Vashita Darji

Case Study 1: Visualization of LinkedIn Data statistically
Case Study 2: Extraction of LinkedIn Data

“Visualization of LinkedIn Data statistically”

Vashita Darji [1]
[1]Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad

Abstract: In this growing world of technology, LinkedIn is used extensively in growing one own’s professional network. People use it as a means to connect with the professionals. It provides the opportunity of the extensive growth by building networks and connections. In this study I have tried to visualize the network of LinkedIn profile using the graph. The graph representation will help the user to view the entire network at a glance and provide statistical analysis.

Keywords: Social media network analysis, LinkedIn visualization, statistical, graph

1 Introduction

In this era where people use LinkedIn as the platform to connect professionally, build networks, to find their career trajectory, search for the opportunities, visualizing your LinkedIn data can be an insightful way. Many people around the world uses LinkedIn as the way to keep them updated about the new technologies, innovations taking place in their field of interest. It is also used as the medium to connect and collaborate with the people of your similar interest. It helps the one to boost their own skills by learning from others. It also serves the purpose of providing a detailed guided roadmap of the career of the interest as one can find many blogs, articles and other useful information required to start one's journey in the particular field.

While LinkedIn helps us in many ways to provide us with helpful information, it gets difficult to get the required data in one frame. One needs to do the proper research in mutual connections to find the exact information if required. So, it’s hard to visualize the entire network of your connections in one frame.

Thus, graphs can solve this problem and provide us with the visualization of the entire network. In this study I have tried to solve this problem using graph. It is the model prepared in which one could visualize his/her network and explore the different companies their connections work at and what positions they hold.

A Research Contributions

- The paper proposes an approach to represent the LinkedIn data statistically using an appropriate algorithm.
- It will help the general audience to view the entire data at a glance with the statistics.

B Organization

The rest of the paper is organized as follows: Section II presents a detailed discussion on state-of-the-art approaches along with the comparative analysis of existing works. Section III discusses the system model and problem formulation. Then, Section IV describes the proposed approach in detail. Next, experimental results are discussed in Section V. At last, the paper is concluded in Section VI.

2 Background / Related Work

Table 1. Table captions should be placed above the tables.

Proposed Approach	Year	Short Description	Advantages	Limitations
A Malathi, D Radha	2016	Analysis and visualization of Social Media Network	Providing insight about representing any network as graph	Complex graph structure
Puneet Garg, Rinkle Rani, Sumit Miglani	2015	Minning Professional's data from LinkedIn	Classifying data using clustering techniques	
Meredith M. Skeels, Jonathon Grudin	2009	Work place use of social media	Provide statistical information about user of different social media	Needs to be more precise

3 System model and Problem Formulation

A System Model

1. Data Import:
   Here in this paper the required data is imported as follow:
   Dataset: Downloaded through the option provided by LinkedIn
   Tools: Libraries like pandas, janitor, network imported through python in google colab platform.

2. Data Loading: The data is loaded into data-frame named df_ori and the name of data file is “Connections.csv”.

3. Data Cleaning: In this step we have removed the rows with the missing values if the missing values are for First, Last Name and Position.

4. Data Analysis: Here we have displayed horizontal bar graph with 10 companies which has the most connections.

5. Categorizing Data: For this we have maintained 2 columns, for which First column have name of Companies and second keeps the count of the number of connections. We have stored this data into data-frame named df_company.

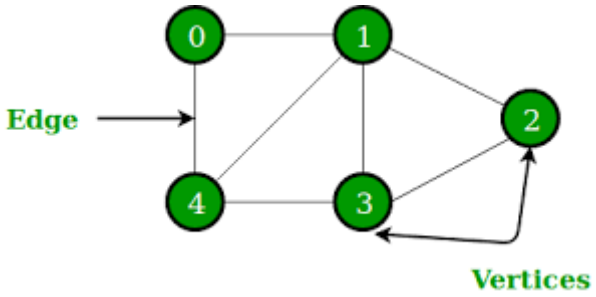
6. Data Reduction: For the data reduction, we have stored the data into new data frame named df_reduced_company. Give the data of df_company we dropped the companies which had count less that 1. This will help to reduce our data.

