B565 (Spring 2022) - DATA MINING
HOMEWORK 1

Name: Ameya Dalvi **Email: abdalvi@iu.edu**

1) Discuss why a document-term matrix is an example of a data set that has asymmetric discrete or asymmetric continuous features.

Answer:

A document-term matrix is a mathematical matrix that represents the frequency of terms in a set of documents. In the document term matrix, rows represent documents in the collection and the columns represent terms in the document.

Below is a representation of a Document-Term Matrix:

Doc1: I like Data Mining
Doc2: I like Data and I like Machine Learning

	I	**like**	**data**	**mining**	**and**	**machine**	**learning**
Doc1	1	1	1	1	0	0	0
Doc2	2	2	1	0	1	1	1

For the two given documents Doc1 and Doc2, the document matrix is represented by taking the unique words from both the documents and cataloguing their frequency in each of the documents.

Asymmetric discrete features are basically features for which the output values of a feature do not have equal relevance. In this example above, in Doc1 words “and” “machine” and “learning” do not occur, hence only the discrete non-zero frequency values in the document-term matrix, help us with identifying the documents’ contents thus rendering the zero-frequency values irrelevant. If the values are converted into continuous values, the same logic applies to them as well. Hence we can say that a document-term matrix is an example of a data set that has asymmetric discrete or asymmetric continuous features

2) Distinguish between noise and outliers. Be sure to consider the following questions.

- Is noise ever interesting or desirable? Outliers?
- Can noise objects be outliers?
- Are noise objects always outliers?
- Are outliers always noise objects?
- Can noise make a typical value into an unusual one or vice versa?

Answer:

Noise:
Noisy data is data that contains a substantial quantity of extra useless information known as noise. This involves data corruption, and the word is frequently used interchangeably with the term corrupt data.

Outliers:
Outlier is one single data point that differs significantly from all other data points in the dataset. Contrary to noise, it is not useless or unwanted, rather just a data point that falls beyond our expected range of values.

Is noise ever interesting or desirable? Outliers?
Noise is redundant data in your dataset that adds no value to your data, it is not desirable, whereas Outliers are data points that deviate from other data points in some way; they may be used to get significant insights on the dataset's nature.

Can noise objects be outliers?
Yes. There's a chance that some of the noise in the dataset is an outlier, and certain data distortions are desired.

Are noise objects always outliers?
No. Noise objects are not always outliers, noise could be null values in your data but outlier is a data point having potential of becoming a data point.

Are outliers always noise objects?
No. Outliers could become noise but not necessarily that that’s the case always.

Can noise make a typical value into an unusual one or vice versa?
Yes. Noise can affect the way the data is

3) Implement a notebook on Kaggle to explore this dataset. This dataset lists the number of antibiotic resistance genes (AMR), and the presence or absence of the CRISPR-Cas systems in the genomes included in the file. Report what you have learned by including the html output from your notebook in the PDF file you are going to submit. What to check? The distribution of the two variables (AMR, CRISPR-Cas), and if there is any correlation between the two variables.

We did an exploratory data analysis on the Efaecium_AMRC dataset, These are the steps followed for the same: Link for my Kaggle file

Importing Efaecium_AMRC dataset into a dataframe:

	genome_ID	CRISPR_Cas	AMR
0	GCA_010120755.1_ASM1012075v1	0	8
1	GCA_001720945.1_ASM172094v1	0	21
2	GCA_009697285.1_ASM969728v1	0	13
3	GCA_900639535.1_E8202_hybrid_assembly	0	11
4	GCA_002007625.1_ASM200762v1	0	18
...	...	...	...
2218	GCA_001058635.1_ASM105863v1	0	24
2219	GCA_900148625.1_Hp_24-1_05	0	16
2220	GCA_000981965.1_ASM98196v1	0	0
2221	GCA_002158235.1_ASM215823v1	0	10
2222	GCA_900080195.1_Isolate_3	0	18

2223 rows × 3 columns

Inference: Getting details of the dataset; There are three columns: “genome_ID”, ”CRISPR_Cas”, “AMR” and 2223 rows.

Getting a summary of whole data using pandas.DataFrame.descirbe()

	CRISPR_Cas	AMR
count	2223.000000	2223.000000
mean	0.024741	10.330184
std	0.155371	6.661470
min	0.000000	0.000000
25%	0.000000	3.000000
50%	0.000000	12.000000
75%	0.000000	16.000000
max	1.000000	31.000000

Inference: Initial distribution metrics like mean, standard deviation, minimum and maximum values, 1st Quartile, 2nd Quartile, 3rd Quartile values for all the columns.

