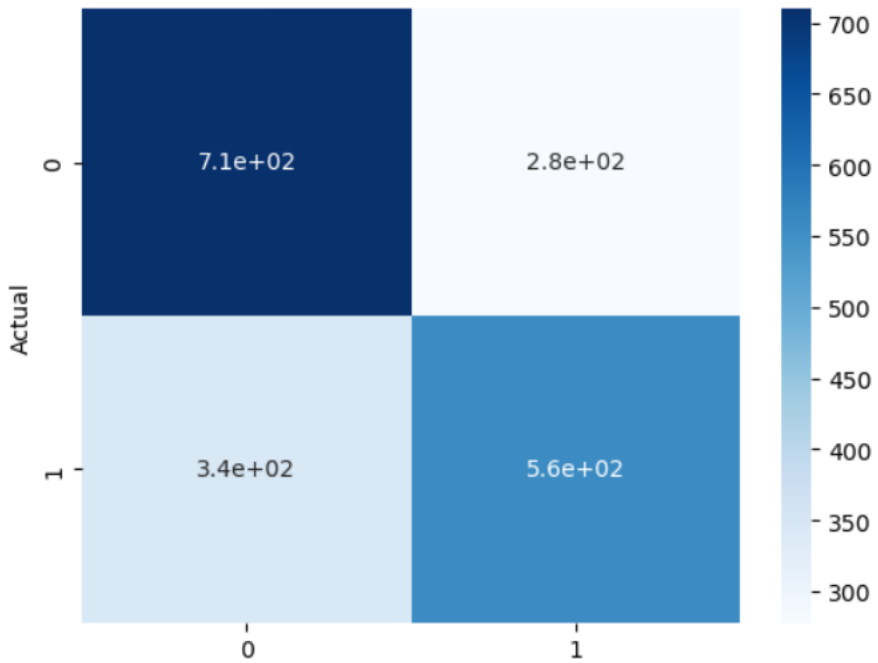
הסבר:

- metric: המדד המשמש לחישוב המרחק בין המרכזים לנקודות הנתונים. במקרה שלנו, האפשרויות הן אוקלידית, מנהטן ומינקובסקי.

- shrink_threshold: פרמטר רגוליזציה המסייע במניעת התאמת יתר(OVERFITTING). הוא מכווץ את המרכזים לכיוון הממוצע הכולל של הנתונים כאשר המרחקים שלהם קטנים מהסף. במקרה שלנו, האפשרויות הן ללא, 0.1, 0.5 ו-1.0.

תוצאות שקיבלנו:

```
Best hyperparameters: {'metric': 'manhattan', 'shrink_threshold': 0.5}
Best score: 0.67
              precision    recall  f1-score   support

           0       0.67      0.72      0.70       988
           1       0.67      0.62      0.64       901

    accuracy                           0.67      1889
   macro avg       0.67      0.67      0.67      1889
weighted avg       0.67      0.67      0.67      1889
```