7. Graph Creation: We have created graph using the library Networkx which can plot graph. Here using the dataset df_reduced_company, we created nodes which will represent the company. Size of the node will be determined by the count frequency. At center there is root user. Edges display the relation between the user and the companies in his/her connection.
8. Display: The graph is saved as an HTML file ("company_graph.html") and displayed using the `display(HTML())` function.

B Problem Formulation

While huge amount of useful data is available on the LinkedIn profile, it possesses challenge to view the entire data in one frame and identify patterns in the organizations and connections of our connections. This model will help the user to view the entire data in one frame with the help of graph and tree representation.

The graph can be represented here as an undirected graph:



G = {V, E}
V = set of vertices
E = set of edges

For our study set of vertices would be companies:

V = {Nirma University, Google, Microsoft}

And the edge between the root and the vertex will represent the relations between them.

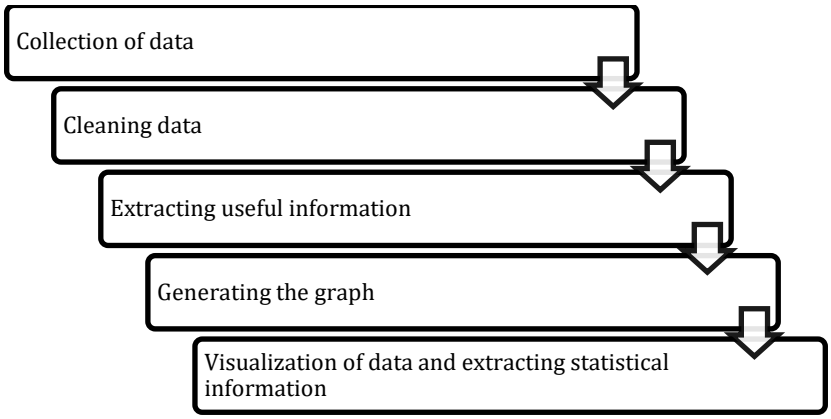
E = {e1, e2, e3}

E1 = edge between root and Nirma university. (for e.g.)

The size of the vertices will be determined by the count of connections.

4 Proposed Methodology

1. Methodology:



Graph Representation Algorithm used:

1. Initialize the empty graph object.
2. Initialize yourself as central root.
3. Now iterate through the reduced data frame.
4. Store the company name and count.
5. Add new node of the company, decide its size accordingly and give the title to the node.
6. Add edge from the root node to the new node.
7. Repeat step 4,5,6 for all the companies.
8. Now generate the graph by providing it's root node.

2. Rationale: In this process we have taken data available and processed it i.e. working with missing values. We have later analyzed it and presented it in network graph. Thus, making it easier to visualize the connections between different companies and their positions and interpret it.

3. Data and Tools:

Data: The primary dataset is a CSV file containing information about LinkedIn connections.

Tools: Python, pandas, Janitor, Pyvis, NetworkX

4. Procedure:
    a. Importing Libraries: The data is loaded into data-frame named df_ori and the name of data file is "Connections.csv".

b. Data Loading: The script loads data from a CSV file named "Connections.csv" into a Pandas DataFrame called `df_ori` using the `pd.read_csv` function. This step prepares the raw data for analysis.

c. Data Processing: In this step we have removed the rows with the missing values if the missing values are for First, Last Name and Position.

d. Data Analysis and Visualization: Here we have displayed horizontal bar graph with 10 companies which has the most connections.

e. Data Categorization: For this we have maintained 2 columns, for which First column have name of Companies and second keeps the count of the number of connections. We have stored this data into data-frame named df_company.

f. Data Reduction: For the data reduction, we have stored the data into new data frame named df_reduced_company. Give the data of df_company we dropped the companies which had count less that 1. This will help to reduce our data.

g. Graph Creation: We have created graph using the library Networkx which can plot graph. Here using the dataset df_reduced_company, we created nodes which will represent the company. Size of the node will be determined by the count frequency. At center there is root user. Edges display the relation between the user and the companies in his/her connection.

h. Display: The graph is saved as an HTML file ("company_graph.html") and displayed using the `display(HTML())` function.

