ISSN 2063-5346

MANAGING ENTERPRISE DATA AND PROVIDING A PERSONALIZED EMPLOYEE DATA VISUALIZATION

Nausheen Fatima[1], Potnuru Yashwanth[2], Tene Harishwar Reddy[3], Mohammad Faiz[4*]

Article History: Received: 01.02.2023 Revised: 07.03.2023 Accepted: 10.04.2023

Abstract

This paper explores about the today's data-driven world, organizations need to effectively manage their enterprise data to make informed business decisions. It explores various techniques for managing enterprise data, including data governance, data quality, data integration, and data analytics. And it discusses about the importance of providing personalized employee data visualizations to enhance decision-making and productivity. It explores the benefits of using employee data visualizations and highlights the challenges of implementing such visualizations in an enterprise setting. Overall, the findings of this study can help to give the better recommendations for organizations to successfully manage their enterprise data and provide personalized employee data visualizations that effectively meet their business needs.

Keywords— Data Analytics, Clusterization, Visualization, KNN Algorithm.

[1,2,3,4*]School of Computer Science & Engineering Lovely Professional University, Phagwara, Punjab, India

[1]nausheen.28838@lpu.co.in, [2]ugandhar2412@gmail.com, [3]harishwarreddy199@gmail.com, [4]faiz.techno20@gmail.com

DOI:10.31838/ecb/2023.12.s1-B.191

I. Introduction

Managing enterprise data involves the process of collecting, organizing, storing, and analyzing large amounts of data within an organization. This data can come from various sources such as customer interactions, employee records, financial transactions, and operational processes.The goal of managing enterprise data is to provide insights into business operations and inform decision-making.

One aspect of managing enterprise data is providing personalized employee data visualization. This involves presenting data in a way that is tailored to each individual employee's needs and preferences. For example, an employee might want to see their performance metrics compared to their team's average, or they might want to track their progress towards achieving certain goals. By providing personalized data visualizations, employees can gain a better understanding of their own performanceand make informed decisions about how to improve their work. Additionally, personalized data visualizations can help managers identify areas where employees may need additional support or training

Managing the vast amount of enterprise data, including employee data, financial data, and operational data. without proper management, data silos can arise, leading to inefficiencies, errors, and increased costs. Additionally, providing personalized employee data visualization can be difficult, requiring expertise in data analysis, visualization, and user experience design. Overcoming these challenges is crucial for organizations to improve their data-driven decision-making capabilities and enable their employees to be more productive and effective. Organizations face challenges in managing their vast amount of enterprise data, including employee data, financial data, and operational data. Without proper management, data silos can arise, leading to inefficiencies, errors, and increased costs. Additionally, providing personalized employee data visualization can be difficult, requiring expertise in data analysis, visualization, and user experience design. Overcoming these challenges is crucial for organizations to improve their data-driven decision-making capabilities and enable their employees to be more productive and effective.

The generic analytics dashboard framework that can work across multiple analytics systems, making it easier to analyze data from different sources. This framework allows for the visualization of data stored internally by the application, along with data fetched from sources such as Google Analytics. By configuring the required fields in a JSON format file, the dashboard can fetch and visualize the data in corresponding graphs. This framework can be useful for managing enterprise data and providing personalized employee data visualization [6] [14][15].

This work addresses the challenge of managing and extracting knowledge from large amounts of data generated in different sectors. The proposed solution, VIM, is a generic data analytics system that provides personalized data visualization, data pre-processing, and knowledge extraction from various data sources. The system offers features such as visualization of data, pattern mining, and drift analysis. The paper presents the architecture and system description of VIM, along with a developed data analytics web application tool. The contributions of this work are building a scalable and user-friendly system that can be extended to other datasets [3] [16][17].

II. Literature Survey

In [1] author proposes a solution in developing scalable DBMS for update-heavy and analytical workloads in cloud infrastructure, analyzes successful design choices and outlines principles for the next generation of cloud-bound DBMS. It also addresses the design space for supporting large single and multitenant systems, and open research challenges in cloud data management.

In [2] author proposed Pyramid Viz is a tool for finding patterns in data. It shows the patterns as a pyramid of blocks, with colours indicating how often they occur. The blocks are connected by lines to show which items go together. It's easy to use and good for finding patterns in big data. Ongoing work includes more testing and user studies [18].

In [3] author proposed a web-based tool called VIM for visualizing, pre-processing, and mining data for knowledge extraction. It can generate association rules, analyze data drift, and extract information from datasets. VIM is developed using Python Django framework and GraphLab library and is publicly usable. Future work includes improving its capabilities to eliminate noise and extract important features automatically. Overall, VIM is a comprehensive tool for data mining and big data analytics for research and knowledge-based systems [19].

In [4] author proposed a tool, cpmViz, visualizes radioactive contamination data from mobile and static sensors over time. It allows interactive exploration in both spatial and temporal dimensions using linked views. We introduce new uncertainty measures for spatial and temporal aspects of the data and successfully identify major events and areas that require more static sensors. Overall, cpmViz makes exploring the complex dataset easy and helps officials determine the severity of contamination at the city's nuclear power plant [20].

In [5] author proposed eVADE, a tool that uses machine learning and visualization techniques to extract meaningful insights from Earth observation data. The tool is demonstrated through a study of deforestation in the Carpathian Mountains in Romania and has the potential to provide valuable insights beyond human perception and visualization [21].

In [6] author presents a dashboard framework that combines data from different analytics sources, using two configuration files for customization. The framework uses a plugin architecture for easy addition of new data sources and is data-driven to automatically adjust to changes in source data. The prototype dashboard solution can work with local or remote data, and future plans include a drag-and-drop UI toolkit and integration with more analytics systems [22].

In [7] author proposed VAMD, a visual analytics system that allows for the analysis of large and heterogeneous datasets in a combined view. It includes a user-friendly interface for uploading data, running machine learning algorithms, and exploring connections within the data, as demonstrated through a disaster response scenario with social media data. The system aims to support multimodal dataset exploration with scalable techniques, filling a gap in merging and visualizing heterogeneous data sources [23].