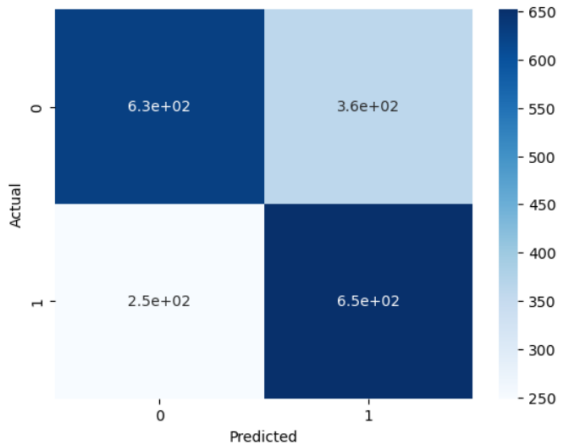


ניתן לראות פה שיש יותר איזון בתוצאות החיזויים של המודל הזה(נסביר בהמשך). ה-PRECISION של המסווג הוא 0.67 עבור מחלקה 0 ו-0.67 עבור מחלקה 1, מה שאומר שמתוך כל המסמכים שסווגו כמחלקה 0, 67% היו למעשה במחלקה 0, ומתוך כל המסמכים שסווגו כמחלקה 1, 67% היו בפועל במחלקה 1. ה-RECALL של המסווג הוא 0.72 עבור מחלקה 0 ו-0.62 עבור מחלקה 1, כלומר המסווג זיהה נכון 72% מכל מסמכי מחלקה 0 ו-62% מכל מסמכי מחלקה 1.

```
documents classifed mine but they are not mine:

11      level ofunderstanding sufficient effective xa...
12     com/loi/uism20explainable artificial intellige...
13     il literature acquisition analysisthis detaile...
19      specifically, seek answer-how attacker deceiv...
38      compute metrics takencarefully tasks applicat...
48      number published articles (y axis) xai biblio...
51      lime [local interpretable model (agnostic) ex...
66     arun [82] proposed concerns role xai ininfluen...
67     050contribution cc ceteris paribus09 pfn [mg/l...
69     note: interpret representedby color mission di...
75      computation kurtosis skewnesscan represented ...
77     markus langer, daniel oster, timo speith, holg...
86     in contribution, exploring ways ren-der machin...
88     , taxi origin point distributions collected ye...
92      usesdata grad-cam effort extend theevaluation...
93      besides,this technique extends grad-cam and, ...
95      understanding productionparameters process qu...
96     information communication technology advanceme...
99      addition renowned algorithms, aresome algorit...
101    88 nick wallace, xe2x80x9ceuxe2x80x99s explana...
Name: Tokens, dtype: object

documents classifed not mine but they are  mine:

4       breusch-godfrey durbin-wat-son tests serial c...
9       formally defne deviation, xf0x9dx9bxa5pa, fac...
14     x00)x00,x00rx00x00ax00dx00vx00ax00nx00cx00ex00...
25     x00 x00jx00ox00ux00rx00nx00ax00lx00 x00ox00fx0...
26     x00 x00ax00lx00lx00 x00tx00hx00ex00 x00wx00ex0...
28     x00 x00ux00nx00dx00ex00rx00 x00ax00lx00lx00 x0...
37      fusing atdd assurance caseswe propose combine...
42      ranking short texts anddocuments, twitterrank...
70     x00 x00rx00x00wx00hx00ex00nx00 x00ox00rx00gx00...
73     x00 x00lx00ex00tx00 x00(x00x16x00xx00;x00x16x0...
82     x00 x00tx00hx00ex00rx00ex00fx00ox00rx00ex00,x0...
87      presently, rise xe2x80x98intelligent machinex...
97      traditional legal theory derives interests co...
100    x00rx00x00[x006x009x00]x00 x001x00ix001x00yx00...
102    criteria construct bias free frameworkany appr...
107     ethodswe trained neural network models predic...
110     usually, abd performed strict set rules, prot...
113    r porale processing accent variation spoken la...
115    x00 x00kx00nx00ox00wx001x00ex00dx00gx00ex00rx0...
117     specifically, ppd experiencing mothers, rface...
Name: Tokens, dtype: object
```

<u>אחרי שימוש בתכונות שנבחרו על ידי ה-RFE:</u>

```
Best hyperparameters: {'metric': 'euclidean', 'shrink_threshold': 1.0}
Best score: 0.66
              precision    recall  f1-score   support

           0       0.72      0.63      0.67       988
           1       0.64      0.72      0.68       901

    accuracy                           0.68      1889
   macro avg       0.68      0.68      0.68      1889
weighted avg       0.68      0.68      0.68      1889
```



ה-PRESICION של המסווג הוא 0.72 עבור מחלקה 0 ו-0.64 עבור מחלקה 1, מה שאומר שמתוך כל המסמכים שסווגו כמחלקה 0, 72% היו למעשה במחלקה 0, ומתוך כל המסמכים שסווגו כמחלקה 1,

64% היו בפועל במחלקה 1. ה-RECALL של המסווג הוא 0.63 עבור מחלקה 0 ו-0.72 עבור מחלקה 1, כלומר המסווג זיהה נכון 63% מכל מסמכי מחלקה 0 ו-72% מכל מסמכי מחלקה 1.

ציון F1 הוא הממוצע ההרמוני של ה-RECALL ו PRESICION, המספק מדד לאיזון בין שני המדדים. ציון F1 הוא 0.67 עבור שתי המחלקות, מה שמצביע על ביצועים דומים יחסית של המסווגן עבור שתי המחלקות. ה-ACCURACY של המסווג הוא 0.68, מה שאומר שהוא סיווג נכון 68% מכלל המסמכים.

```
documents classifed mine but they are not mine:

11      level ofunderstanding sufficient effective xa...
12     com/loi/uism20explainable artificial intellige...
19      specifically, seek answer-how attacker deceiv...
38      compute metrics takencarefully tasks applicat...
48      number published articles (y axis) xai biblio...
51      lime [local interpretable model (agnostic) ex...
56      snapshot overview customer churninformation c...
66     arun [82] proposed concerns role xai ininfluen...
67     050contribution cc ceteris paribus09 pfn [mg/l...
75      computation kurtosis skewnesscan represented ...
77     markus langer, daniel oster, timo speith, holg...
86     in contribution, exploring ways ren-der machin...
88     , taxi origin point distributions collected ye...
92      usesdata grad-cam effort extend theevaluation...
93      besides,this technique extends grad-cam and, ...
95      understanding productionparameters process qu...
96     information communication technology advanceme...
99      addition renowned algorithms, aresome algorit...
101    88 nick wallace, xe2x80x9ceuxe2x80x99s explana...
105     figure 5 illustrates set stakeholders as-sess...
Name: Tokens, dtype: object

documents classifed not mine but they are  mine:

4       breusch-godfrey durbin-wat-son tests serial c...
9       formally defne deviation, xf0x9dx9bxa5pa, fac...
14     x00)x00,x00rx00x00ax00dx00vx00ax00nx00cx00ex00...
25     x00 x00jx00ox00ux00rx00nx00ax001x00 x00ox00fx0...
26     x00 x00ax001x001x00 x00tx00hx00ex00 x00wx00ex0...
28     x00 x00ux00nx00dx00ex00rx00 x00ax001x001x00 x0...
37      fusing atdd assurance caseswe propose combine...
42      ranking short texts anddocuments, twitterrank...
70     x00 x00rx00x00wx00hx00ex00nx00 x00ox00rx00gx00...
73     x00 x001x00ex00tx00 x00(x00x16x00xx00;x00x16x0...
87      presently, rise xe2x80x98intelligent machinex...
100    x00rx00x00[x006x009x00]x00 x001x00ix001x00yx00...
102    criteria construct bias free frameworkany appr...
107     ethodswe trained neural network models predic...
110     usually, abd performed strict set rules, prot...
113    r porale processing accent variation spoken la...
117     specifically, ppd experiencing mothers, rface...
123    rruser: preconditions running clinic here?rdoc...
127     rppd prediction rcomparison predictive models...
133     work alsoneeded extend variety types explanat...
Name: Tokens, dtype: object
```

