MENTAL HEALTH IN TECHNOLOGY-RELATED JOBS

An Unsupervised Learning Project

Author: Jedidiah Amos
Matriculation No.: 32009079
Course: DLBDSMLUSL01
Date: 24.08.2025

Table of Contents

List of Figures

1. Introduction

The Human Resources (HR) department is launching a program to address mental health concerns among respondents and has enlisted me as the expert data scientist to analyze survey results from technology related employees.

1.1 How clustering can help HR

HR needs a clear and structured analysis of a complex dataset with high dimensionality, missing values, and unstandardized text inputs. By applying unsupervised machine learning clustering, survey respondents were grouped based on their responses, enabling insightful visualizations that highlighted key patterns. These clusters will help HR interpret the data more effectively and make informed decisions in applying workplace changes.

1.2 Introducing the dataset and its source

The dataset consisted of 63 features (columns) and 1,433 respondents (rows), making it highly dimensional. Early Exploratory Data Analysis (EDA) revealed a significant presence of missing values across all features, with some columns having more than half of their entries missing. Additionally, categorical features - which included various data types such as strings, floats, and integers - were highly unstandardized, with some containing over 50 to 1,000 unique values. This level of variability would make visualization, interpretation, and meaningful comparisons between features difficult without employing clustering techniques, such as K-means clustering.

1.3 Aim

The aim of this project was to clean and transform the dataset before applying clustering techniques to uncover relationships between features and their overall significance. These insights, supported by visualizations and analysis, will help the HR department understand mental health challenges in the tech sector and develop effective policies/strategies. The machine learning pipeline is outlined in this document to provide the HR department with context on how the model is developed, helping them understand both its benefits and limitations. The workflow pipeline follows four main steps:

1. **Data Preprocessing** - Cleaning the dataset to ensure quality before building the machine learning model.
2. **Dimensionality Reduction (PCA)** - Reducing the feature space while preserving key variance.
3. **Clustering (K-Means)** - Grouping data points for easier interpretation.
4. **Visualization & analyses** - Presenting findings to HR to support workplace improvements throughout the process.

This approach mirrored the workflow pipeline in the notebook file unsupervised_ML_for_MH_in_TECH.ipynb (hereafter referred to as TECH.ipynb) provided with this document. The model was built in **python** using the **jupyter lab** environment.

2. Data preprocessing

Before building the machine learning model, the dataset underwent preprocessing to ensure quality and consistency. Significant time was spent refining the data into a structure suitable for clustering. The preprocessing steps were modified from Learn with Ankith (2023) and included:

1. **importing** necessary libraries,
2. **reading** the dataset,
3. conducting a **data inspection**/sanity check,
4. **Exploratory Data Analysis (EDA)** being performed to identify patterns,
5. **missing value** treatments,
6. **outlier** treatment,
7. removal of duplicates or **garbage values**,
8. data **normalization** for consistency,
9. one-hot **encoding** to prepare categorical features for analysis,
10. and **scaling**, primarily for age.

Collectively, these steps refine the dataset, making it suitable for clustering and visualization.

2.1 Importing libraries, reading dataset and data inspection

The first three steps of data preprocessing were straightforward. Necessary libraries were imported to support subsequent preprocessing tasks. The dataset, provided in CSV format, was uploaded to the Jupyter environment in the same directory as TECH.ipynb. It was then loaded into the notebook as a dataframe (df) using pd.read_csv. To ensure ease of use in later stages, all 63 column names were renamed and shortened based on the list provided by Olteanu (2020). Initial data inspection revealed a structure of 1,433 respondents (rows) and 63 features (columns), confirming its high dimensionality. Given this, dimensionality reduction techniques, such as PCA (Principal Component Analysis), were employed to optimize the feature space (Section 3). Further inspections such as identifying which features has more null values as well as the datatypes were also performed.