Table 1: Various existing models with limitations

References	Title	Publication Year	Journal	Methodologies used	Limitations
[5]	An Interactive Visual Analytics Tool for Big Earth Observation Data Content Estimation	2019	IEEE	Data Analytics, Evade tool framework, data Mining algorithms	IT affected by the computational requirements and scalability of the proposed techniques for processing and analyzing large EO datasets
[4]	A Web-Based Visualization Tool for Uncertain Spatiotemporal Data	2017	IEEE	Human centered computing, Visualization, d3.js, PostgreSQL	It's difficult to explore spatiotemporal dataset and these are not user friendly
[6]	A Generic Visualization Framework based on a Data Driven Approach for the Analytics data	2017	IEEE	Open source tools which are Python, R , D3.js , Visualization Components	Biased results it means misleading results and overcomplications, Misinterpretation
[3]	A Big Data Analytics Tool for Data Visualization and Knowledge Mining	2017	IEEE	Visualization features of VIM, Pattern Mining,	Data Quality it impacts the accuracy, Technical Complexity , Data Security and cost is also too high.

III. Proposed Methodology

The proposed methodology for managing enterprise data and providing personalized employee data visualization using the KNN algorithm can be divided into the following steps [24] [25] [26].

1. **Data Collection:** The first step is to collect data from various sources within the organization, such as HR systems, performance evaluation reports, attendance records, etc.This data will be used to build a comprehensive dataset that includes all relevant employee information [8].
2. **Data Cleaning and Pre-processing:** Once the data has been collected, it will need to be cleaned and pre-processed. This involves removing any duplicate records, correcting errors, and standardizing the data so that it can be used in the KNN algorithm.

3. **Feature Engineering:** Feature engineering is the process of selecting the most relevant features from the dataset that will be used to train the KNN model. This involves analyzing the data and selecting the features that are most likely to have a significant impact on the model's accuracy.
4. **KNN Model Training:** After the features have been selected, the next step is to train the KNN model using the dataset. The KNN algorithm is a supervised learning algorithm that requires labeled data to train the model.
5. **Personalized Employee Data Visualization:** Once the KNN model has been trained, it can be used to generate personalized employee data visualizations [9],[10]. This involves inputting a specific employee's data into the model, and the model will then predict which employees are most similar to them based on their data. The model will then generate a visualization that shows the employee's data alongside the data of the most similar employees.
6. **Continuous Model Improvement:** The KNN model will need to be continuously improved to ensure that it remains accurate over time. This involves monitoring the model's performance, analyzing any errors or inaccuracies, and making adjustments to the model as needed.

Overall, this methodology will allow organizations to manage their enterprise data more effectively and provide personalized employee data visualizations that can be used to improve employee performance and engagement [12], [13].

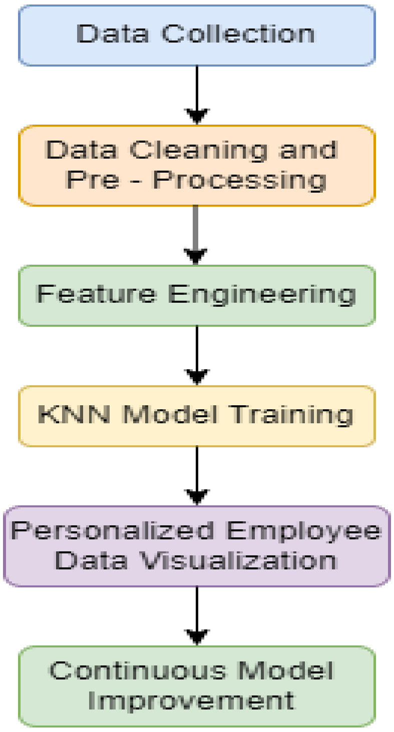


Figure 1: Proposed a new model of managing new data

And Personalized employee data Visualization.

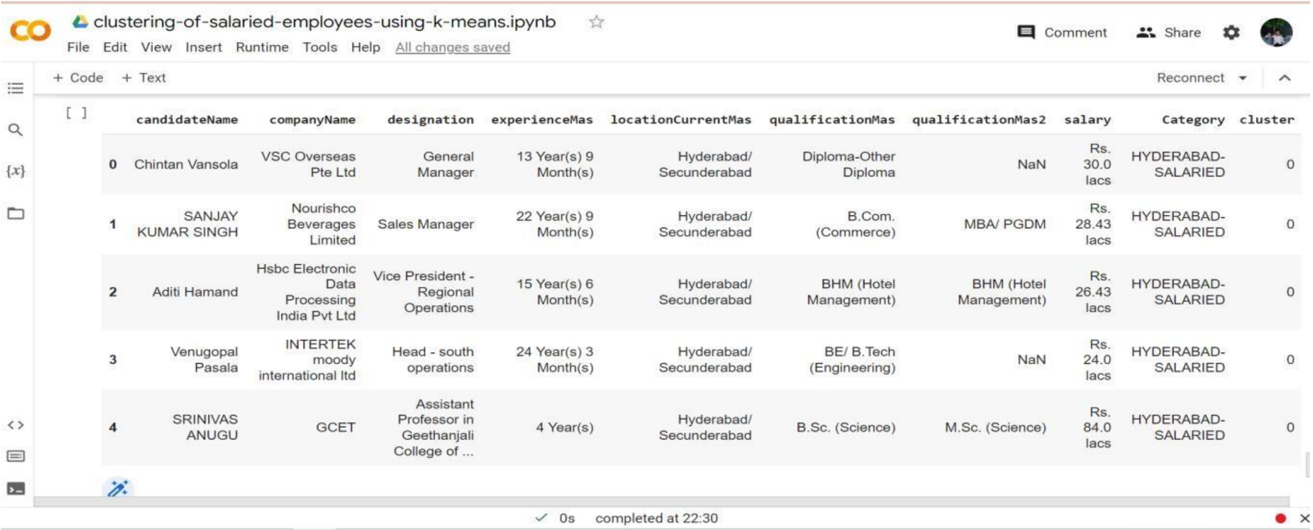
IV. Implementation Details

1. Import the required libraries:
   - numpy
   - pandas
   - warnings
   - missingno
   - matplotlib.pyplot
   - seaborn
   - LabelEncoder and normalize from sklearn.preprocessing
   - scipy.cluster.hierarchy
   - AgglomerativeClustering and KMeans from sklearn.cluster
   - silhouette_score, calinski_harabasz_score,