Expected Outcomes:

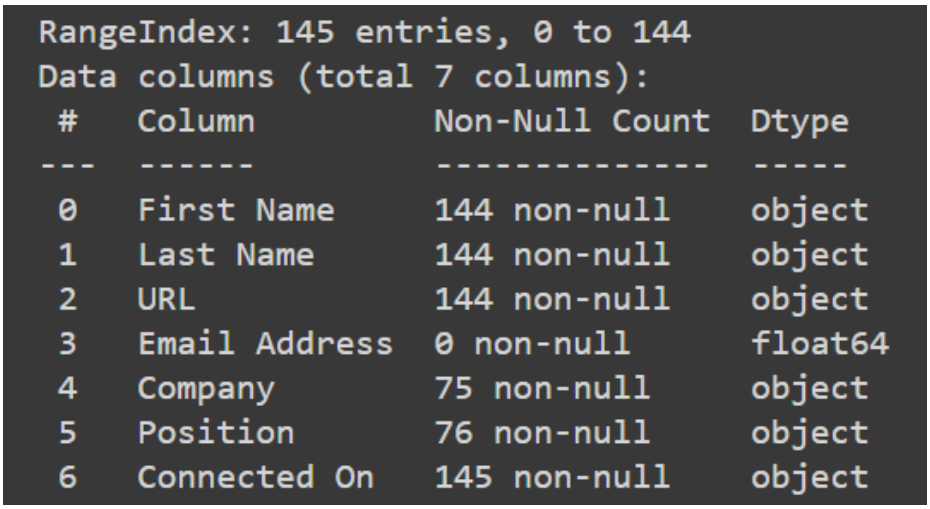
1. Graph Representation:
   Creation of graph that represents connections between companies and the root user. The graph nodes represent the companies and the edges represent the relation between the root user's connections and the companies.

2. Graph Interaction:
   It will develop an interactive graph, which can be zoomed in and zoomed out providing user interaction.

3. Data Presentation:
   The final network graph "company_graph.html" will help to view, interact and explore the connections through graph directly.

Overall, this focuses on the representation and visualization of the data and connections at a glance.

C Figures and Tables



```
RangeIndex: 145 entries, 0 to 144
Data columns (total 7 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   First Name     144 non-null    object
 1   Last Name      144 non-null    object
 2   URL            144 non-null    object
 3   Email Address  0 non-null      float64
 4   Company        75 non-null     object
 5   Position       76 non-null     object
 6   Connected On   145 non-null    object
```

Fig. 1. Dataset

	First Name	Last Name	URL	Email Address	Company	Position	Connected On
0	Rahul	Morabiya	https://www.linkedin.com/in/rahul-morabiya-587...	NaN	NaN	NaN	16-Oct-23
1	Ayush	Patel	https://www.linkedin.com/in/ayush-patel-78133a251	NaN	NaN	NaN	08-Oct-23
2	Shreyansh	Patel	https://www.linkedin.com/in/shreyansh-patel-5b...	NaN	NaN	NaN	07-Oct-23
3	Abhishek	Jani	https://www.linkedin.com/in/abhishek-jani-41ba...	NaN	NaN	NaN	07-Oct-23
4	Shreya	Agrawal	https://www.linkedin.com/in/shreya-agrawal-ba7...	NaN	Nirma University, Ahmedabad, Gujarat, India	Undergraduate Student	07-Oct-23

Fig. 2. Before dropping the columns

	url	company	position	connected_on
4	https://www.linkedin.com/in/shreya-agrawal-ba7...	Nirma University, Ahmedabad, Gujarat, India	Undergraduate Student	07-Oct-23
6	https://www.linkedin.com/in/utsav-thakkar	Goldman Sachs	Software Engineer	26-Sep-23
9	https://www.linkedin.com/in/sahil-singh-a405a4226	IEEE Student Branch DA-IICT	Executive Committee Member	24-Sep-23
11	https://www.linkedin.com/in/maulik-bhavnagariy...	Pixer Digital	Web Developer	14-Sep-23
12	https://www.linkedin.com/in/viraj-panchal-b28b...	Dhirubhai Ambani Institute of Information and ...	Teaching Assistant	09-Sep-23

Fig. 3. After cleaning the data (dropping the columns)

