טקסט כנתונים
עיבוד שפה טבעית וניתוח רשתות
פרויקט הקורס

מרצה: מיה שטמר

תאריך הגשה: 22\06\23

מגישים: עוז גיל – און 316468586

עידו דוויק 318321627

1) <u>בחירת נושא והצגת הנתונים</u>

1.1) חלק א': ניתוח רשתות

בסיס הנתונים שבחרנו לעבוד איתו נלקח מאתר **Kaggle**, בסיס הנתונים הוא של קווי הרכבת בצרפת, והנסיעות שבוצעו בקווים אלו בין השנים 2015-2021. לנתונים האלו ביצענו ממוצע והנתונים שמוצגים הם ממוצע לפי ששת השנים הללו.

לכל קו בבסיס הנתונים ניתנה ביקורת על ידי נוסע שנסע בקו, כך שבסופו של דבר ניתחנו 130 ביקורות.

1.2) הצגת הנתונים

מבנה בסיס הנתונים בעבור ניתוח רשתות:

בסיס הנתונים שאיתו עבדנו מורכב משדות של: תאריך, מספר סידורי של הביקורת, תחנת יציאה, תחנת הגעה, מרחק (מ'), זמן נסיעה ממוצע, מס' נוסעים בשש שנים, מס' נסיעות צפוי ב6 שנים, מס' הגעות כולל לתחנה היעד, וביקורת.

סינון בסיס הנתונים:

בתחילת העבודה, בסיס הנתונים היה מורכב מ32 עמודות לא רלוונטיות עבורנו ועבור הניתוח שביצענו. מלבד זאת, הנתונים תאמו לשנה אחת ולא לכלל השנים בהם נאספו הנתונים. ביצענו את ההתאמות הנדרשות כגון: הסרת העמודה של מספר הפעמים בהם בוטל הקו בשנה האחרונה, מספר הפעמים שהקו איחר, איחור ממוצע לפי דקות, מספר איחורי בשל שיפוצים, וכדומה.

כך נראה בסיס הנתונים המועדכן:

	A	B	C	D	E	F	G	H	I	J	K	
1	Year	Review_ID	Departure station	Arrival stat	Destinatior	Average tra	Average Pass	Number of	Expectanc	Average Passen	Arrivals	Review_Text
2	2015-2	1	ANGOULEME	PARIS MO	20255	9272.484	46,361,091.00	18985	0.0092	10,255,175.00	16	Avoid Ouigo if you can. We booked the high speed Ouigo tra
3	2015-2	2	PARIS MONTPARNASSE	LA ROCHE	28768	11880.25	56,154,088.00	13253	0.00642	8,043,951.00	1	Worst train service ever. I had to take it for an year between
4	2015-2	3	LE MANS	PARIS MO	43544	3889.922	49,812,731.00	27621	0.01338	7,799,889.33	16	I just took one of the new Thalys trains between Le Mans anc
5	2015-2	4	ST MALO	PARIS MO	38180	11591.6	41,837,717.00	5973	0.00289	9,505,842.33	16	A very comfortable and hassle-free trip to Paris from St Malc
6	2015-2	5	PARIS MONTPARNASSE	ST PIERR	36325	4188.106	52,786,286.00	26317	0.01275	7,829,200.17	1	Very bad experience. Train was relayed an hour, disorganize
7	2015-2	6	PARIS MONTPARNASSE	TOULOUS	36044	19373.92	57,420,390.00	9967	0.00483	8,584,410.00	1	I travelled from Paris to Toulouse today and I was utterly disa
8	2015-2	7	TOULOUSE MATABIAU	PARIS MO	39251	19619.39	46,676,477.00	11008	0.00533	9,242,131.50	16	The worst train service I habe experienced....they are slaps i
9	2015-2	8	PARIS EST	METZ	28821	5633.795	43,262,580.00	18906	0.00916	7,973,805.67	1	
10	2015-2	9	PARIS EST	REIMS	2362	3082.998	59,488,882.00	13331	0.00646	9,423,410.67	1	I travelled alone with my 2 year old son from Paris EST to Me
11	2015-2	10	PARIS NORD	DOUAI	39831	4468.981	57,749,640.00	11781	0.00571	7,502,627.67	1	We took a train first class from Paris Nord to Reims. The firs
12	2015-2	11	LYON PART DIEU	LILLE	39367	12446.36	41,387,457.00	16853	0.00817	8,806,220.17	3	Used the train to go from Charles De Gaulle Airport to Paris
13	2015-2	12	LILLE	MARSEILL	20234	19620.99	44,879,619.00	11783	0.00571	9,938,235.50	6	Resumen: Terrible company, terrible customer service, alwa
14	2015-2	13	TOURCOING	BORDEAU	43109	8087.952	57,893,494.00	643	0.00031	8,760,897.83	3	We rode the Train ? Grande Vitesse-Lyria on March 7 from T
15	2015-2	14	LYON PART DIEU	MARNE LA	37901	3181.29	55,829,165.00	7306	0.00354	8,332,880.00	2	Train delayed. Aisles and luggage areas filled with people wh
16	2015-2	15	MARSEILLE ST CHARLE	MARNE LA	17738	6185.04	61,022,016.00	7236	0.00351	9,536,551.17	2	I had a train booked from Marseille to Marne on April 17, 202
17	2015-2	16	ANNECY	PARIS LY(	12885	14719.6	50,601,734.00	10771	0.00522	8,926,152.83	24	A first class TGV fare used to mean food service, functioning
18	2015-2	17	CHAMBERY CHALLES L	PARIS LY(	26004	12041.08	60,739,859.00	12887	0.00624	9,629,902.67	24	The worst train service I have ever seen in the world.
19	2015-2	18	PARIS LYON	LYON PAF	17919	7960.986	59,581,017.00	37423	0.01813	9,313,133.50	6	Wifi is broken. Booking first class seat facing forward ,actua
20	2015-2	19	LYON PART DIEU	PARIS LY(	26494	7980.718	59,979,972.00	38498	0.01866	7,649,078.33	24	Great train directly to Paris Lyon. Comfortable seats with flip
21	2015-2	20	PARIS LYON	MARSEILL	5524	13197.46	58,309,838.00	29616	0.01435	8,640,340.17	6	We enjoyed our 1st Class TGV train trips, mostly on time, co

את הניתוחים ביצענו באמצעות תוכנות: Python, Gephi.

2) <u>ניתוח רשתות</u>

2.1) נתונים כללים על הרשת:

- צמתים: כלל התחנות ברכבת בצרפת, מחוז פריז (60 סה"כ)
- קשתות: כל קשת היא נסיעה אפשרית בין תחנה לתחנה – כ130 קשתות
- כיוון הרשת: כל רשומה בבסיס הנתונים מייצגת מסלול בין תחנה לבין תחנה אחרת – על כן, הגרף מכוון לפי צומת שיש מסלול בינה לבין צומת אחרת
- בסיס הנתונים המרכיב את הרשת, יש מספר תכונות אשר מאפשרות ניתוח ויזואלי מעניין: מרחק בין תחנה לתחנה, זמן ממוצע לנסיעה, כמות נוסעים ב6 שנים ומספר הנסיעות הצפויות במסלול הנוכחי

שאלות מחקר:

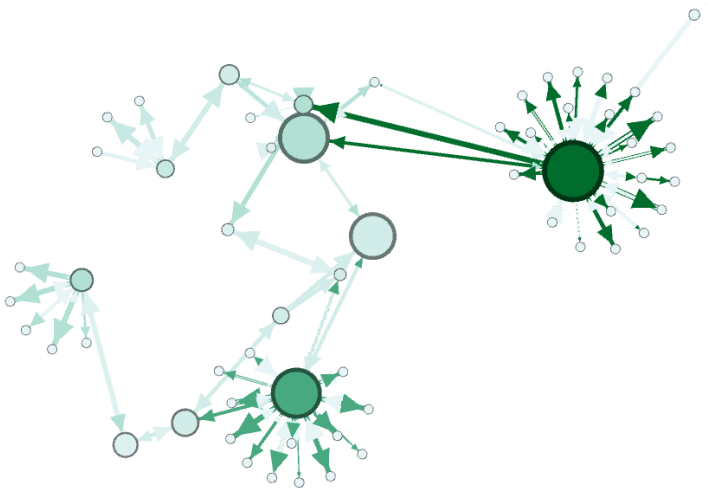
מהן הקהילות בגרף?