and davies_bouldin_score from sklearn.metrics

- PCA from sklearn.decomposition

2. Read the data from the CSV file "hyderabad-salaried-employees.csv" into a pandas DataFrame called "df".
3. Display the first few rows of the DataFrame using the **.head()** method.



	candidateName	companyName	designation	experienceMas	locationCurrentMas	qualificationMas	qualificationMas2	salary	Category	cluster
0	Chintan Vansola	VSC Overseas Pte Ltd	General Manager	13 Year(s) 9 Month(s)	Hyderabad/ Secunderabad	Diploma-Other Diploma	NaN	Rs. 30.0 lacs	HYDERABAD-SALARIED	0
1	SANJAY KUMAR SINGH	Nourishco Beverages Limited	Sales Manager	22 Year(s) 9 Month(s)	Hyderabad/ Secunderabad	B.Com. (Commerce)	MBA/ PGDM	Rs. 28.43 lacs	HYDERABAD-SALARIED	0
2	Aditi Hamand	Hsbc Electronic Data Processing India Pvt Ltd	Vice President - Regional Operations	15 Year(s) 6 Month(s)	Hyderabad/ Secunderabad	BHM (Hotel Management)	BHM (Hotel Management)	Rs. 26.43 lacs	HYDERABAD-SALARIED	0
3	Venugopal Pasala	INTERTEK moody international ltd	Head - south operations	24 Year(s) 3 Month(s)	Hyderabad/ Secunderabad	BE/ B.Tech (Engineering)	NaN	Rs. 24.0 lacs	HYDERABAD-SALARIED	0
4	SRINIVAS ANUGU	GCET	Assistant Professor in Geethanjali College of ...	4 Year(s)	Hyderabad/ Secunderabad	B.Sc. (Science)	M.Sc. (Science)	Rs. 84.0 lacs	HYDERABAD-SALARIED	0

Figure 2 : Reading the CSV file

Figure 2 Shows that the code was imported in the necessary Python libraries, read data from a CSV file into a pandas DataFrame called "df", and display the first few rows of the DataFrame using the .head() method. The expected output would be a brief overview of the data in tabular format, showing the first few rows of the DataFrame with each row representing an observation and each column representing a variable.

4. Display information about the DataFrame using the **.info()**method.
5. Display descriptive statistics of the DataFrame using the **.describe()** method.
6. Display the number of missing values in each column of the DataFrame using the **.isna().sum()** method.
7. Display the number of unique values in each column of the DataFrame using the **.nunique()** method.
8. Display the count of values in each unique location in the "locationCurrentMas" column ofthe DataFrame using the **.value_counts()** method.
9. Visualize the distribution of missing values in the DataFrame using the **missingno.matrix()** function, with the DataFrame as its input, and with the following optional parameters:
   - **figsize=(10,4)** to set the size of the figure.
   - **fontsize=10** to set the font size of the text in the figure.

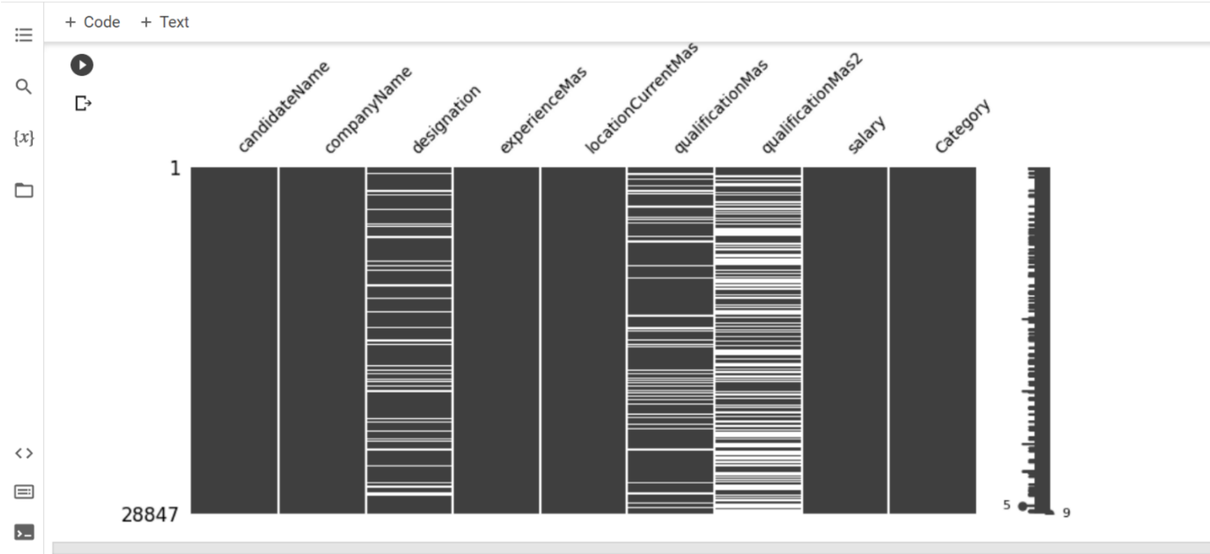


Figure 3 : Visualizing the Missing Matrix Function

Figure 3 shows that the Further code displays information about the DataFrame using the .info() method, including the number of rows and columns, the data types of each column, and the number of non-null values in each column. It also displays the basic descriptive statistics of the DataFrame using the .describe() method, including the count, mean, standard deviation, minimum, and maximum values of each numerical column. Additionally, the code displays the number of missing values in each column of the DataFrame using the .isna().sum() method and the number of unique values in each column using the .nunique() method. The .value_counts() method is used to display the count of values in each unique location in the "locationCurrentMas" column of the DataFrame. Finally, the code visualizes the distribution of missing values in the DataFrame using the missingno.matrix() function, with optional parameters to set the size and font size of the figure. The visualization can be used to identify patterns in the missing data and help inform decisions about how to handle it in the analysis.