4.6 KNN

KNN, הוא אלגוריתם למידת מכונה פופולרי המשמש במשימות סיווג טקסט. אלגוריתם ה-KNN פועל על ידי אימון תחילה על קבוצה של דוגמאות מתויגות, ולאחר מכן שימוש במרחקים בין הדוגמאות החדשות, ללא תווית, או נתוני הבדיקה, לבין הדוגמאות המתויגות כדי לסווג את הדוגמאות החדשות.

בסיווג טקסט, ניתן להשתמש ב-KNN כדי למצוא את k השכנים הקרובים ביותר לדוגמאות הבדיקה בנתוני האימון, כאשר "הקרוב ביותר" מוגדר על ידי מדד מרחק, כגון מרחק אוקלידי או דמיון קוסינוס, בין וקטורי הטקסט. המחלקה של רוב ה-k השכנים הקרובים ביותר משמשת לאחר מכן לסיווג נתונים לבדיקה.

יתרון אחד בשימוש ב-KNN לסיווג טקסט הוא שזהו אלגוריתם פשוט וניתן לפירוש וכמו כן לא צריך אימון המודל. בנוסף, KNN אינו מניח הנחות לגבי ההתפלגות הבסיסית של הנתונים ויכולה להיות יעילה עם מערכי נתונים גדולים. עם זאת, KNN יכול להיות יקר מבחינה חישובית, במיוחד כאשר מספר התכונות, או גודל וקטור הטקסט, גדל. זה גם דורש הגדרת הערך של k, מה שיכול להיות מאתגר במקרים מסוימים.

יישום:

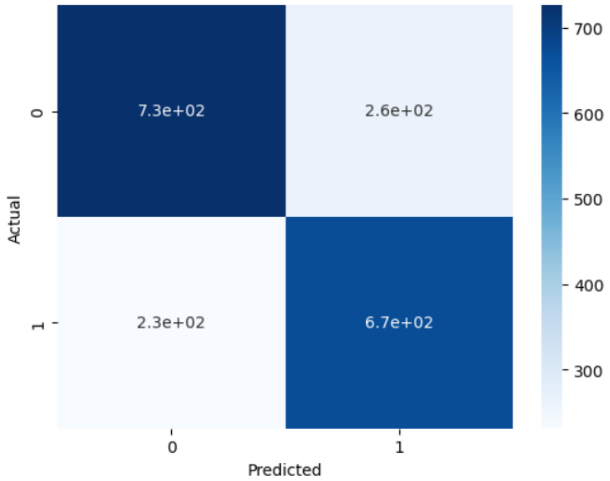
```
knn = KNeighborsClassifier()

param_grid = {'n_neighbors': [3, 5] , 'p': [1, 2]}

grid_search = GridSearchCV(knn, param_grid, cv=kfold, n_jobs=-1)

grid_search.fit(X_train, y_train)


print('Best hyperparameters: {}'.format(grid_search.best_params_))
print('Best score: {:.2f}'.format(grid_search.best_score_))

y_pred= grid_search.predict(X_test)


print(classification_report(y_test, y_pred))


cm = confusion_matrix(y_test, y_pred)

# Plot the confusion matrix as a heatmap
sns.heatmap(cm, annot=True, cmap='Blues')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()


misclassified_tokens = np.where(y_test != y_pred)[0:10]

filtered_df = df.loc[misclassified_tokens][['Tokens', 'id']]

print("documents classifed mine but they are not mine:\n")
print(filtered_df[filtered_df['id']== 1]['Tokens'].head(20))

print()

print("documents classifed not mine but they are  mine:\n")
print(filtered_df[filtered_df['id']== 0]['Tokens'].head(20))
```

- n_neighbors: היפרפרמטר זה שולט במספר השכנים הנחשבים בעת ביצוע חיזוי. ב-KNN, כל נקודת נתונים מסווגת על סמך מחלקת הרוב של K השכנים הקרובים ביותר. בקוד זה, אנו מחפשים את הערכים של 3 ו-5, כדי לראות איזה ערך נותן את הביצועים הטובים ביותר.

- p: היפרפרמטר זה שולט במדד המרחק המשמש לחישוב המרחק בין נקודות במרחב התכונה. ב-KNN, מדדי המרחק הנפוצים ביותר הם מרחק אוקלידי (p=2) ומרחק מנהטן (p=1). כאן אנו מחפשים את הערכים של 1 (מרחק מנהטן) ו-2 (מרחק אוקלידי).