D Code snippets

```
#importing libraries required
!pip install pyjanitor pyvis --quiet
import pandas as pd
import janitor
import datetime
#importing required model
from IPython.core.display import display, HTML
from pyvis import network as net
import networkx as nx
import warnings
warnings.filterwarnings("ignore",
category=DeprecationWarning)
#reading the file and printing it's
information
```

```
df_ori = pd.read_csv("Connections.csv")
df_ori.info()
#cleaning the data
df = (df_ori.clean_names().drop(columns=['first_name',
'last_name', 'email_address']).dropna(subset=['company',
'position']))
df.head()
#plotting bar-graph
df['company'].value_counts().head(10).plot(kind="barh").inv
ert_yaxis();
#categorizing the data based on companies and its count
df_company = df['company'].value_counts().reset_index()
df_company.columns = ['company', 'count']
df_company = df_company.sort_values(by="count",
ascending=False)
df_company.head(10)
#reducing the size of the data-set
print(df_company.shape)
df_company_reduced = df_company.loc[df_company['count']>=2]
print(df_company_reduced.shape)
#creating graph
# initialize graph
g = nx.Graph()
g.add_node('root') # intialize yourself as central node

#iterate through the data frame
for _, row in df_company_reduced.iterrows():

  # store company name and count
  company = row['company']
  count = row['count']

  title = f"<b>{company}</b> - {count}"
  positions = set([x for x in df[company ==
df['company']]['position']])
  positions = ''.join('<li>{}</li>'.format(x) for x in
positions)
```

```
    position_list = f"<ul>{positions}</ul>"
    hover_info = title + position_list

    g.add_node(company, size=count*4, title=hover_info)
    g.add_edge('root', company)

# generate the graph
nt = net.Network(notebook=True, cdn_resources='in_line')
nt.from_nx(g)
nt.show('company_graph.html')
display(HTML('company_graph.html'))
```