10. Remove any rows from the DataFrame where the "companyName", "designation", "candidateName", "Category", or "locationCurrentMas" columns have missing values using the **.dropna()** method with the **axis=0**, **inplace=True**, and **subset=['companyName', 'designation','candidateName','Category','locationCurrentMas']** arguments.
11. Create a deep copy of the DataFrame using the **.copy()** method with the **deep=True** argument and assign it to the variable "final".
12. Replace any missing values in the "qualificationMas" and "qualificationMas2" columns with the string "Missing" using the **.fillna()** method with the **value='Missing'** argument.
13. Visualize the distribution of missing values in the DataFrame using the **missingno.matrix()** function with the DataFrame as its input and with the optional parameters **figsize=(10,4)** and **fontsize=10**.
14. Remove the "candidateName", "Category", and "locationCurrentMas" columns from the DataFrame using the **.drop()** method with the **columns=['candidateName','Category','locationCurrentMas']** argument.

15. Convert the "salary" column from "Rs. X Lakhs" (string) to "X" (float) by:
    - Converting the "salary" column to a string using **.astype('str')**.
    - Removing the "Rs. " and " lacs" strings from each element in the "salary" column using the **.apply()** method with two lambda functions.
    - Converting the "salary" column to a float using **.astype(np.float64)**.
16. Convert the "experienceMas" column from "A Year(s) B Month(s)" (string) to "Month(s)" (float) by defining a function **process_exp()** that takes two arguments: the DataFrame (**dataframe**) and the column index (**colid**) of "experienceMas".
    - Convert the "experienceMas" column to a string using **.astype('str')**.
    - Replace any occurrences of "Fresher" with "0 Year(s) 0 Month(s)" using the **.replace()** method.
    - Loop through each row in the "experienceMas" column, split the value into a list of elements using the **.split()** method, and extract the number of years and months of experience from the list.
    - Multiply the number of years by 12 and add the number of months to get the total number of months of experience.
    - Convert the "experienceMas" column to a float using **.iloc[]** and **.pd.to_numeric()**.
    - Return the modified DataFrame.
17. Use label encoding to convert any categorical columns in the DataFrame to numerical values using a **for** loop, **.select_dtypes(include=['object'])** to select only object data types, and the **LabelEncoder()** class from the **sklearn.preprocessing** module.
18. Normalize the numerical data in the DataFrame using the **normalize()** function from the **sklearn.preprocessing** modulewith the **axis=0** argument and assign the result to the variable "data".
19. Create a heatmap of the correlation matrix of the normalized data using **sns.heatmap()** with the arguments **data.corr()**, **vmin=-1**, **vmax=1**, and **annot=True**.

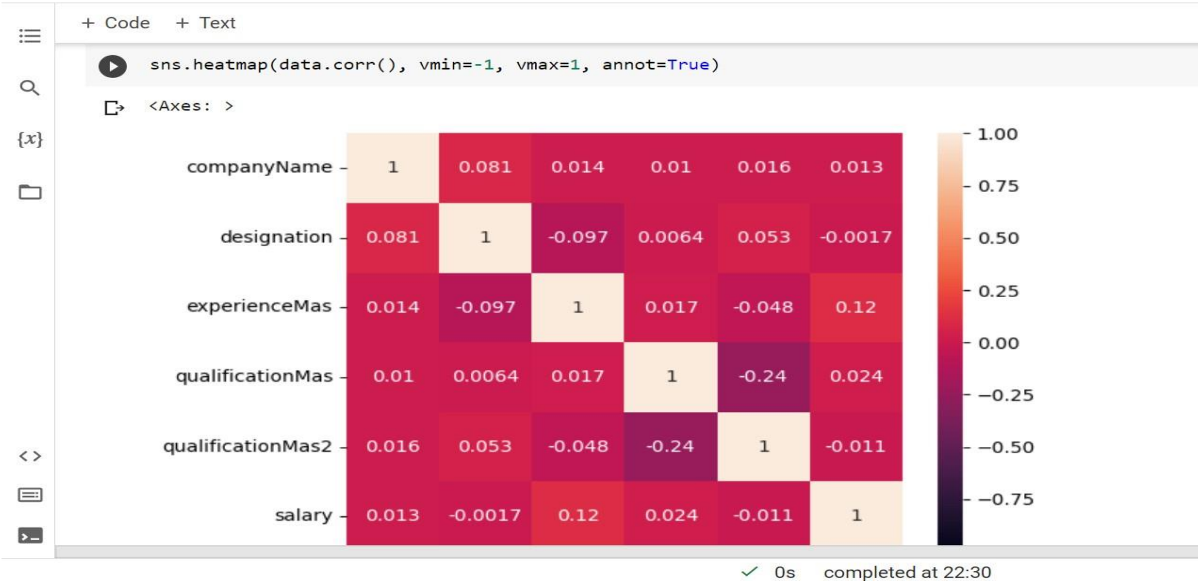


Figure 4 : Visualizing Heat Map Diagram

Figure 4 shows that about further analysis In Step 10, missing rows in certain columns of the DataFrame are removed using the .dropna() method. And It creates a deep copy of the DataFrame for future use. Step 12 replaces missing values in certain columns with the string "Missing" using the .fillna() method. Step 13 visualizes the distribution of missing values in the DataFrame using the missingno.matrix() function. Step 14 removes certain columns from the DataFrame using the .drop() method. Step 15 converts the "salary" column from a string format to a float format using a series of string manipulationand conversion methods. Step 16 converts the "experienceMas" column from a string format to a float format by defining a custom function that extracts the number of months of experience. Step 17 uses label encoding to convert categorical columns to numerical values. And it normalizes the numerical data in the DataFrame using the normalize() function from the sklearn.preprocessing module. Finally, It creates a heatmap of the correlation matrix of the normalized data using the sns.heatmap() function.