תוצאות שקיבלנו:

```
Best hyperparameters: {'n_neighbors': 5, 'p': 2}
Best score: 0.75
              precision    recall  f1-score   support

           0       0.76      0.73      0.75       988
           1       0.72      0.74      0.73       901

    accuracy                           0.74      1889
   macro avg       0.74      0.74      0.74      1889
weighted avg       0.74      0.74      0.74      1889
```



ה-PRESICION של המסווג הוא 0.76 עבור מחלקה 0 ו-0.72 עבור מחלקה 1, מה שאומר שמתוך כל המסמכים שסווגו כמחלקה 0, 76% היו למעשה במחלקה 0, ומתוך כל המסמכים שסווגו כמחלקה 1, 72% היו בפועל במחלקה 1. ה-RECALL של המסווג הוא 0.73 עבור מחלקה 0 ו-0.74 עבור מחלקה 1, כלומר המסווג זיהה נכון 73% מכל מסמכי מחלקה 0 ו-74% מכל מסמכי מחלקה 1.

ציון F1 הוא הממוצע ההרמוני של ה-RECALL ו PRESICION, המספק מדד לאיזון בין שני המדדים. ציון F1 הוא 0.73 עבור שתי המחלקות, מה שמצביע על ביצועים דומים יחסית של המסווגן עבור שתי המחלקות. ה-ACCURACY של המסווג הוא 0.74, מה שאומר שהוא סיווג נכון 74% מכלל המסמכים.

```
documents classifed mine but they are not mine:

12     com/loi/uism20explainable artificial intellige...
18      niethammer, xe2x80x9cmultiple instance learni...
50     xc2xae companies associated, hand, witha colle...
51      lime [local interpretable model (agnostic) ex...
54     actions defined ayes = {atg6, ge, aie6/ttegt w...
56      snapshot overview customer churninformation c...
64      work providedcomprehensive xai categorization...
67     050contribution cc ceteris paribus09 pfn [mg/l...
77     markus langer, daniel oster, timo speith, holg...
86     in contribution, exploring ways ren-der machin...
88     , taxi origin point distributions collected ye...
96     information communication technology advanceme...
105     figure 5 illustrates set stakeholders as-sess...
108     moreover, survey slrguideline review methods ...
128     science approaches tackle problem explainable...
129    xe2x80x99xe2x80x9d essential detect atrial fib...
132     hand, hitachi abb confirmedthat identified pr...
135     challenging prohibit harmful domains usingcom...
139     kagal, explain-ing explanations: approach eva...
151     figure 12 displays temporal spatialimportance...
Name: Tokens, dtype: object

documents classifed not mine but they are  mine:

3      rrthe dnn decisions explained de-rscriptions r...
4       breusch-godfrey durbin-wat-son tests serial c...
14     x00)x00,x00rx00x00ax00dx00vx00ax00nx00cx00ex00...
26     x00 x00ax001x001x00 x00tx00hx00ex00 x00wx00ex0...
28     x00 x00ux00nx00dx00ex00rx00 x00ax001x001x00 x0...
36      respondents national news reportsrrwere biase...
42      ranking short texts anddocuments, twitterrank...
49      large regional national mainstream media inte...
70     x00 x00rx00x00wx00hx00ex00nx00 x00ox00rx00gx00...
73     x00 x001x00ex00tx00 x00(x00x16x00xx00;x00x16x0...
78     1 general purpose argumentation systemsrras ge...
82     x00 x00tx00hx00ex00rx00ex00fx00ox00rx00ex00,x0...
87      presently, rise xe2x80x98intelligent machinex...
102    criteria construct bias free frameworkany appr...
113    r porale processing accent variation spoken la...
123    rruser: preconditions running clinic here?rdoc...
127     rppd prediction rcomparison predictive models...
134    x00rx00x00tx00hx00ex00rx00x00ax00cx00tx00ix00o...
138     aggregation bias (or ecological fallacy) aris...
148     instance, evans (2008) reports thedecrease ag...
Name: Tokens, dtype: object
```

החלטנו לא לעשות עוד מסווג KNN לתכונות שנבחרו על ידי ה- RFE כי ה- KNN לקח הרבה זמן כדי לעשות סיווג ולמידת ההיפרפרמטרים גם לקחה זמן רב

4.7 דיון לגבי התוצאות והשוואה בין מודלים וכמו כן הסבר על למה קיבלנו ציונים אלו במדדים של הערכה

KNN

	precision	recall	f1-score	support
0	0.76	0.73	0.75	988
1	0.72	0.74	0.73	901
accuracy			0.74	1889
macro avg	0.74	0.74	0.74	1889
weighted avg	0.74	0.74	0.74	1889

Rocchio

	precision	recall	f1-score	support
0	0.67	0.72	0.70	988
1	0.67	0.62	0.64	901
accuracy			0.67	1889
macro avg	0.67	0.67	0.67	1889
weighted avg	0.67	0.67	0.67	1889

Bernoulli Naïve Bayes

	precision	recall	f1-score	support
0	0.89	0.42	0.57	988
1	0.60	0.94	0.73	901
accuracy			0.67	1889
macro avg	0.74	0.68	0.65	1889
weighted avg	0.75	0.67	0.65	1889

Gaussian Naïve Bayes

	precision	recall	f1-score	support
0	0.96	0.34	0.51	988
1	0.58	0.98	0.73	901
accuracy			0.65	1889
macro avg	0.77	0.66	0.62	1889
weighted avg	0.78	0.65	0.61	1889

ניתן לראות שאלגוריתם KNN עבד הכי טוב במקרה שלנו מבחינת כלל המדדים, לכן אם הייתה לנו משימה לבחור מסווג היינו בוחרים את KNN אבל ברמת בטיחות לא כל כך גבוהה. אחריו בא האלגוריתם של ROCHHIO שהוא גם נתן ביצועים טובים וכמעט מאוזנים בין כלל המדדים כמו ב-KNN. המודלים של NAÏVE BAYES נתנו תוצאות לא טובות וגם ציוני המדדים אינם מאוזנים בין שתי המחלקות.