5 EXPERIMENTAL RESULTS AND DISCUSSION

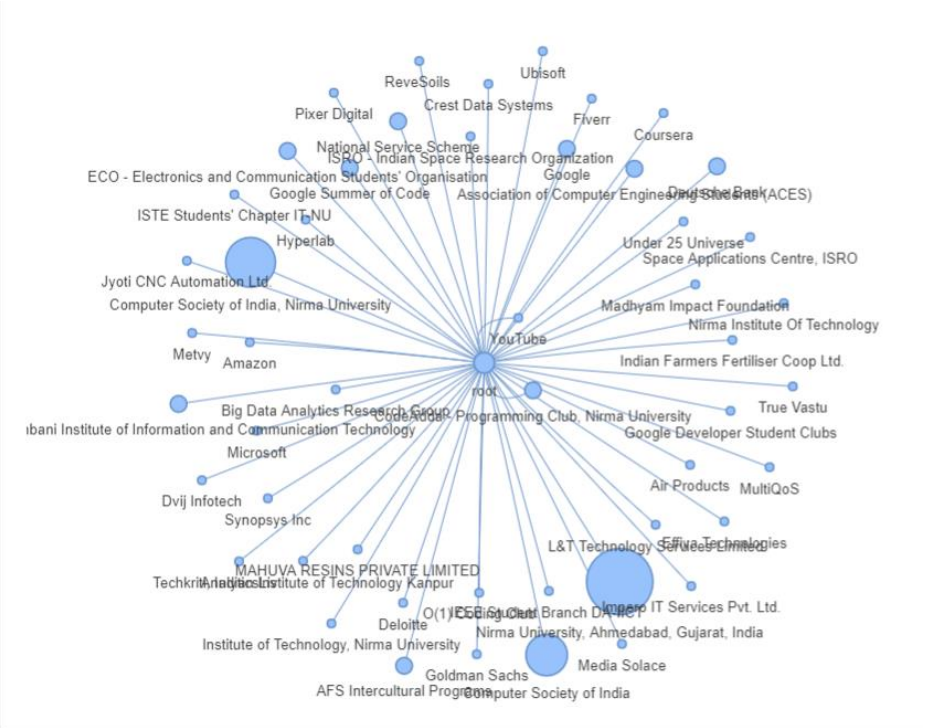


Fig. 1. Visualization of the entire network

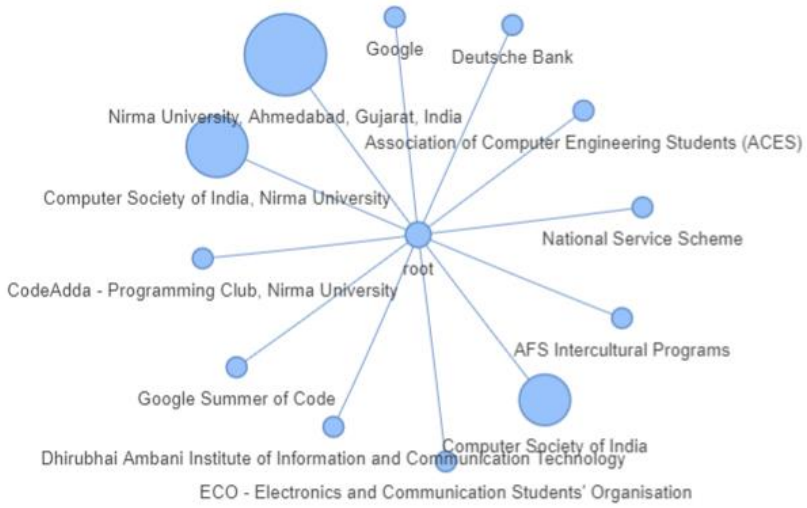


Fig. 2. Visualization of the reduced network

Here, by introducing this model to our dataset, we were able to visualize all the small and big organizations that our networks work at. It can help to explore the organization and help the one in their future aspect according to their interest.

6 Conclusion

In this study we have analyzed the LinkedIn data successfully and we can divide into following major components:

Top Companies: We have identified top 10 companies where our connections work at which will help us to explore our future trajectories and provide us with more options.

Data Preprocessing: We have performed data cleaning by removing rows with missing information which can also be used in other analysis

Data Reduction: To have idea about the major companies where the connections work at, we used the idea of df_company_reduced data and kept only those companies whose count was greater or equal to 2 which will help in identifying major companies.

Interactive Network Graph: Network graph provides the clear idea about the connections at a glance and also visually appealing.

In conclusion, this study provides us with valuable insights of our network, which can help us to view the data at a glance. It will help to identify where the connections work at, provide with name of them at a glance which can help the user in searching jobs, research area or any such particular requirement.

References

[1] A Malathi, D Radha.: Analysis and visualization of Social Media Network (2016).
[1] Puneet Garg, Rinkle Rani, Sumit Miglani.: Minning Professional's data from LinkedIn International Conference on Advances in Computing and Communications (ICACC) (2015).
[1] Meredith M. Skeels, Jonathon Grudin.: When Social Networks Cross Boundaries: A Case Study of Workplace Use of Facebook and LinkedIn ,2009 ACM International Conference on Supporting group work (2009).

“Extraction of LinkedIn Data”

Vashita Darji [1]
[1]Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad

Abstract: In this era where social media is used intensely to build connections and grow within one’s network, LinkedIn is used extensively in growing one own’s professional network. It is used as a means to connect with the professionals. It provides the opportunity of the extensive growth by building networks and connections. In this study we have divided the dataset of the user of the LinkedIn profile based on their positions. This can help us to categorize the huge data and divide it into subgroups which can provide meaning insights. By utilizing techniques of TF-IDF vectorization and K-Means clustering in this study we will provide a way to categorize the data and highlight the career-related pattern.

Keywords: Extraction, treemap, k-means clustering, TF-IDF, vectorization, LinkedIn

1 Introduction

LinkedIn is the world’s largest network which helps its users to connect professionally. Here in this study using vectorization and clustering technique we have tried to divide the data based on the positions of the users. This will help to analyze and visualize the data and help us to categorize.

We have preprocessed the data and transformed the textual job position descriptions into numerical vectors using Term Frequency-Inverse Document Frequency (TF-IDF) vectorization. The K-Means clustering algorithm is used to divide LinkedIn connections into clusters based on the TF-IDF vectors. We have also listed the name of the users along with the clusters which will give us the clear idea.