20. Initialize PCA transformation with 2 principal components
21. Fit PCA transformation on data
22. Transform data using PCA transformation
23. Create an empty list **Sum_of_squared_distances**
24. Create a range of values for K from 2 to 8 (inclusive)
25. For each value of K in the range: a. Apply KMeans clustering with K clusters to the transformed data b. Fit the KMeans model to the data c. Append the sum of squared distances of samples to their closest cluster center to **Sum_of_squared_distances**
26. Plot the values of K on the x-axis and the sum of squared distances on the y-axis
27. Label the x-axis as "Values of K"
28. Label the y-axis as "Sum of squared distances/Inertia"
29. Set the title of the plot as "Elbow Method For Optimal k"
30. Display the plot.

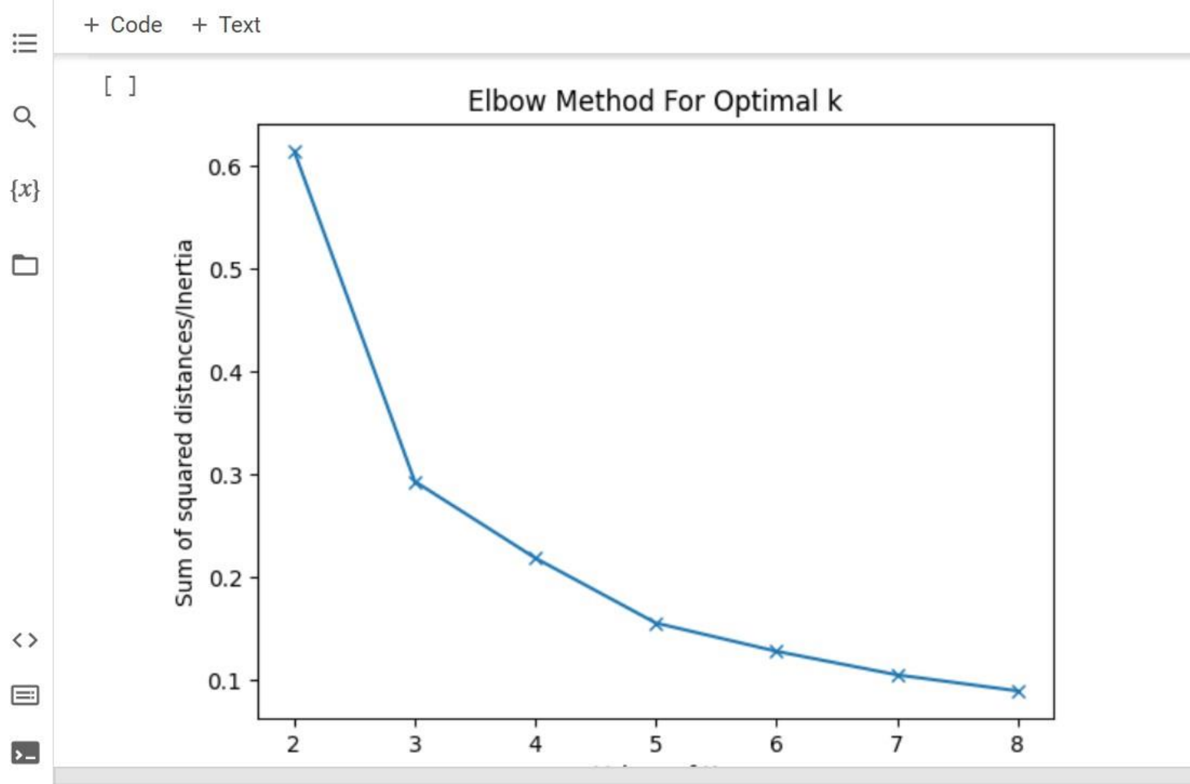


Figure 5 : Elbow Method for Optimal K

Figure 5 shows that it performs the Elbow Method for optimal k in k-means clustering. The first step is to initialize a PCA transformation with 2 principal components and fit it on the data. Then, the data is transformed using the PCA transformation. An empty list called Sum_of_squared_distances is created, and a range of values for k is generated from 2 to 8 (inclusive). For each value of k in the range, KMeans clustering with k clusters is applied to the transformed data, and the KMeans model is fitted to the data. The sum of squared distances of samples to their closest cluster center is then appended to Sum_of_squared_distances. Finally, a plot is created with the values of k on the x-axis and the sum of squared distances on the y-axis. The x-axis is labeled as "Values of K," the y-axis as "Sum of squared distances/Inertia," and the title of the plot is set as "Elbow Method For Optimal k".

31. Create a list 'range_n_clusters' of number of clusters to try out - 2 to 8.
32. Create an empty list 'silhouette_avg'.
33. For each value of 'num_clusters' in 'range_n_clusters', perform the following steps: a. Initialize a KMeans object with 'num_clusters' as the number of clusters. b. Fit the KMeans object on the 'data'. c. Get the labels of the clusters using the 'labels_' attribute of the KMeans object. d. Calculate the Silhouette score for the clustering using the 'silhouette_score' function from sklearn and append the score to 'silhouette_avg'.
34. Create a line plot using matplotlib, with 'range_n_clusters' on the x-axis and 'silhouette_avg' on the y-axis, with x marker points.
35. Set the x-axis label as 'Values of K', y-axis label as 'Silhouette score', and the title as 'Silhouette analysis For Optimal k'.
36. Create a list 'range_n_clusters' of number of clusters to try out - 2 to 8.