2.2 EDA

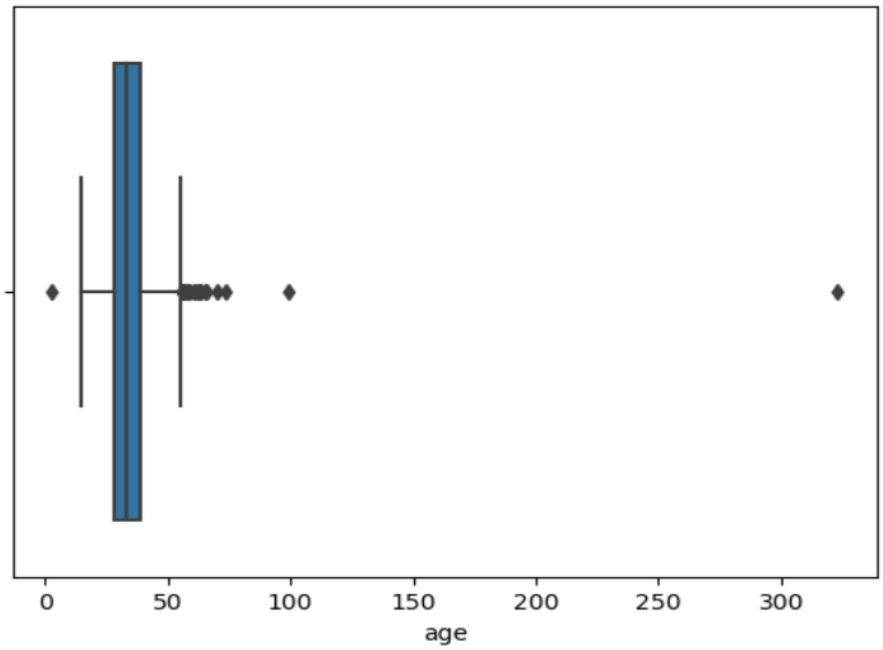
Summary statistics were generated for both numerical and categorical features in the dataset, which comprised 7 numerical columns and 56 categorical columns. The analysis of categorical features revealed significant variability, with several columns containing over 90 unique values, and some exceeding 1,000 distinct entries, reflecting a wide range of responses from survey participants. For numerical columns, descriptive statistics - including count, mean, standard deviation, minimum, 25th percentile, median (50th percentile), 75th percentile, and maximum - were calculated.

Figure 1: Summary statistics for categorical columns

	comp_no_empl	mh_coverage_flag	mh_coverage_awareness_flag	mh_employer_discussion	mh_resources_provided	mh_
count	1146	1146	1013	1146	1146	
unique	6	4	3	3	3	
top	26-100	Yes	No	No	No	
freq	292	531	354	813	531	