מהי התחנה שהכי הרבה קווים מגיעים אליה?

מהו המסלול המהיר ביותר להגיע מתחנה אחת לאחרת, הן מבחינת זמנים והן מבחינת מספר החלפות?

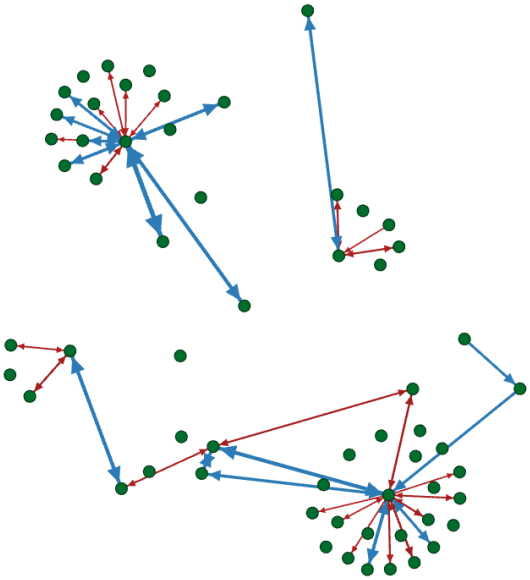
2.2) וויזואליזציה

בוויזואליזציה הראשונה שבחרנו להציג, הדגשנו את הדרגה של כל צומת על ידי הגדלת גודל הצומת בהתאם. בנוסף, גודל הקשת נקבע על פי המרחק בין תחנה לתחנה. לפי תצוגה זו נוכל להבין מי הצמתים (התחנות) הכי מקושרים בגרף ומי הצמתים הכי קרובים אחד לשני מבחינת מרחק.



מהגרף המתואר ניתן לראות כי יש מספר צמתים מרכזיים מהם יוצאים הרבה קשתות לעבר צמתים אחרות. כלומר, ניתן לומר שצמתים אלו הן בבסיסם תחנות מחוזיות מרכזיות שעל הנוסעים לעבור דרכם על מנת להגיע ליעדים ספציפיים.

בתצוגה הבאה, גודל הקשת מציין את מספר הנסיעות הצפויות בין תחנה לתחנה, כאשר כחול מסמן את הקשתות עם כמות רבה יותר של נסיעות ואדום מסמן את הקשתות עם פחות נסיעות. כמו כן, ניתן לראות שאנו רואים תת גרף של הגרף המקורי, היות והמטרה בתצוגה זו הייתה להתמקד אך ורק ב50% מהתחנות אשר מהוות את מספר הנסיעות הגבוה ביותר.

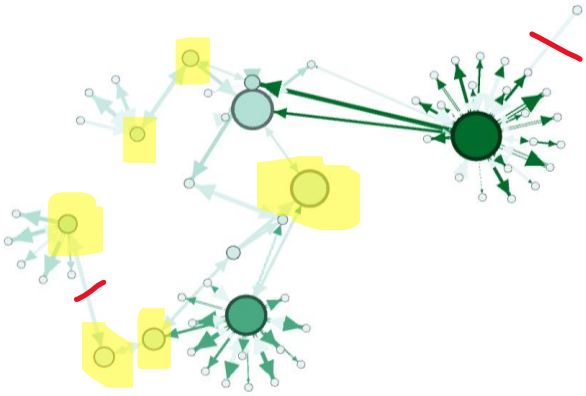


2.3) מטריקות עיקריות:

קוטר הגרף הוא 9. צפיפות הגרף הינה 0.037.