Along with the processing and clustering of the data, we have also used treemap to visualize the data. In this treemap we have illustrated the distribution of LinkedIn connections across different job positions and the relationships between clusters. The treemap's coloring is based on the job position descriptions and the size of the rectangle assigned to the nodes of treemap depends on the number of employees.

Thus, in this study we have tried to visualize the data using vectorization and clustering techniques which will help the user to visualize the pattern in their connections.

A Research Contributions

- This study helps in vectorization of text and clustering techniques to analyze LinkedIn user’s data based on job position descriptions.

- This study provides insights into the clustering of LinkedIn user profiles, identifying patterns among job descriptions within one's professional network.
- It will help the general audience to view the entire data at a glance with the statistics.

B Organization

The rest of the paper is organized as follows: Section II presents a detailed discussion on state-of-the-art approaches along with the comparative analysis of existing works. Section III discusses the system model and problem formulation. Then, Section IV describes the proposed approach in detail. Next, experimental results are discussed in Section V. At last, the paper is concluded in Section VI.

2 Related Work

Table 1.

Proposed Approach	**Year**	**Short Description**	**Advantages**	**Limitations**
A Malathi, D Radha	2016	Analysis and visualization of Social Media Network	Providing insight about representing any network as graph	Complex graph structure
Puneet Garg, Rinkle Rani, Sumit Miglani	2015	Minning Professional's data from LinkedIn	Classifying data using clustering techniques	
Danny Bradhury	2011	Mining LinkedIn Data	Provides way how LinkedIn data can be used	Narrower interpretation

3 System model and Problem Formulation

A System Model

1. Data Source (LinkedIn Connections Data): The data collected is the data of the user which can be downloaded through the option provided by LinkedIn on their app through request.

2. Data Preprocessing:

   This component involves:

   - Data Loading: We have read the LinkedIn connections data from a CSV file names “Connections.csv” into data-frame named df.

   - Data Cleaning: We have discarded the rows for which the value of ‘Position’ column is missing

3. Feature Extraction (TF-IDF Vectorization): We have converted the textual job position descriptions into numerical vectors that is in the matric form using the Term Frequency-Inverse Document Frequency (TF-IDF) technique.

4. Clustering (K-Means): Using K-Means clustering we have grouped the LinkedIn connections into clusters based on their TF-IDF vectors. For simplicity we have set number of clusters to be 5.

5. Cluster Analysis: Here we have analyzed the clusters:

   - Top Terms: It will identify and prints the top terms associated with each cluster.

   - Cluster Members: It will provide a list of the first names of LinkedIn connections in each cluster.

6. Data Visualization (Treemap): We have used Plotly Express to plot the treemap. Here the size of the node will be determined the count of the position. The treemap is dynamic and will allow the user to interact and will also show the first names of the users who hold that position and the count.

B Problem Formulation

While huge amount of data is available on the LinkedIn, analyzing and data at a glance will help us to identify the patterns and have a more clear understanding about our network and patterns. The problem can be formulated as below.

Here for K-means Clustering: The distance measure we have used default Euclidean distance measure to find out the similarity.
Let us assume two points, such as (x_1, y_1) and (x_2, y_2) in the two-dimensional coordinate plane. The Euclidean distance formula is given by:

$$\mathbf{d = \sqrt{[(x_2 - x_1)^2 + (y_2 - y_1)^2]}}$$

Where,

"d" is the Euclidean distance

(x_1, y_1) is the coordinate of the first point

(x_2, y_2) is the coordinate of the second point.

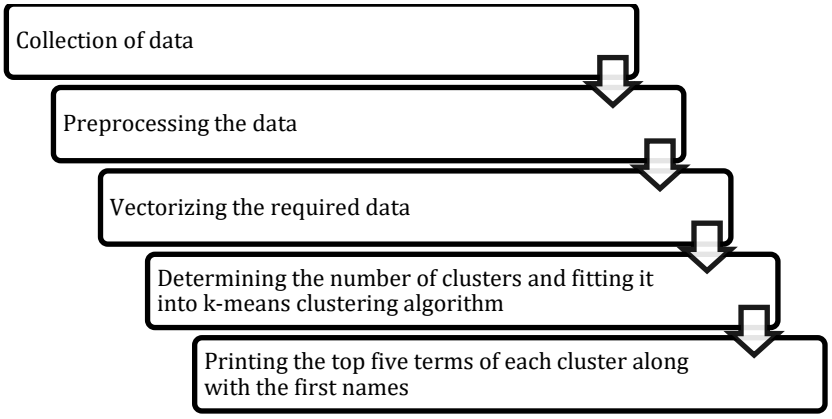
For tree, each node represents the position:

Tree node : size determined by number of count of the position.