Getting general information of the data using pandas.DataFrame.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2223 entries, 0 to 2222
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   genome_ID   2223 non-null   object
 1   CRISPR_Cas  2223 non-null   int64 
 2   AMR         2223 non-null   int64 
dtypes: int64(2), object(1)
memory usage: 52.2+ KB
```

Inference: Variable datatype, null value count etc can be observed.

Counting total number of null values in each column

```
genome_ID     0
CRISPR_Cas    0
AMR           0
dtype: int64
```

Counting total number of records for each output value for both the columns

```
data.value_counts("CRISPR_Cas")
```

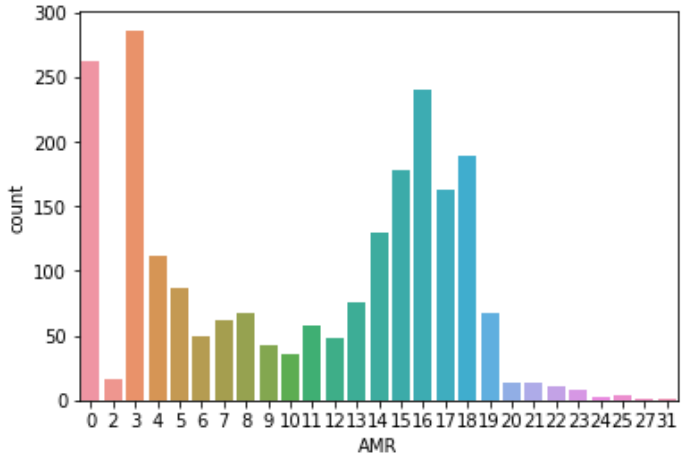
```
CRISPR_Cas
0    2168
1      55
dtype: int64
```

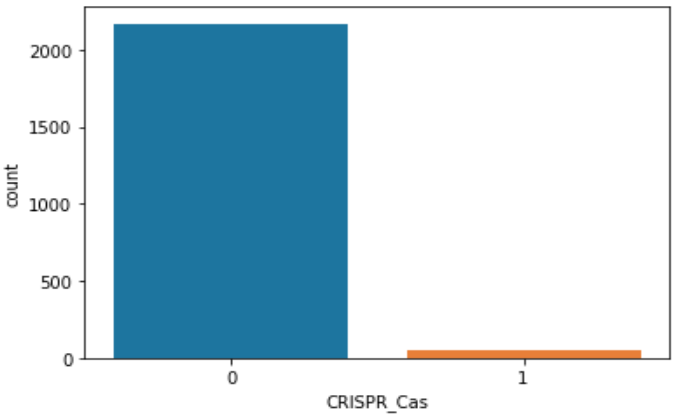
+ Code + Markdown