הגרף הינו גרף קשיר במובן החזק, היות וניתן להרכיב מסלול כלשהו בין כל 2 תחנות אקראיות.

הגרף כולל מס' נקודות שהן נקודות חיתוך: רוב התחנות המחוזיות הן נקודות חיתוך מאחר ובמידה ונסיר אותן, יהיו צמתים רבים ללא קשת, כלומר תחנות רבות שלא יהיה ניתן להגיע אליהן (מסומן במרקר צהוב) בנוסף, ישנם מספר גשרים ספציפיים אשר מחברים תחנות שלמות באמצעות קו יחיד (קו אדום). באמצעות מציאת הגשרים ונקודות החיתוך, נוכל ללמוד אילו תחנות הן המקשרות ואילו מסלולים הם המחברים בין תחנות ספציפיות.



2.4) מדדי מרכזיות:

ישנם 4 צמתים קריטיים בגרף עם מדד "בין מרכזיות" גדול משמעותית משאר הצמתים. הצמתים האלו הם התחנות המרכזיות והגדולות בהן מרוכזת רוב התחבורה של צרפת:
Paris Lyon, Lyon Part Dieu, Paris Montparnasse, Rennes. משמעות המרכזיות של הצמתים היא שהן הגורם המרכזי שמקשר בין חלקים שונים בגרף.

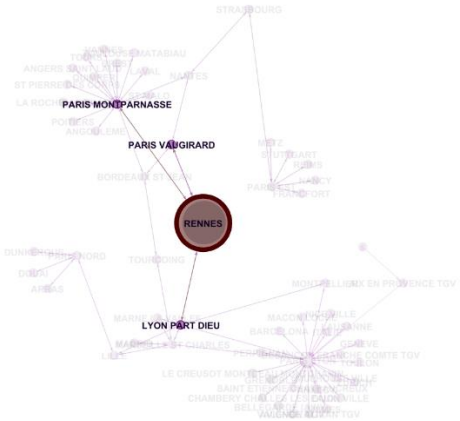
Nodes Edges Configuration Add node Add edg

Id	Betweenness Centrality
PARIS LYON	2049.5
LYON PART DIEU	1650.166667
PARIS MONTPARNASSE	1616.833333
RENNES	1483.666667

חלק מצמתים אלו הם גם בעלי הדרגה הגבוהה ביותר בגרף. בנוסף, לצמתים הבאים הדרגה הנכנסת והדרגה היוצאת הגבוהות ביותר (in/out degree):
Paris Lyon, Paris Montparnasse, Paris Est, Lyon Part.
על אף שלצמתים אחרים היה מדד מרכזיות גדול יותר, ההבדל בין הצמתים עם הדרגה הגבוהה יותר יכול לנבוע מהעובדה שלא כל צומת בעלת מסלול לכל צומת נוספת (למשל יש אפשרות לנסוע מתל אביב לבאר שבע אבל אין מסלול מבאר שבע לתל אביב).

Id	Degree	Weighted In-Degree	Weighted Out-Degree
PARIS LYON	49	25.0	24.0
PARIS MONTPARNASSE	32	16.0	16.0
PARIS EST	12	6.0	6.0
LYON PART DIEU	12	6.0	6.0

ניתן לראות הסבר מוחשי להבדל בצמתים בעלי ערך גבוהה של מדד המרכזיות לדרגה. התחנה **Rennes** היא בעלת מדד מרכזיות הרביעי בגודלו בגרף מאחר והיא מקשרת בין מספר רב של תחנות. אך הדרגה שלה היא 3 בלבד, כפי שניתן לראות היא איננה בעלת דרגה גבוהה כמו שאר הצמתים בעלי מדד מרכזיות גדול.