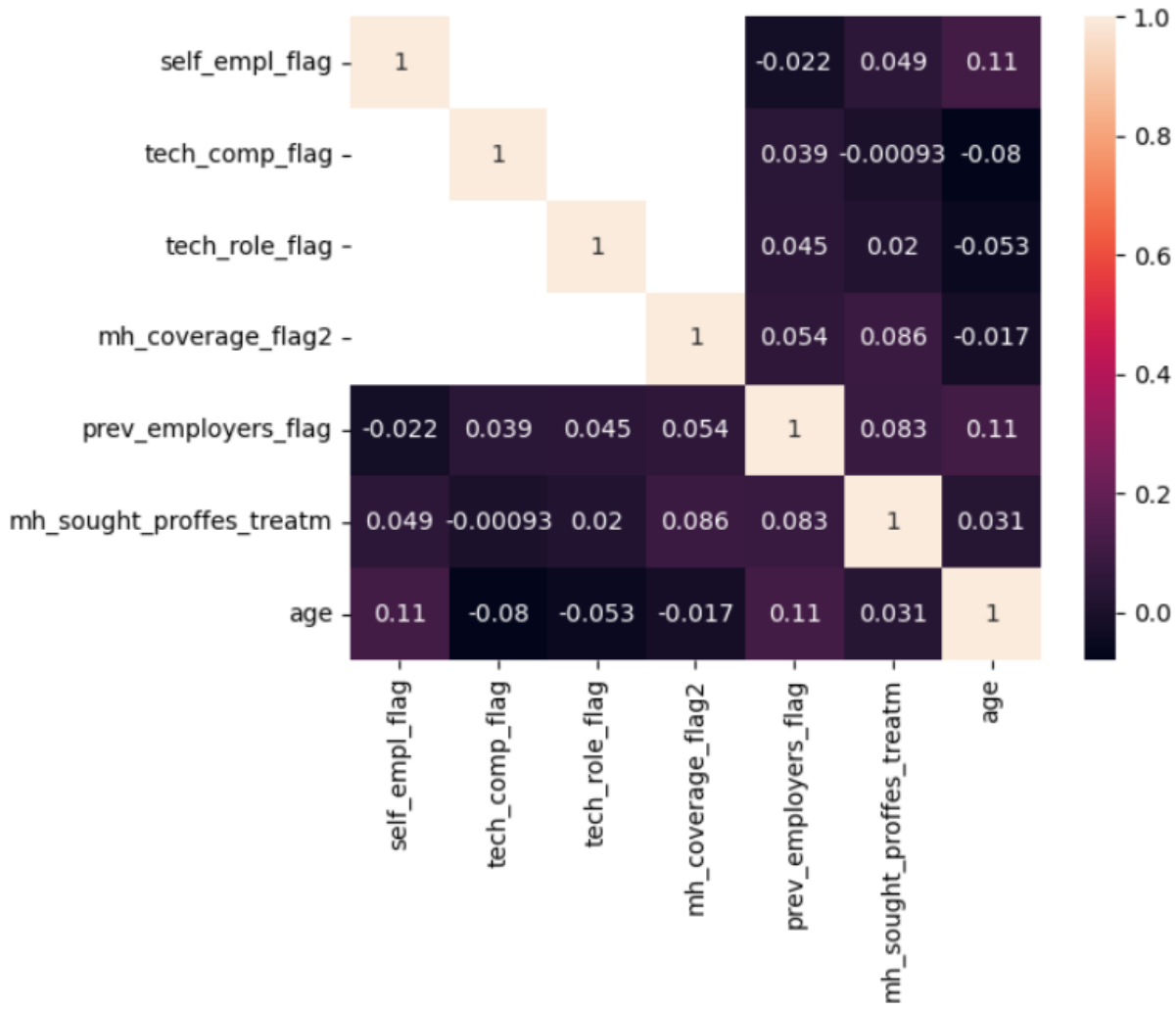
Histogram and boxplot were used to further data exploration of the numerical columns. The histogram showed the count of the Boolean values of 0 and 1 for the respective columns as well as the frequency distribution of the different ages. The box plot of the age variable showed that most values fall between approximately 25 and 45 years, with a median around 35, but there were notable outliers - including extremely low ages (possibly data entry errors or children) and a highly unrealistic value over 100 and 300 that indicated clear data quality issues needed to be addressed.

Figure 2: Boxplot for Age



Furthermore, a correlation matrix combined with a heatmap was used to analyze the relationships and multicollinearity among numerical columns. Generally, correlation values greater than 0.5 indicate a strong relationship, while values below 0.5 suggest a weak correlation. Additionally, a threshold of 0.9 was set, where one column from each highly correlated pair (correlation > 0.9) would be removed to reduce redundant features. Fortunately, no column pairs exhibited a correlation exceeding 0.09, indicating that all numerical columns had very weak correlations.

Figure 3: Correlation x Heatmap of numerical columns



2.3 Missing values

Missing values were initially addressed by dropping columns where the majority of entries were null. As a result, 13 columns with over 50% missing values were removed, reducing the total number of columns from 63 to 50.