```
data.value_counts("AMR")
```

```
AMR
3     286
0     262
16    240
18    189
15    178
17    162
14    129
4     111
5      87
13     76
19     68
8      68
7      62
11     58
6      49
12     48
9      43
10     36
2      16
21     14
20     13
22     11
23      8
25      4
24      3
27      1
31      1
dtype: int64
```

Visualising data in both the columns



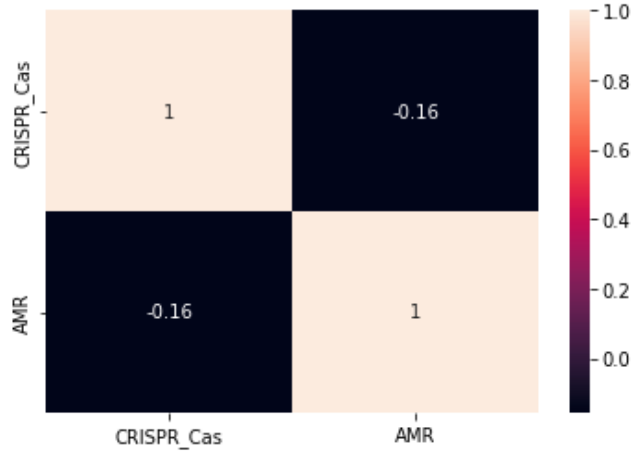


Finding the correlation between the two columns "CRISPR_Cas" and "AMR"

	CRISPR_Cas	AMR
CRISPR_Cas	1.000000	-0.156173
AMR	-0.156173	1.000000

Inference: We get a negative correlation between the two attributes of the dataset

Plotting the correlation using correlation matrix

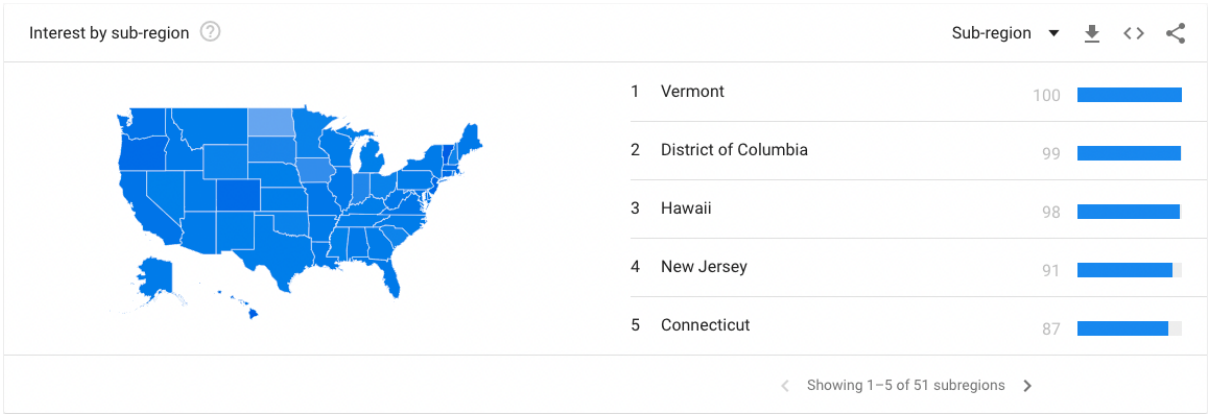


Inference: The visual representation of the correlation matrix

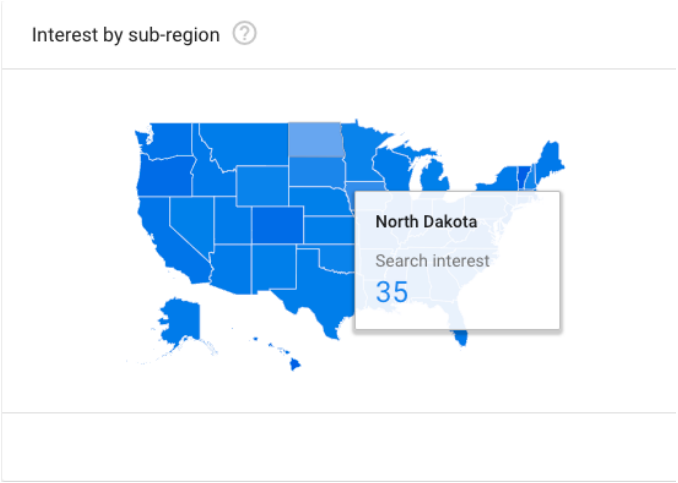
4) Learn about Omicron using Google Trend. Write a brief summary including four highlights of what you have learned.

Google Trends:
Google Trends is a valuable search trends tool that displays the frequency with which a certain search phrase is entered into Google's search engine in relation to the site's overall search traffic over time. Google Trends may be used to compare keyword research and find spikes in keyword search volume caused by events.

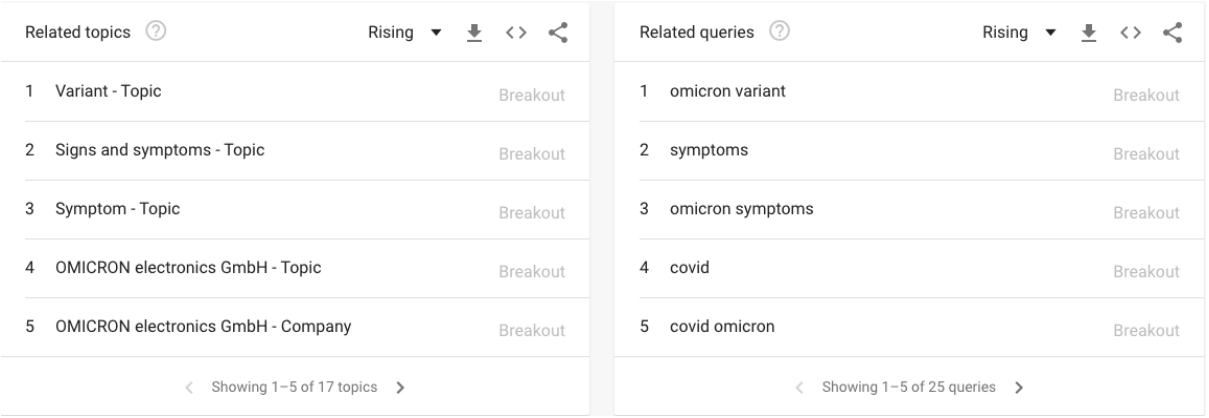
Omicron:
The word “Omicron” had a spike in Google trends due to the recent release of information from South African doctors, who discovered another Covid variant.
Google Trends could be used to see in which location the term “Omicron” was most popular during a specified time frame. Values are calculated on a scale from 0 to 100, where 100 is the location with the most popularity as a fraction of total searches in that location, a value of 50 indicates a location that is half as popular. A value of 0 indicates a location where there was not enough data for this term.



As you can see from the image above, Google trends shows us the popularity of the term Omicron in all of the 51 states in the United States.
We can see that in the state of Vermont the word Omicron was the highest searched keyword in the past 3 months.
On the other hand, New York has shown a score of 73 search interests which is very less compared to Vermont, This might indicate how concerned people are about Omicron in New York as opposed to Vermont.



North Dakota has had less the 35 search interests for Omicron, which can be a factor to it's negligence in handling the Omicron situation as the cases have spiked tremendously in the state of North Dakota



Google Trends also shows you related search topics and related search queries to our main keyword "Omicron" .
People also searched for omicron variant, symptoms, omicron symptoms, covid etc.
Also some irrelevant topics like OMICRON electronics GmbH were also searched.

5) Write a summary for this paper: COVID-19 or Flu? Discriminative Knowledge Discovery of COVID-19 Symptoms from Google Trends Data.

- The length of your summary is about one page.