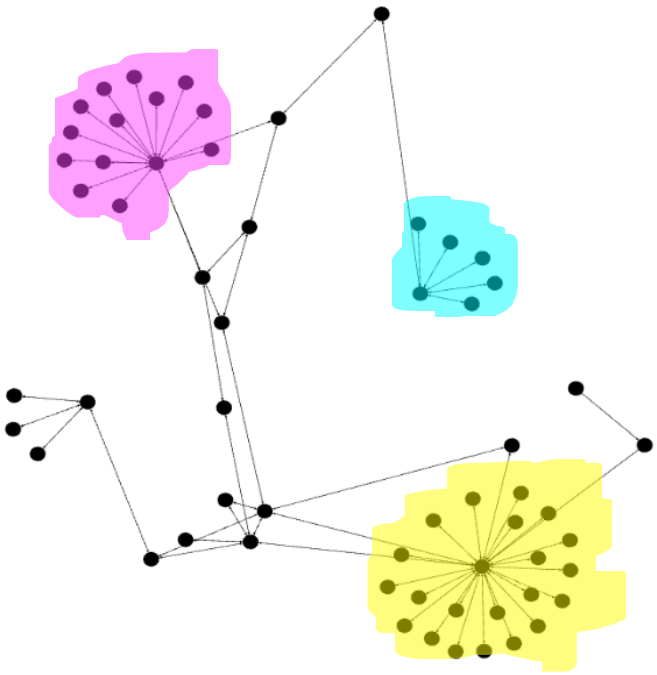
מדד ה-**closeness centrality** הוא [0.377, 0.160]. נמוך יחסית אך עולה בקנה ישיר עם העובדה שהגרף פרוס לאורך 60 צמתים שרובם לא קרובים אחד לשני.

Id	Closeness Centrality
LYON PART DIEU	0.377483
RENNES	0.358491
PARIS LYON	0.35625
PARIS MONTPARNASSE	0.337278
MARSEILLE ST CHARLES	0.32948
TOURCOING	0.314917
MONTPELLIER	0.308108

2.5) קהילות

מהחלוקה לקהילות שביצענו, ניתן לראות מי התחנות המקושרות יותר בתוך הקהילה מאשר מחוץ לקהילה. ניתן לראות שהתחנות המרכזיות שהוצגו למעלה מגדירות קהילות סביבן. ניתן לראות בבירור את החלוקה לקהילות בתצוגה הבאה.

הקהילות מייצגות את התחנות הקשורות אחת לשנייה, איפה מתקיים מטרופולין של תחנות בתוך כל הרשת, איפה אולי כדאי בעתיד לייצר מנוי ספציפי בין התחנות הללו על סמך שימוש נרחב בתוך הקהילה ושל אנשים בתוך אותה סביבה.



3) <u>עיבוד שפה</u>

3.1) עיבוד טקסט מקדים

במהלך העיבוד המקדים, ביצענו תהליך של הורדת אותיות גדולות, הסרת סימני פיסוק, הסרת StopWords, הסרת מספרים, פירוק כל ביקורת למילה נפרדת או במילים אחרות - ביצוע טוקניזציה, ולבסוף ביצוע Stemming כך שמילים עם סיומות דומות או פעלים יחושבו כאותה המילה. לדוגמא; (**train-trains**), (**ticket-tickets**), וכו'..

הקוד עבור תהליך העיבוד המקדים:

```
# Preprocessing
# Lowercasing
reviews_df = reviews_df.apply(lambda x: x.str.lower() if x.dtype == "object" else x)

# Removing punctuation
1 usage
def remove_punctuation(text):
    cleaned_text = re.sub(r'[^\w\s]', '', text)
    return cleaned_text
reviews_df['Review_Text'] = reviews_df['Review_Text'].apply(remove_punctuation)

#Stopwords & Tokenization
1 usage
def tokenize_and_remove_stopwords(text):
    tokens = nltk.word_tokenize(text)
    stopwords_list = stopwords.words('english')
    filtered_tokens = [word for word in tokens if word.lower() not in stopwords_list]
    return filtered_tokens

reviews_df['tokens'] = reviews_df['Review_Text'].apply(tokenize_and_remove_stopwords)
```

```
def remove_numbers(text):
    text_without_numbers = re.sub(r'\d+', '', text)
    return text_without_numbers

reviews_df['Review_Text'] = reviews_df['Review_Text'].apply(remove_numbers)

#Stemming
word_stemmer = PorterStemmer()
reviews_df['Review_Text'] = reviews_df['Review_Text'].apply(lambda x: ' '.join([word_stemmer.stem(word) for word in x.split()]))
print("After Stemming:", reviews_df.head(10))
```