הסיבה לקבלת תוצאות מאוזנות (כלומר, ערכי PRESCION ו RECALL דומים עבור שתי המחלקות) היא ככל הנראה משום שמסווג Rocchio מתאים את גבול ההחלטה על סמך המרחק של כל מסמך למרכז של כל מחלקה. בדרך זו, הוא יכול להתמודד עם מערכי נתונים לא מאוזנים טוב יותר מאשר מסווגים אחרים, שעלולים להיות מוטים למחלקת הרוב. ועבור KNN הוא יכול לבצע ביצועים טובים עבור משימות סיווג טקסט מכיוון שהוא יכול ללכוד את המבנה המקומי בנתונים ויכול להשתמש במבנה זה כדי לבצע תחזיות. בסיווג טקסט, משמעות הדבר היא ש-KNN יכול לזהות דוגמאות טקסט דומות ולהשתמש בהן כדי לבצע תחזיות. אם דוגמאות הטקסט בכל מחלקה דומות מבחינת השימוש במילה והקשר שלהן, אז KNN יכול לזהות את קווי הדמיון הללו ולהציג ביצועים טובים בניבוי תווית המחלקה הנכונה. בנוסף, מכיוון ש-KNN אינו מניח הנחות לגבי התפלגות הנתונים, הוא יכול להתמודד עם

גבולות החלטה לא ליניאריים ואינטראקציות מורכבות של התכונות וכמו כן יכול להתמודד עם מערכי נתונים לא מאוזנים, מה שיכול להועיל במשימות סיווג טקסט.

בניגוד ל- ROCCHIO ו KNN המודלים NAÏVE BAYES נתנו תוצאות לא מאוזנות ואין לנו אפשרות למתן בטיחות בתוצאות שלהם. התוצאה יכולה להיגרם מהסיבה שמודלים אילו מניחים הנחה מאוד רגישה לגבי המונחים במסמכים וההנחה היא שהתכונות שלנו בלתי תלויים אחד בשני.

<u>למה בכללי קיבלנו תוצאות לא כל כך טובות וסיווגים לא נכונים?</u>

יש כל כך סיבות בהקשר של המשימה שלנו, הסיבה הראשונה שעלולה להיות הגורם היא עיבוד מקדים לא טוב על הנתונים שלנו כלומר לא יצרנו תכונות באופן מקצועי. וכמו כן ייתכן שיש אי איזון בנתוני האימון כלומר מחלקה אחת מחסלת את השנייה. בנוסף לכך חיפוש ההיפרפרמטרים נעשתה באופן GREEDY כלומר עצרנו כאשר הגענו למינימום מקומי וייתכן שפעולה זו הייתה נעצרת מהר מדי. וכמו כן ייתכן שהמסמכים שסופקו לנו על ידי בסטודנטים בקורס משתמשים במונחים דומים ואז יהיה לכולם תכונות דומים וזה מקשה על סיווג ומציאת גבולות בין המחלקות. וייתכן שקרה לנו OVERFITTING אן UNDERFITINNG בצורה כלשהי. אם נסתכל למעלה על המסמכים שסווגו לא נכון נראה שזה בגלל שמסמכים רבים מכילים הרבה מונחים של המסמכים שלי והם לא מדברים על אותו נושא מה שמקשה על אלגוריתמים כמו למשל NAÏVE BAYES לזהות דברים כאלה במיוחד כי הם מניחים אי תלות.

5 *Text clustering*

אשכול טקסט, הידוע גם בשם אשכול מסמכים, הוא המשימה לקבץ קבוצה של מסמכי טקסט לאשכולות על סמך קווי הדמיון ביניהם. אשכול כרוך בחלוקת המסמכים במערך נתונים לקבוצות, כך שהמסמכים בתוך כל קבוצה חולקים מאפיין משותף כלשהו, תוך שהם שונים מאלה שבקבוצות אחרות. המטרה של אשכול טקסט היא לזהות תבניות ומבנה בסיסי באוסף גדול של נתוני טקסט לא מובנים, ולהקל על ניווט ושליפה של המסמכים. תהליך אשכול טקסט כולל חילוץ תכונות מנתוני הטקסט, כגון TF או ציוני TF-IDF, ולאחר מכן שימוש באלגוריתמים של clustering כגון K-MEANS.