- State main ideas: problem that the paper tries to address, what data was used,and what was the method that was applied/developed to solve the problem.
- Add your personal opinion. Do you like the paper or not? Why? How do you
- think about the paper?
- Write the summary in your own words; don't copy and paste.

Problem Statement:

To compare the two datasets and identify unique information for Covid 19 symptoms, the researchers recommend using nonnegative discriminative analysis (DNA). The study is backed up by numerical data that shows ageusia, shortness of breath, and anosmia are three distinct Covid 19 symptoms when compared to flu.

The study uses Google Trends data to examine data from two time periods: one in which both Covid 19 and flu are widespread, and another in which just flu is prominent.

Proposed Method:

Data Used**:**

· Background dataset which had information of FLU in years (2018-2019)

· Target dataset which has both COVID and FLU (2020)

The authors of the study employed and contrasted four methods to discover Covid 19's flu-specific symptoms. Nonnegative matrix factorization (NNMF), discriminative principal component analysis (dPCA), contrustive principal component analysis (cPCA), and nonnegative discriminative analysis are some of them (DNA).

Discriminative PCA seeks a projection matrix that maximizes the ratio of projected target data variance over background data variance, whereas cPCA maximizes the difference between target data variance and background data variance, which would aid in the explicit identification of Flu and Covid 19 symptoms.

The sign ambiguity of utilizing dPCA directly to reveal such a symptom is a difficulty, and as a result, it would not be able to cover the discriminative symptoms of both illnesses. To address this issue, the authors proposed using nonnegative matrix factorization to do nonnegative discriminative analysis on DNA (NNMF).

The Algorithm is as follows:

Algorithm 1: DNA.

1: **Input:** Nonzero-mean target and background data $\{\mathbf{x}_i\}_{i=1}^{m}$ and $\{\mathbf{y}_i\}_{i=1}^{n}$; number of dimensions d.
2: **Construct** covariance matrices of $\{\mathbf{x}_i\}$ and $\{\mathbf{y}_i\}$ to obtain $\mathbf{C}_x$ and $\mathbf{C}_y$.
3: **Perform** nonnegative matrix decomposition on $\mathbf{C}_y^{-1}\mathbf{C}_x$ to obtain the two factorization components $\mathbf{W}$ and $\mathbf{H}$.
4: **Output:** $\mathbf{W}$ and $\mathbf{H}$.

Experimental Evaluation:
The authors ran a series of studies using Google Trends symptom searches from 2018 and 2019 as background data and searches from 2020 as target data.

Further investigation revealed that PCA, cPCA, and NNMF were unable to distinguish unique COVID-19 symptoms, however DNA was successful in identifying discriminative symptoms.

Conclusion:
When discriminative data is available, such as in the Google Trends data from 2018 and 2019, this research shows the unique usage of the Discriminative Principal Component, which aids in the detection of COVID and flu symptoms.

As per my opinion, the paper is not detailed and does not explains the process of the DNA analysis.