ניתן לראות המחשה לעשרת הביקורות הראשונות, בתמונה שמוצגת מצד שמאל מופיע בסיס הנתונים ללא הליך עיבוד מקדים. בתמונה שמוצגת מצד ימין מופיע בסיס הנתונים לאחר הליך העיבוד המקדים. ניתן לראות עד כמה חשוב הכנת הבסיס לניתוח טקסט מאחר וקיימות מילים רבות ללא כל תוכן ממשי שניתן לנתח.

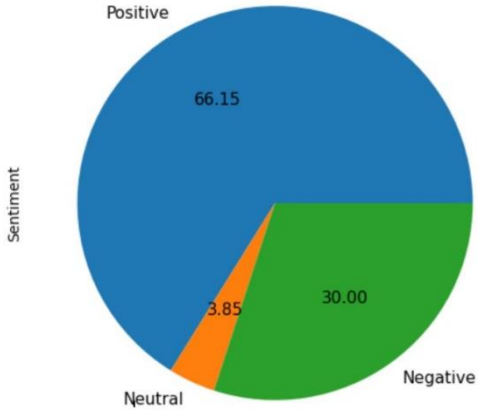
```
                                                Review_Text
Avoid Ouigo if you can. We booked the high spe...
Worst train service ever. I had to take it for...
I just took one of the new Thalys trains betwe...
A very comfortable and hassle-free trip to Par...
Very bad experience. Train was relayed an hour...
I travelled from Paris to Toulouse today and I...
The worst train service I habe experienced.......
The seats on this train were excruciatingly un...
I travelled alone with my 2 year old son from ...
We took a train first class from Paris Nord to...
```

```
avoid ouigo book high speed ouigo train pari a...
worst train servic ever take year pari la roch...
took one new thali train le man pari horrifi p...
comfort hasslefre trip pari st malo newli desi...
bad experi train relay hour disorgan number di...
travel pari toulous today utterli disappoint w...
worst train servic habe experiencedthey slap f...
seat train excruciatingli uncomfort reclin act...
travel alon year old son pari est metz first c...
took train first class pari nord reim first tr...
```

3.2) ניתוח סנטימנט

ביצענו ניתוח סנטימנטים לביקורות בעזרת **TextBlob**. חילקנו את הביקורות ל3 קטגוריות – חיובי, ניטרלי, ושלילי.

```
#sentiment analysis
TextBlob(' '.join(reviews_df['Review_Text'][0]))
TextBlob(reviews_df['Review_Text'][0]).sentiment

# Utility function to classify the polarity of a tweet using textblob.
1 usage
def analize_sentiment_blob(review):
    analysis = TextBlob(review)
    return analysis.sentiment.polarity

#Convert TextBlob polarity to Positive, Negative, or Neutral
1 usage
def analize_sentiment(review):
    analysis = TextBlob(review)
    if analysis.sentiment.polarity > 0:
        return 1
    elif analysis.sentiment.polarity == 0:
        return 0
    else:
        return -1
```



ציפינו שמירב הביקורות יהיו שליליות מאחר והנחנו שלנוסעים שלא נהנו במהלך נסיעתם או שחווית השירות שלהם הייתה שלילית, יהיה חשוב יותר להזהיר אחרים שלא להשתמש בשירות הרכבות, מאשר נוסעים מרוצים שממשיכים בחיי היום-יום שלהם. הופתענו לגלות שכ-66.15 אחוז מהביקורות היו דווקא חיוביות ואילו רק 30 אחוז מהביקורות היו שליליות.