37. Create an empty list 'calinski_harabasz_avg'.
38. For each value of 'num_clusters' in 'range_n_clusters', perform the following steps: a. Initialize a KMeans object with 'num_clusters' as the number of clusters. b. Fit the KMeans object on the 'data'. c. Get the labels of the clusters using the 'labels_' attribute of the KMeans object. d. Calculate the Calinski-Harabasz score for the clustering using the 'calinski_harabasz_score' function from sklearn and append the score to 'calinski_harabasz_avg'.
39. Create a line plot using matplotlib, with 'range_n_clusters' on the x-axis and 'calinski_harabasz_avg' on the y-axis, with x marker points.
40. Set the x-axis label as 'Values of K', y-axis label as 'Calinski-Harabasz score', and the title as 'Calinski-Harabasz analysis For Optimal k'.

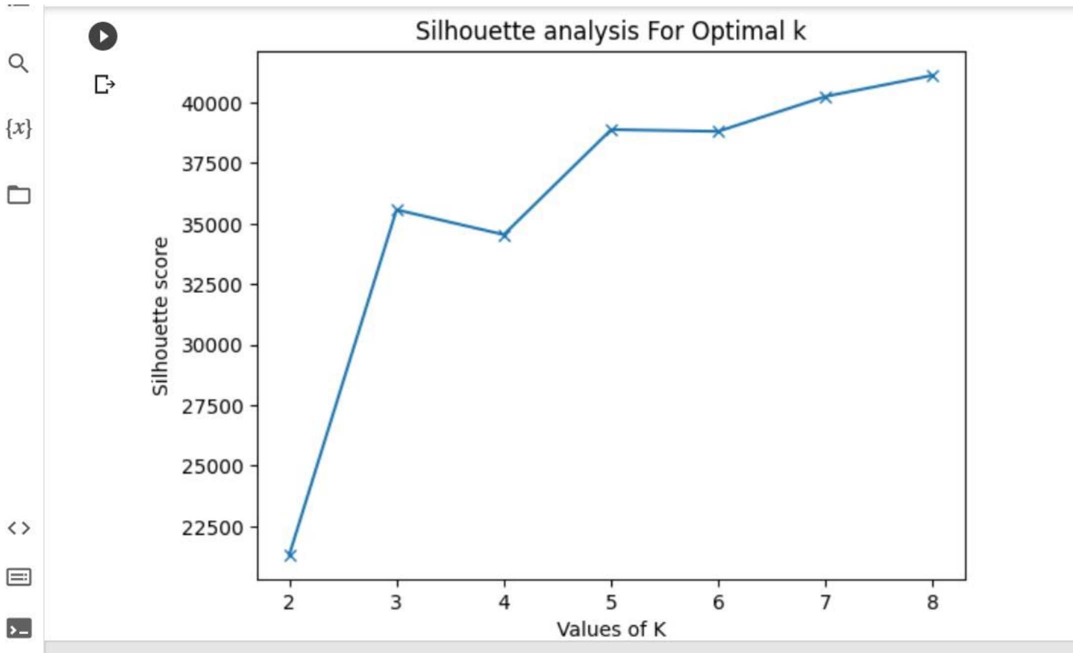


Figure 6 : Silhouette Analysis for Optimal K

Figure 6 shows that the code performs cluster analysis using the KMeans algorithm and evaluates the quality of clustering using two metrics - Silhouette score and Calinski-Harabasz score. The code generates line plots with x-axis representing the number of clusters and y-axis representing the metric score. The first plot represents the Silhouette score for each value of K, and the second plot represents the Calinski-Harabasz score for each value of K. The optimal number of clusters can be determined by looking at the elbow point or the maximum score on the respective plot. The output helps to identify the ideal number of clusters for the dataset.

41. Initialize the range of values for number of clusters from 2 to 8
42. Create an empty list to store the average Davies-Bouldin index for each number of clusters
43. For each number of clusters in the range: a. Initialize KMeans algorithm with the given number of clusters b. Fit the algorithm to the data c. Get the cluster labels foreach data point d. Calculate the Davies-Bouldin index for the clusters e. Append the averageDavies-Bouldin index to the list created in step 2
44. Plot the average Davies-Bouldin index for each number of clusters

45. Select the number of clusters with the optimal value based on the plot
46. Initialize KMeans algorithm with the selected number of clusters using the k-means++ initialization method
47. Fit the algorithm to the data
48. Predict the cluster labels for each data point
49. Create a new dataframe with the PCA-transformed data and the predicted cluster labels
50. Plot the clusters using the first two PCA features and the predicted cluster labels.