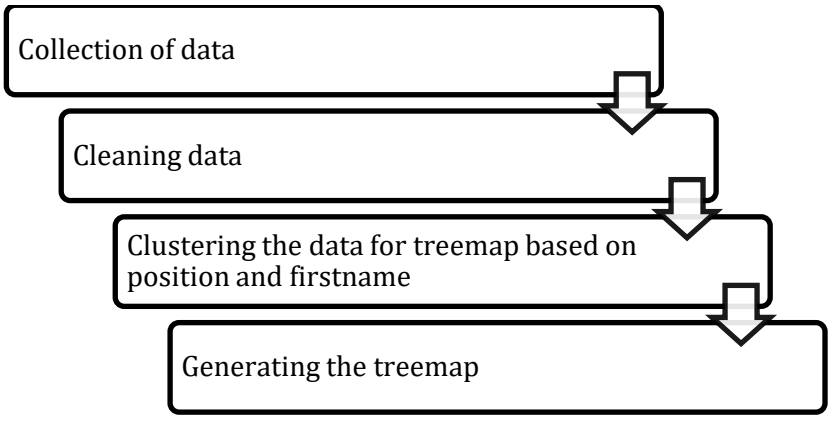
4 Proposed Methodology

1. Methodology:

K-means Clustering:



Generating a treemap:



5. Rationale: In this process we have taken data available and processed it i.e. working with missing values. We have later analyzed it and presented it in network graph. Thus, making it easier to visualize the connections between different companies and their positions and interpret it.

2. Data and Tools:
Data: The primary dataset is a CSV file containing information about LinkedIn connections.
Tools: Python, pandas, scikit learn: Tfidf vectorizer, k-means, plotly express

3. Procedure:

- Data Preprocessing:
  This component involves:
  - Data Loading: **We have** read the LinkedIn connections data from a CSV file names "Connections.csv" into data-frame named df**.**
  - Data Cleaning: We have discarded the rows for which the value of 'Position' column is missing

- Feature Extraction (TF-IDF Vectorization): We have converted the textual job position descriptions into numerical vectors that is in the matric form using the Term Frequency-Inverse Document Frequency (TF-IDF) technique.

- Clustering (K-Means): Using K-Means clustering we have grouped the LinkedIn connections into clusters based on their TF-IDF vectors. For simplicity we have set number of clusters to be 5.

- Cluster Analysis: Here we have analyzed the clusters:

    - Top Terms: It will identify and prints the top terms associated with each cluster.

    - Cluster Members: It will provide a list of the first names of LinkedIn connections in each cluster.

- Data Visualization (Treemap): We have used Plotly Express to plot the treemap. Here the size of the node will be determined the count of the position. The treemap is dynamic and will allow the user to interact and will also show the first names of the users who hold that position and the count.

4. Expected Outcomes:

- Cluster Analysis: We have divided the clusters which has the values of position. Among which we have identifies the top terms.
    - Top Terms: We have identified the top terms from each cluster based on positions.
    - Cluster Members: The first names associated with each cluster top term

- Data Visualization: A treemap that visualizes structure of job positions within clusters. Color of nodes is based on job position descriptions. Sizes of nodes is according to count of LinkedIn connections for each position.