על מנת להבין מהם המילים שהמבקרים שנתנו ביקורת חיובית או שלילית השתמשו בה, עברנו על בסיס הנתונים ומהם חילצנו את המילים הכי שכיחות.
המילים שחזרו על עצמן פעמים רבות בביקורות החיוביות היו: best, great, super, comfort, clean, love, enjoy, good.
ואילו, המילים הכי שכיחות בביקורות השליליות היו: avoid, bad, worst, terrible, cancel, horrible, poor.

3.3) מילות מפתח

```
top 50 frequent word are:
train: 297
ticket: 100
seat: 99
pari: 98
get: 82
time: 67
travel: 64
tgv: 62
servic: 60
class: 59
first: 55
trip: 54
one: 53
lyon: 52
would: 49
arriv: 42
hour: 41
station: 41
experi: 39
franc: 39
go: 38
cancel: 38
book: 37
```

```
#Key Words
tokenizer = nltk.tokenize.RegexpTokenizer(r'\w+')
reviews_df['tokens'] = reviews_df['Review_Text'].apply(nltk.word_tokenize)
text = ' '.join(review for review in reviews_df['Review_Text'])

word_freq = Counter([word for sublist in reviews_df['tokens'] for word in sublist])
top_freq_words = word_freq.most_common(50)

print("top 50 frequent word are: ")
for word, freq in top_freq_words:
    print(f"{word}: {freq}")
```

ממילות המפתח ניתן ללמוד מספר דברים. הראשונה היא שכמובן המילים 'רכבת' ו'כרטיס' מופיעות הכי הרבה כיאה לביקורות על קווי הרכבות. כמצופה, פריס וליון מופיעות הרבה פעמים כיאה לכך שישנן תחנות רבות בפריז ובפרט תחנה גדולה שנקראת **Paris Lyon**. דבר נוסף שניתן ללמוד הוא שהמילה **service** מופיעה גם כן הרבה אך לה יש מספר משמעויות, בעיקר מבחינת אופי הביקורת לגבי השירות אם היא חיובית או שלילית.