Figure 4: New column count

Afterward, null values were handled based on the type of categorical column. Columns requiring mode imputation had missing values replaced with the single most frequently occurring value in the dataset. Meanwhile, other categorical columns were processed using

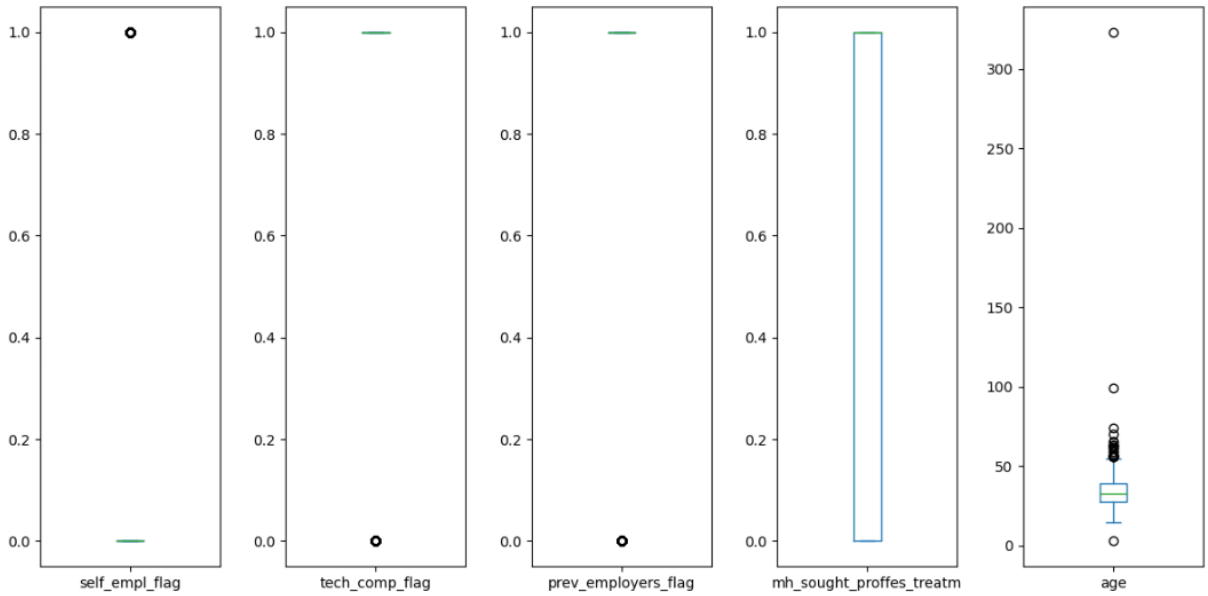
SimpleImputer from scikit-learn with strategy="most_frequent", which selects and imputes missing values based on the most common entry. Notably, the comp_no_empl column was interpreted as a string, preventing median imputation despite being a continuous variable. As a result, mode imputation was applied to handle missing values in this column, while the remaining categorical columns were imputed using the most frequent value via SimpleImputer.

For the numerical columns, all binary columns were imputed using the mode method, as it is the most suitable approach for handling binary data. The age column, however, was imputed using the k-nearest neighbor (KNN) method, which was deemed the best technique for preserving its distribution and relationships within the dataset.

2.4 Outliers

First, outliers in the numerical columns were addressed. After the initial removal of 13 columns with majority null values, the number of numerical columns was reduced from 7 to 5. A new boxplot was generated to visualize outliers in these remaining 5 numerical columns. Since all numerical columns except age contain binary values, their boxplots were not particularly meaningful, as they only illustrate the frequency distribution between two values. However, for the age column, most respondents were between 25-45 years old, while some entries exceeded 100, with one value surpassing 300, which was highly unrealistic. Ages up to 70 are plausible. As an example, Bill gates was 59 years old when he was still Microsoft's chairman of the board. In conclusion, the presence of true outliers were in the age column and likely due to data entry errors (e.g., someone accidentally input 300 instead of 30). These inconsistencies needed to be corrected to ensure data integrity.