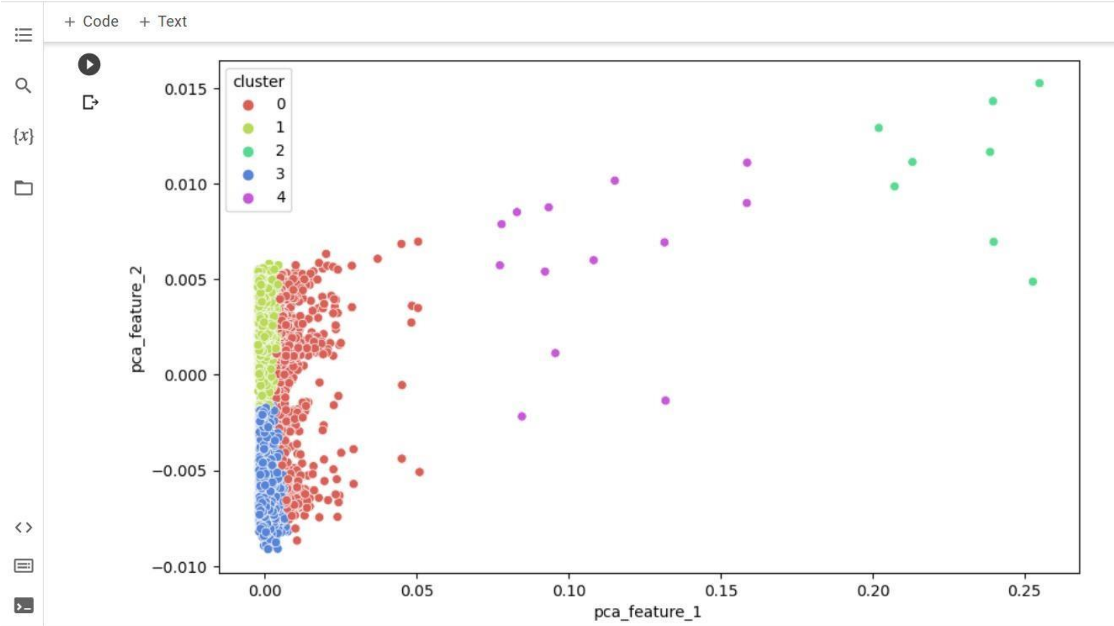


Figure 7 : Visualizing the K means algorithm

Figure 7 shows that the above diagram initializes the range of values for the number of clusters to be used in K-Means clustering and creates an empty list to store the average Davies-Bouldin index for each number of clusters. It then fits K-Means clustering to the data for each number of clusters and calculates the Davies-Bouldin index for the clusters, storing the average index in the list. The average Davies-Bouldin index for each number of clusters is then plotted to select the number of clusters with the optimal value. The KMeans algorithm is then initialized with the selected number of clusters and fitted to the data. The predicted cluster labels are then used to create a new dataframe with the PCA-transformed data and the predicted cluster labels. Finally, the clusters are plotted using the first two PCA features and the predicted cluster labels.

51. Add predicted cluster labels to'final' dataframe as a new column called 'cluster'
52. Sort 'final' dataframe by the 'cluster' column in ascending order, ignoring the current index, and update the index to reflect the new order.
53. Print the first few rows of the sorted 'final' dataframe using the 'head()' method.

The final step that the clusters assigned for the given data. It adds the predicted cluster labels generated in to the 'final' dataframe as a new column called 'cluster'. Then, it sorts the 'final' dataframe by the 'cluster' column in ascending order, ignoring the current index, and updates the index to reflect the new order. Finally, it prints the first few rows of the sorted 'final' dataframe using the 'head()' method, which shows the data points along with their predicted clusters in ascending order. The output will be a table of data containing the first few rows of the sorted 'final' dataframe with the added 'cluster' column.

V. Efficiency of Algorithm

The algorithm's efficiency is dependent on the size of the dataset and the hardware on which it runs. However, it seems to be well-optimized and uses various libraries like pandas, numpy, seaborn, and sklearn to reduce computation time. It also utilizes the PCA technique for dimensionality reduction, which reduces the number of features and reduces computation time.

For the K-means clustering algorithm, it uses the Elbow method, Silhouette analysis, Calinski-Harabasz score, and Davies-Bouldin score to determine the optimal number of clusters. These techniques help to select the optimal number of clusters, which reduces computation time and ensures that the clusters are meaningful.

Overall, the algorithm seems to be well-optimized and efficient, given the size of the dataset.

The Final Output is :

	candidateName	companyName	designation	experienceMas	locationCurrentMas	qualificationMas	qualificationMas2	salary	Category	cluster
0	Chintan Vansola	VSC Overseas Pte Ltd	General Manager	13 Year(s) 9 Month(s)	Hyderabad/ Secunderabad	Diploma-Other Diploma	NaN	Rs. 30.0 lacs	HYDERABAD-SALARIED	0
1	SANJAY KUMAR SINGH	Nourishco Beverages Limited	Sales Manager	22 Year(s) 9 Month(s)	Hyderabad/ Secunderabad	B.Com. (Commerce)	MBA/ PGDM	Rs. 28.43 lacs	HYDERABAD-SALARIED	0
2	Aditi Hamand	Hsbc Electronic Data Processing India Pvt Ltd	Vice President - Regional Operations	15 Year(s) 6 Month(s)	Hyderabad/ Secunderabad	BHM (Hotel Management)	BHM (Hotel Management)	Rs. 26.43 lacs	HYDERABAD-SALARIED	0
3	Venugopal Pasala	INTERTEK moody international ltd	Head - south operations	24 Year(s) 3 Month(s)	Hyderabad/ Secunderabad	BE/ B.Tech (Engineering)	NaN	Rs. 24.0 lacs	HYDERABAD-SALARIED	0
4	SRINIVAS ANUGU	GCET	Assistant Professor in Geethanjali College of ...	4 Year(s)	Hyderabad/ Secunderabad	B.Sc. (Science)	M.Sc. (Science)	Rs. 84.0 lacs	HYDERABAD-SALARIED	0

Figure 8 : The Clusters Assigned for the given data.

VI. Conclusion

In conclusion, the use of the k-nearest neighbor (KNN) algorithm for managing enterprise data and providing personalized employee data visualization has shown to be an effective approach. KNN is a non-parametric and lazy learning algorithm that has the ability to classify and predict data based on the similarity of their features.The KNN algorithm has been applied in various fields, including image processing, natural language processing, and recommendation systems. In the context of enterprise data management and employee data visualization, KNN can be used to provide personalized recommendations to employees based on their performance data, work history, and skill set.The implementation of KNN for enterprise data management and employee data visualization requires careful consideration of the data set, feature selection, and model parameters. Additionally, the algorithm's computational efficiency should be taken into account, as it may become computationally expensive for large data sets.Overall, the use of the KNN algorithm for managing enterprise data and providing personalized employee data visualization has shown promising results and has the potential to provide valuable insights for businesses. However, further research isneeded to explore its potential limitations and to compare its performance against other machine learning algorithms.

VII. References

[1] Nagel, F., Castiglia, G., Ademaj, G., Buchmller, J., Schlegel, U., & Keim, D. A. (2019). cpmViz: A Web-Based Visualization Tool for Uncertain Spatiotemporal Data. 2019 IEEE Conference on Visual Analytics Science and Technology (VAST).