3.4) ניתוח נושאי

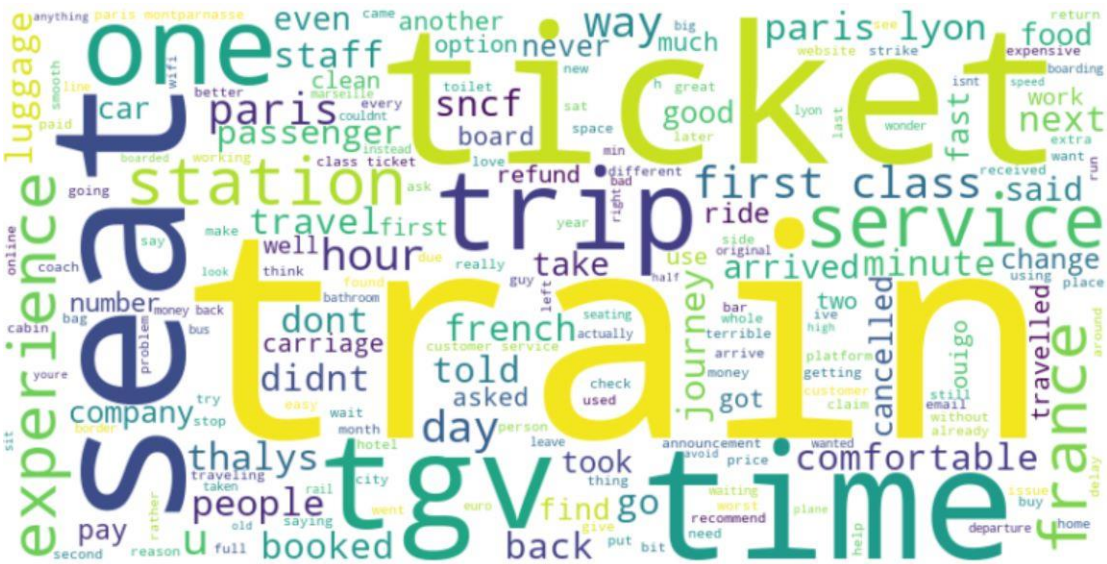
בתחילת התהליך של ניתוח נושאי, מידלנו את בסיס הנתונים לפי המודל '**BOW**', מודל הסוכם כמה מופעים יש מאותה המילה בבסיס הנתונים. לאחר מכן, הגבלנו את הוקטור ל-3000 מילים על מנת לאמוד את כמות המילים שברשותנו. גילינו שיש 2374 מילים שעלינו לנתח לאחר כל התהליך המקדים של הכנת הטקסט לעיבוד. ביצענו ניתוח נושאי ופירקנו כל מילה לערכים יחידיים (**SVD**). השתמשנו בשיטה **LSA** מאחר והוא יחסית פשוטה ואינטואיטיבית להבנה. היא מייצגת מסמכים ומילים כווקטורים, מה שמקל על פירוש היחסים והבנתם הסמנטית. ביצענו חלוקה ל-2 נושאים עיקריים לפי החלוקה של **SVD** על מנת להבין לאיזה נושא המילה מקבלת את הציון הגבוה ביותר. הציון נלקח לפי הערך המוחלט.

```
#BOW Model
subreviews = reviews_df['Review_Text'].iloc[0:3000]  # Extract the 'Review_Text' column for
vectorizer = CountVectorizer(max_features=3000)  # Limit the vocabulary to 1000 most freque
bow_model = vectorizer.fit_transform(subreviews)
print(bow_model.toarray())

# creating dataframe from bag of words matrix representation
df_bow = pandas.DataFrame(bow_model.toarray(), columns=vectorizer.get_feature_names_out())

#Topic Modeling
print("BOW Densed", bow_model.todense())
dense_matrix = reviews_df.values

svd = TruncatedSVD(n_components=2)# n_components = number of desired topics
lsa = svd.fit_transform(df_bow)

topic_encoded_df = pandas.DataFrame(lsa, columns = ['topic_1', 'topic_2'])
topic_encoded_df["df_bow"] = subreviews
display(topic_encoded_df[["df_bow","topic_1","topic_2"]])

dictionary = vectorizer.get_feature_names_out()
```

```
term_topic = pandas.DataFrame(svd.components_, index = ["topic_1","topic_2"], columns = dictionary).T

term_topic['abs_topic_1'] = np.abs(term_topic['topic_1'])
term_topic['abs_topic_2'] = np.abs(term_topic['topic_2'])
term_topic = term_topic.sort_values('abs_topic_1', ascending=False)
```

בדקנו ומצאנו שמילים כמו: **train, get, would, service, first, arrived, class** היו מהמילים שהכי מקושרות לנושא 1. ניתן להעריך שנושא אחד עשוי להיות לפי נושא של איכות שירות טובה שהנוסעים מרוצים ממנה או שממליצים על נסיעה ברכבת.
לגבי נושא 2, המילים שהכי מקושרות אליו הן: **tickets, said, asked, manager, told**. ניתן להעריך שנושא זה הוא לגבי תלונות הנוסעים, ואולי אף לנושאי שיח לקוח אל מול מנהל.

3.5) ענן מילים



לבסוף, הרכבנו ענן מילים.
הצלחנו לראות מילים בולטות כמו: **paris, service, seat, comfortable , lyon** .
התחנה **Paris Lyon** היא בין התחנות התחנות המרכזיות גם בגרף שהצגנו בחלק א' של פרויקט זה.
ניתן לראות בענן המילים בקטן יחסית את המילים השכיחות בניתוח הסנטימנטים שביצענו לביקורות השליליות, מילים כמו:**expensive, terrible, avoid** , וכו'..