Figure 5: Another boxplot of outliers for numerical columns



The outliers for the age column were identified to be 3, 15, 99 and 323. These outliers were fixed by first calculating the median of the ages (excluding the outliers) and then replacing them with the median value. As for the categorical columns, outliers don't apply since they consist of distinct categories rather than numerical values that can be unusually high or low.

2.5 Duplicates & garbage value

To check for duplicate rows in the dataset, the duplicated() function was used without any arguments to examine entire rows to see if any were exact copies across all columns. The output confirmed that there were zero duplicate rows, indicating that each record in the dataset was unique. This step ensured data integrity by verifying that no redundant entries were present before proceeding with further analysis.

2.6 Normalization / Standardization

With 70 unique entries, the gender column required standardization into three categories: Male, Female, and Other. A cleaning function - str(gender).strip().lower() - was implemented to convert values to lowercase, remove extra spaces, and match them against predefined lists of male and female identifiers. Any values not fitting these categories were classified as "Other", ensuring a consistent and structured dataset.

Additionally, the values in the comp_no_empl, country_live, and country_work columns were renamed to ensure consistency. This standardization improves data visualization and plotting, making it easier to analyze trends across these variables.

The work_position column contained a large number of unique categories, making analysis difficult. To simplify it, a threshold of 40 occurrences was set - categories appearing fewer than 40 times were grouped under "Other". The frequency of each category was first counted, and then a new column, work_position_grouped, was created where values meeting the threshold remained unchanged, while less frequent categories were reassigned as "Other".

2.7 Encoding

After completing the prior preprocessing steps, the dataset was ready for one-hot encoding. One-hot encoding was chosen over label encoding because it preserves categorical relationships without assigning arbitrary numerical values that could create a false ranking. This method ensures that each category is represented distinctly, preventing misinterpretation in machine learning models. As a result, the feature space expanded to 310 columns, with 1,433 entries.

Figure 7: One hot-encoding of the categorical columns

work_position_Back-end Developer|Front-end Developer	work_position_DevOps/SysAdmin	work_position_Executive Leadership	work_position_Front-end Developer	work_position_Front-end Developer|Back-end Developer	wor
0	0	0	0	0	
1	0	0	0	0	
0	0	0	0	0	
0	0	0	0	0	
0	0	0	0	0	
0	0	0	0	0	
0	0	0	0	0	
0	0	0	0	0	
0	0	0	0	0	
0	0	0	0	0	
0	0	0	1	0	
0	0	1	0	0	
0	0	0	0	0	
0	0	0	0	0	
0	0	0	0	0	
0	0	0	0	0	
0	0	0	1	0	
0	0	0	0	0	

Figure 6: Shape of the encoded dataframe

```
dummy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1433 entries, 0 to 1432
Columns: 310 entries, self_empl_flag to remote_flag_Sometimes
dtypes: float64(2), int64(3), uint8(305)
memory usage: 482.9 KB
time: 235 ms (started: 2025-05-27 21:33:48 +11:00)
```