5.1 התהליך

ראשית התחלנו את התהליך על ידי קריאת רשימת המילות עצורה שצוות הקורס נתן לנו:

```
In [3]:
stop_words = pd.read_csv('stop_words_english.txt', delimiter=',')
stop_words = list(stop_words['able'])
stop_words
```

```
'abroad',
'according',
'accordingly',
'across',
'actually',
'adj',
'after',
'afterwards',
'again',
'against',
'ago',
'ahead',
"ain't",
'all',
'allow',
'allows',
'almost',
'alone',
'along',
'alongside',
```

אחרי זה השתמשנו בפונקציה שתקרא לנו את המסמכים ותוך כדי תבצע ניקוי למסמכים הללו:

```
In [6]:
def read_text(dir):

    # variable for the text
    corpus = ""

    # names of all the documents in the directories
    names = os.listdir(dir)

    # loop through names
    for name in names:

        # create a file path with the curent name
        file_path = dir + os.sep + name
        if name.endswith(".txt"):

            # open the file and read text
            with open(file_path, "rb") as f:
                content = f.read()
                content = str(content)

                # add the text to corpus
                corpus = corpus + content

    # remove unwanted characters from the text
    corpus = corpus.replace("\\n", "")
    corpus = corpus.replace("\\", "")

    # case fold - lower the corpus
    corpus = corpus.lower()

    # remove stop words
    # list of stop words
    corpus = [word for word in corpus.split() if word.lower() not in stop_words]
    corpus = " ".join(corpus)


    # return the corpus
    return corpus
```

הסברנו על הפונקציה קודם לכן. אחרי זה אני ושותפתי תייגנו את המסמכים כדי לבדוק ביצעיו של אלגוריתם KMEANS, התיוג היה בהתאם למסמכים שבחרנו מהתקיות בדרייב:

```
In [11]:
# make a dataframe out of tokens

# df for tokens
df1 = pd.DataFrame({"Tokens": d1, "id": 0})
df2 = pd.DataFrame({"Tokens": d2, "id": 1})
df3 = pd.DataFrame({"Tokens": d3, "id": 2})
df4 = pd.DataFrame({"Tokens": d4, "id": 3})

# join df1 to df4
df = pd.concat([df1, df2, df3, df4])

# shuffle the final df
df = df.sample(frac=1).reset_index(drop=True)
```

Our documents	0
artificial intelligence explainability	1
Twitter bias	2
Social bias	3

ואז ייצגנו את המסמכים על ידי ווקטורים בייצוג TF-IDF וקיבלנו את הווקטורים הבאים:

In [16]:
```
x
```
Out[16]:

	0	1	2	3	4	5	6	7	8	9	...	90	91	92	93
0	-0.147688	-0.103717	0.437612	-0.419473	0.105552	0.185115	0.002213	0.571502	0.377865	-0.277576	...	0.209140	-0.068945	0.394455	-0.359389
1	-0.122479	-0.078429	0.353057	-0.345364	0.088037	0.138709	0.011516	0.476084	0.303385	-0.232388	...	0.178396	-0.051862	0.320302	-0.286168
2	-0.012477	-0.013976	0.046950	-0.047184	0.010093	0.021088	-0.000921	0.055168	0.044393	-0.031711	...	0.018103	-0.009128	0.040873	-0.035295
3	-0.068107	-0.047400	0.244812	-0.229788	0.055199	0.097585	-0.006500	0.306699	0.198929	-0.154059	...	0.116474	-0.040473	0.212982	-0.194265
4	-0.258854	-0.015518	0.869176	-0.807350	0.177261	0.148859	0.048876	1.201658	0.564582	-0.581812	...	0.605361	-0.059494	0.729059	-0.587536
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
18148	-0.242359	-0.116438	0.708239	-0.697894	0.179522	0.245627	0.037065	0.956968	0.591342	-0.480403	...	0.404581	-0.090585	0.625084	-0.558436
18149	-0.141403	-0.064871	0.433855	-0.424131	0.107525	0.141033	0.013883	0.593832	0.339667	-0.285615	...	0.250069	-0.054487	0.386232	-0.329486
18150	-0.079239	-0.077100	0.204744	-0.206292	0.051061	0.114525	0.001765	0.272016	0.200028	-0.129145	...	0.074767	-0.036491	0.202339	-0.188287
18151	-0.233379	-0.111286	0.728265	-0.705165	0.180054	0.223331	0.017014	1.008782	0.570043	-0.482116	...	0.414368	-0.093620	0.654882	-0.561961
18152	-0.074951	-0.076304	0.219233	-0.219269	0.055419	0.129006	0.000857	0.269051	0.225369	-0.139503	...	0.083112	-0.044750	0.201765	-0.202167