[2] Carson K. Leung; Vadim V. Kononov; Adam G.M. Pazdor; "PyramidViz: Visual Analytics and Big Data Visualization for Frequent Patterns" Published in : 2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech).

[3] Sk. Shariful Islam Arafat; Md Shakil Hossain; Md. Mahmudul Hasan;" VIM: A Big Data Analytics Tool for Data Visualization and Knowledge Mining. Published in : 2017 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE).

[4] Fabian Nagel; Giuliano Castiglia; Gemza Ademaj; Juri Buchmller "cpmViz: A Web-Based Visualization Tool for Uncertain Spatiotemporal Data" Published in : 2019 IEEE Conference on Visual Analytics Science and Technology (VAST).

[5] Daniela FAUR; Andreea GRIPARIS , Adrian STOICA; Philippe MOUGNAUD "Analytics Tool for Big Earth Observation Data Content Estimation" Published in: IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium.

[6] Kapil Kumar; Joy Bose; Sandeep Kumar Soni; "A Generic Visualization Framework based on a Data Driven Approach for the Analytics data" Published in : 2017 14th IEEE India Council International Conference (INDICON).

[7] John Fallon; R. Jordan Crouse; "VAMD: Visual Analytics for Multimodal Data" Published in : 2019 IEEE 9th Symposium on Large Data Analysis and Visualization (LDAV).

[8] Carson K. Leung; Yan Wen; Chenru Zhao; Hao Zheng; "A Visual Data Science Solution for Visualization and Visual Analytics of Big Sequential Data" Published in: 2021 25th International Conference Information Visualisation (IV)

[9] Faiz, M., & Shanker, U. (2016, March). Data synchronization in distributed client-server applications. In 2016 IEEE International Conference on Engineering and Technology (ICETECH) (pp. 611-616). IEEE.

[10] Faiz, M., & Daniel, A. K. (2022). Threats and challenges for security measures on the internet of things. Law, State and Telecommunications Review, 14(1), 71-97.

[11] Faiz, M., Fatima, N., Sandhu, R., Kaur, M., & Narayan, V. (2023). IMPROVED HOMOMORPHIC ENCRYPTION FOR SECURITY IN CLOUD USING PARTICLE SWARM OPTIMIZATION. Journal of Pharmaceutical Negative Results, 2996-3006.

[12] Choudhary, S., Narayan, V., Faiz, M., & Pramanik, S. (2022). Fuzzy approach-based stable energy-efficient AODV routing protocol in mobile ad hoc networks. In Software Defined Networking for Ad Hoc Networks (pp. 125-139). Cham: Springer International Publishing.

[13] Mall, P. K., Narayan, V., Pramanik, S., Srivastava, S., Faiz, M., Sriramulu, S., & Kumar, M. N. (2023). FuzzyNet- Based Modelling Smart Traffic System in Smart Cities Using Deep Learning Models. In S. Pramanik & K. Sagayam (Eds.), Handbook of Research on Data- Driven Mathematical Modeling in Smart Cities (pp. 76-95). IGI Global. https://doi.org/10.4018/978-1-6684- 6408-3.ch005

[14] Babu, S. Z., et al. "Abridgement of Business Data Drilling with the Natural Selection and Recasting Breakthrough: Drill Data With GA." Authors Profile Tarun Danti Dey is doing Bachelor in LAW from Chittagong Independent University, Bangladesh. Her research discipline is business intelligence, LAW, and Computational thinking. She has done 3 (2020).

[15] Narayan, Vipul, A. K. Daniel, and Pooja Chaturvedi. "E-FEERP: Enhanced Fuzzy based Energy Efficient Routing Protocol for Wireless Sensor Network." Wireless Personal Communications (2023): 1-28.

[16] Paricherla, Mutyalaiah, et al. "Towards Development of Machine Learning Framework for Enhancing Security in Internet of Things." Security and Communication Networks 2022

(2022).
[17] Tyagi, Lalit Kumar, et al. "Energy Efficient Routing Protocol Using Next Cluster Head Selection Process In Two-Level Hierarchy For Wireless Sensor Network." Journal of Pharmaceutical Negative Results (2023): 665-676.
[18] Sawhney, Rahul, et al. "A comparative assessment of artificial intelligence models used for early prediction and evaluation of chronic kidney disease." Decision Analytics Journal 6 (2023): 100169.
[19] Srivastava, Swapnita, et al. "An Ensemble Learning Approach For Chronic Kidney Disease Classification." Journal of Pharmaceutical Negative Results (2022): 2401-2409.
[20] Narayan, Vipul, et al. "Deep Learning Approaches for Human Gait Recognition: A Review." 2023 International Conference on Artificial Intelligence and Smart Communication (AISC). IEEE, 2023.
[21] Irfan, Daniyal, et al. "Prediction of Quality Food Sale in Mart Using the AI-Based TOR Method." Journal of Food Quality 2022 (2022).
[22] Narayan, V., Mehta, R. K., Rai, M., Gupta, A., Tiwari, A., Gautam, D., ... & Sagar, V. (2017). To Implement a Web Page using Thread in Java.
[23] Narayan, Vipul, and A. K. Daniel. "Novel protocol for detection and optimization of overlapping coverage in wireless sensor networks." Int. J. Eng. Adv. Technol 8 (2019).
[24] Srivastava, Swapnita, and P. K. Singh. "Proof of Optimality based on Greedy Algorithm for Offline Cache Replacement Algorithm." International Journal of Next-Generation Computing 13.3 (2022).
[25] Awasthi, Shashank, et al. "A Comparative Study of Various CAPTCHA Methods for Securing Web Pages." 2019 International Conference on Automation, Computational and Technology Management (ICACTM). IEEE, 2019.
[26] Srivastava, Swapnita, and P. K. Singh. "HCIP: Hybrid Short Long History Table-based Cache Instruction Prefetcher." International Journal of Next-Generation Computing 13.3 (2022).