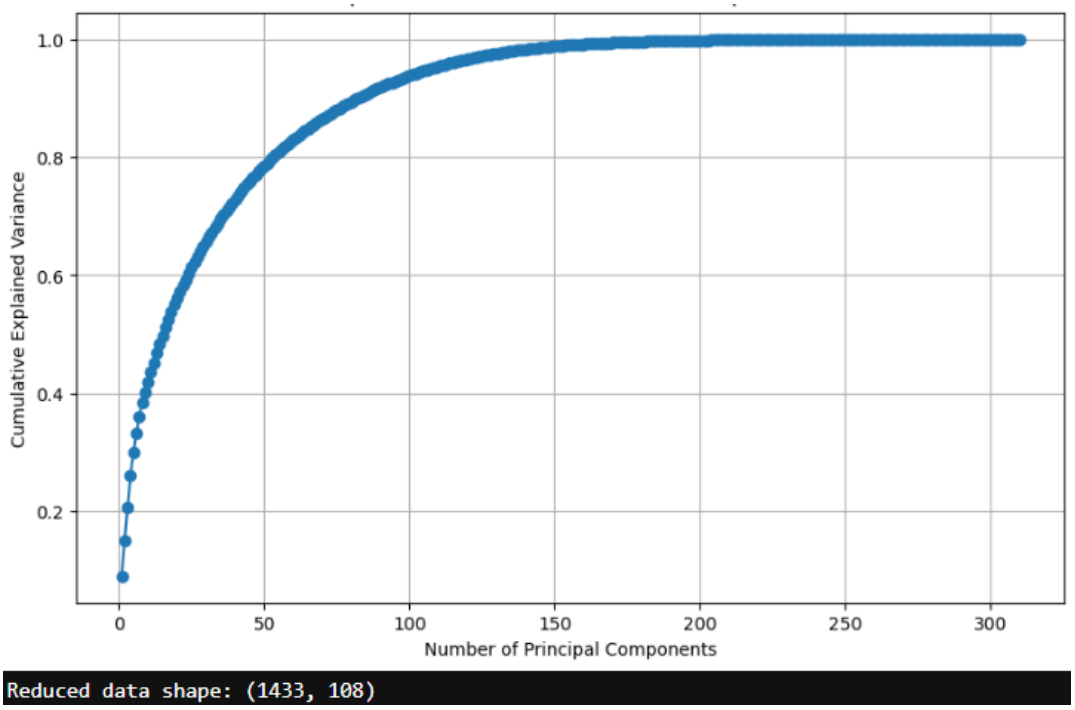
2.8 Scaling

Lastly, the age column of the one-hot encoded dummy dataset was scaled, as PCA is sensitive to differences in feature scales, and age had a significantly larger range compared to the binary variables. Additionally, k-means clustering is also affected by feature scaling as changing the scale of even one feature leads to changing the shape of the feature space (Wongoutong, 2024).

3. Dimensionality reduction via PCA

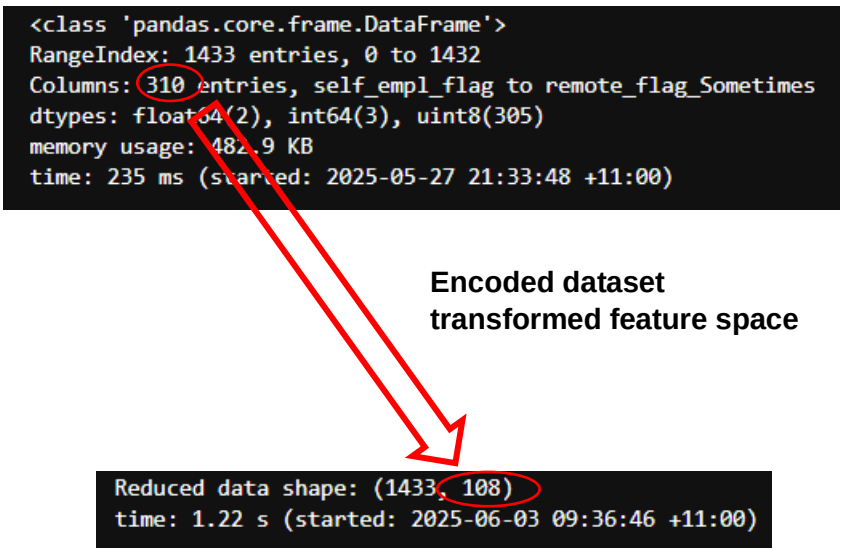
Principal Component Analysis (PCA) was used to reduce dimensionality while preserving most of the variance. First, PCA was applied without limiting the number of components, transforming the dataset into principal components. This allowed for the visualization of a scree plot, which displayed the cumulative explained variance of each component - helping determine the optimal number of components to retain most of the variance (Soriano & Kebabci, n.d.). The scree plot also highlighted the point of diminishing returns, where additional components contribute little explanation (GeeksforGeeks, 2025).