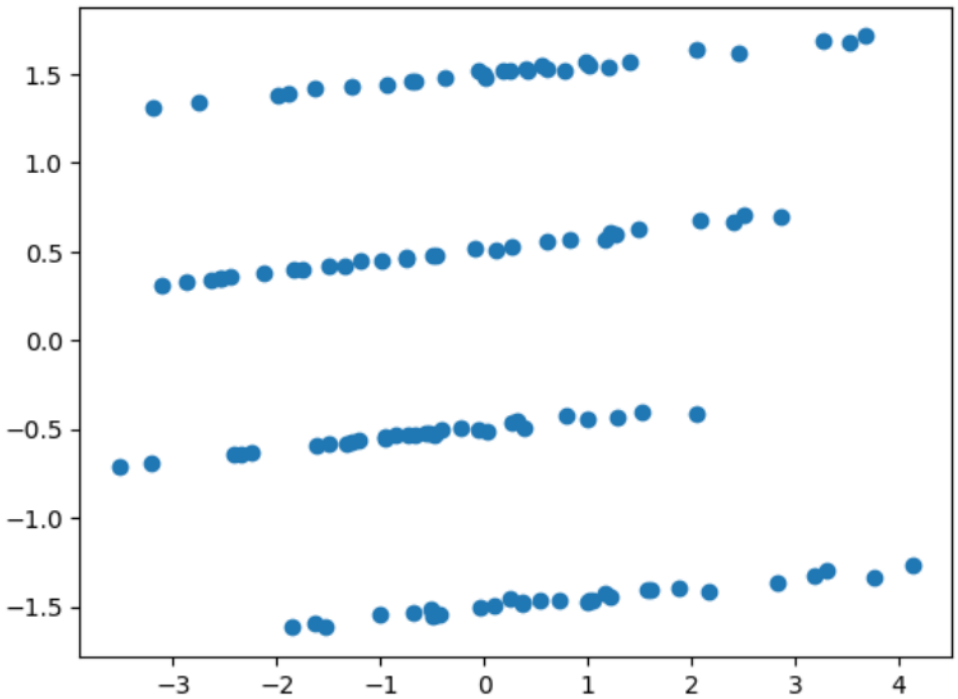
לצורך ייעול התהליך של הקיבוץ בחרנו מדגם מייצג של 120 תצפיות באופן רנדומלי:

```
x_0 = x[x['label'] == 0].copy()
x_1 = x[x['label'] == 1].copy()
x_2 = x[x['label'] == 2].copy()
x_3 = x[x['label'] == 3].copy()
```

```
subset_0 = x_0.sample(n=30, random_state=np.random.RandomState())
subset_1 = x_1.sample(n=30, random_state=np.random.RandomState())
subset_2 = x_2.sample(n=30, random_state=np.random.RandomState())
subset_3 = x_3.sample(n=30, random_state=np.random.RandomState())
```

```
new_x = pd.concat([subset_0, subset_1, subset_2, subset_3])
new_x
```

5.1.1 ביצענו תהליך של הורדת ממדיות על ידי שימוש ב PCA לצורך קבלת תחושה על המסמכים:



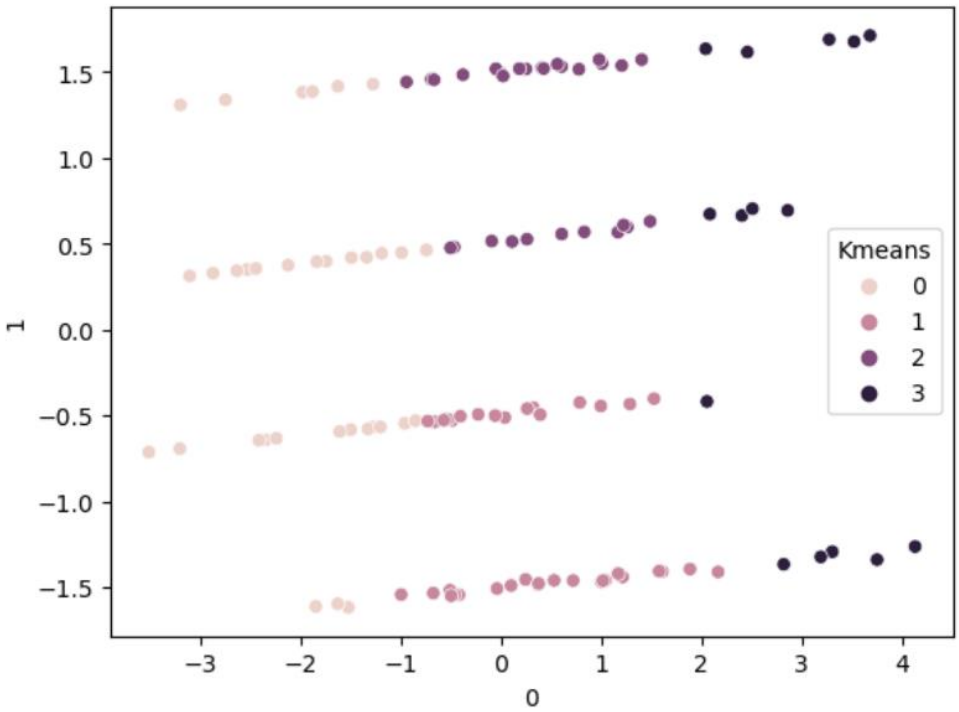
השאלה עכשיו היא האם KMEANS יצליח באמת להביא אותנו ל 4 אשכולות? התשובה שלנו לפני ביצוע האלגוריתם היא לא כי KMEANS לא מצליח הרבה באשכול מסמכים אם צורת ההתפלגות שלהם היא לא כדורית.

השתמשנו באלגוריתם וביצענו את האשכול.