C Figures and Tables

```
RangeIndex: 145 entries, 0 to 144
Data columns (total 7 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   First Name      144 non-null    object
 1   Last Name       144 non-null    object
 2   URL             144 non-null    object
 3   Email Address   0 non-null      float64
 4   Company         75 non-null     object
 5   Position        76 non-null     object
 6   Connected On    145 non-null    object
```

Fig. 1. Dataset

D Code snippets

```python
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.cluster import KMeans

# Load the dataset
df = pd.read_csv('Connections.csv')

# Convert the 'Position' column to numeric vectors using TF-IDF
df = df[df['Position'].notna()]
vectorizer = TfidfVectorizer(stop_words='english')
X = vectorizer.fit_transform(df['Position'].values.astype('U'))

# Determine number of clusters
n_clusters = 5
kmeans = KMeans(n_clusters=n_clusters)
df['Cluster'] = kmeans.fit_predict(X)

# Get order of centroids and terms
order_centroids = kmeans.cluster_centers_.argsort()[:, ::-1]
terms = vectorizer.get_feature_names_out()
```

```
# Display results
for i in range(n_clusters):
    print(f"Cluster {i}:")
    for ind in order_centroids[i, :5]:  # printing top 5 terms for each cluster
        print(f" {terms[ind]}")
    print(df[df['Cluster'] == i]['First Name'].tolist())
    print()
import pandas as pd
import plotly.express as px

# Aggregate data for the treemap
cluster_sizes = df.groupby('Position').size().reset_index(name='Counts')
cluster_positions = df.groupby('Position')['First Name'].apply(lambda x: ', '.join(x)).reset_index()

agg_df = pd.merge(cluster_sizes, cluster_positions, on='Position')
```

```
# Create the treemap
fig = px.treemap(agg_df,
                 path=['Position', 'First Name'],
                 values='Counts',
                 title="LinkedIn Positions Clustering Treemap",
                 color='Position',
                 color_continuous_scale='RdBu'
                )

fig.show()
```

5 EXPERIMENTAL RESULTS AND DISCUSSION

```
Cluster 0:
 engineer
 software
 president
 senior
 vice
['Utsav', 'Viraj', 'Nabhag', 'Henil', 'Sadia', 'Haridev', 'Pimal', 'Richa', 'Heli', 'Jatin', 'Rajesh', 'Darshil', 'Arunima', 'Riddhi', 'Kavish', 'M

Cluster 1:
 student
 coordinator
 undergraduate
 placement
 committe
['Shreya', 'Gaurav', 'Juhi', 'Aditya', 'Yaksh', 'Devansh', 'Raja']

Cluster 2:
 member
 committee
 executive
 year
 freelance
['Sahil', 'Priya', 'Hitesh', 'Pankti', 'Vidhi', 'Kavan', 'Harvy', 'Krishi']

Cluster 3:
 intern
    research
    project
    development
    js
  ['Keyuri', 'Priyanshu', 'Heni', 'Saral ', 'Priyanshu', 'Moin']

  Cluster 4:
   developer
   web
   stack
   end
   javascript
  ['Maulik', 'Vaibhav', 'Ashray', 'Pratik', 'Bhaumil', 'Meet']
```

Fig.1.1 Result of the clustering

Cluster	Top terms	First Name
0	Engineer, software, president, senior, vice	'Utsav', 'Viraj', 'Nabhag', etc
1	student, coordinator, undergraduate, placement, committee	Shreya', 'Gaurav', 'Juhi',etc
2	member, committee, executive, year, freelance	'Sahil', 'Priya', 'Hitesh', etc
3	intern, research, project, development, js	'Keyuri', 'Priyanshu', 'Heni', etc
4	developer, web, stack, end, javascript	'Maulik', 'Vaibhav', 'Ashray', etc

Fig.2. A treemap visualization is generated, displaying the clustered data in a hierarchical format. Each node in the treemap represents a job position, and the color of nodes is determined by the job position descriptions. The size of nodes corresponds to the number of LinkedIn connections in each position.

6 Conclusion

In this study we were able to categorize the data of the LinkedIn user based on their position titles. This study includes vectorization and clustering techniques which can be useful in the future studies as well to categorize the data.

This provides us the picture at a glance and will help to recognize career patterns of our connections clearly. By this the user can thereafter easily find contacts of his/her interested field. This will serve as a tool to build more effective connections.

References

[1] A Malathi, D Radha.: Analysis and visualization of Social Media Network (2016).
[1] Puneet Garg, Rinkle Rani, Sumit Miglani.: Minning Professional's data from LinkedIn International Conference on Advances in Computing and Communications (ICACC) (2015).
[4] Danny Bradhury.: Data Mining with LinkedIn. In: Science Direct, pp. 5–8. Publisher (2011).