5.1.2 ביצוע KMEANS ו תוצאות שקיבלנו ו הצגת גרפים והסבר למה היה טעויות

```
kms = KMeans(n_clusters=4)
kms.fit(pd.DataFrame(x_reduced_new))
```

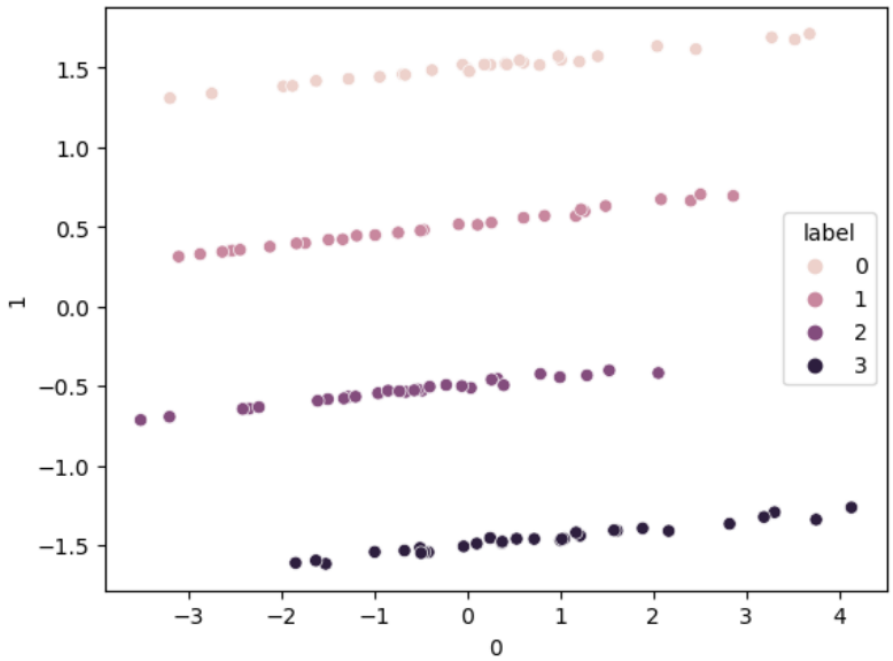
```
[30]: KMeans(n_clusters=4)
```



מה שציפינו אליו בהתייחס לתגיות שנתנו למסמכים:

```
sns.scatterplot(data=X_prev_df, x=0, y=1, hue='label')
```

```
]: <AxesSubplot:xlabel='0', ylabel='1'>
```

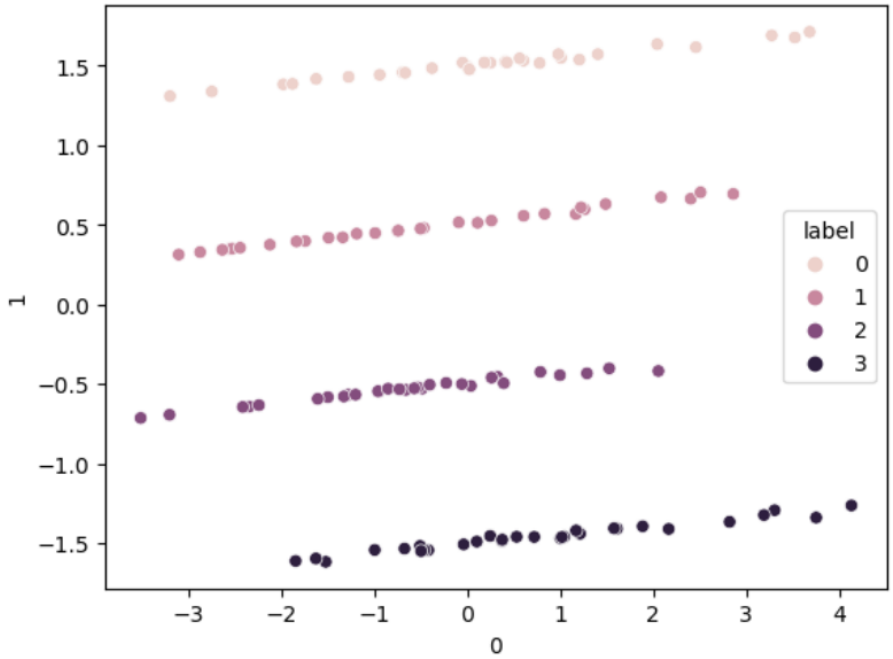


הגרף הראשון מראה תוצאות של KMEANS שנתן ביצועים לא טובים על הנתונים שלנו. יש כמה סיבות למה KMEANS לא עבד טוב, ראשית הממדיות של הנתונים שלנו הינה גבוהה מה שיקשה על KMEANS בחישוב המרחקים שהוא מחשב, וכמו כן אולי השימוש במרחק האוקלידי הוא מדד לא טוב לביצוע המשימה הזו. לבסוף KMEANS מאוד רגיש להפלגות הנתונים ומאוד לא יעיל כאשר הנתונים אינם כדוריים, ייתכן ש-k-means clustering לא יעבוד טוב מכיוון שההנחה הבסיסית של אשכולות איזוטרופיים מופרת. נתונים שאינם כדוריים יכולים לכלול אשכולות מוארכים, אשכולות בעלי צורה לא סדירה או אשכולות עם צפיפות או שונות משתנים. במקרים אלה, האשכולות עלולים לחפוף או להיות בעלי גבולות מורכבים שקשה ל-k-means להפרידם.

לבסוף אנחנו לא היינו נותנים אימון בתוצאות ה-KMEANS ולא יכלנו להניב מסקנות אחרי הרצת ה KMEANS, התיוג הידני היה יותר ברור:

```
sns.scatterplot(data=X_prev_df, x=0, y=1, hue='label')
```

```
]: <AxesSubplot:xlabel='0', ylabel='1'>
```



תודה שקראת את כל העבודה!