Figure 8: Scree plot of explained variance vs number of components



Finally, PCA was reapplied with `n_components=0.95`, ensuring that only the principal components (PCs) necessary to retain 95% of the variance were kept, effectively reducing redundancy while maintaining essential patterns in the data. The transformed dataset now has fewer features - 108 PCs as seen in figure 8 and the following figure 9 - making the model more efficient while retaining essential data patterns.

Figure 9: Reduced dimensions

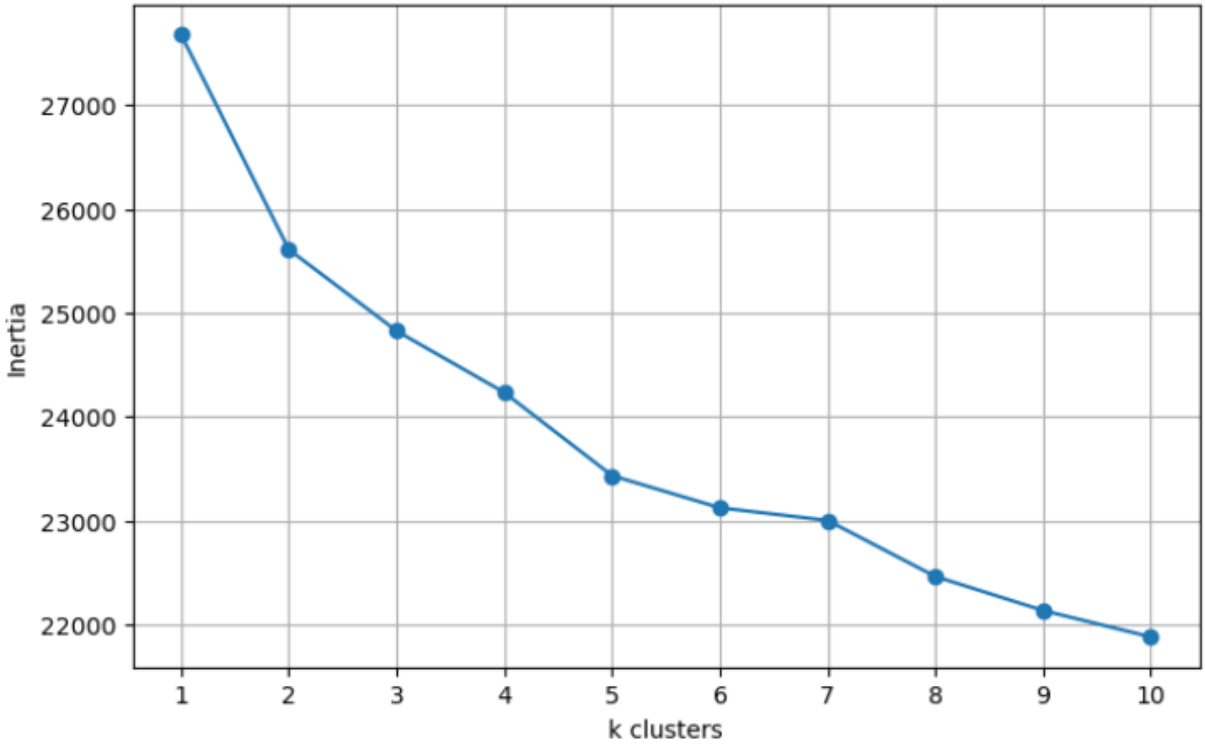


4. K-means clustering

K-means was used as the clustering algorithm to group the data points. First, the elbow method was applied to determine the number of k clusters. An elbow curve was plotted, illustrating the relationship between the number of clusters and inertia (within-cluster SSE - sum of squared errors). This visualization helped identify the optimal number of clusters, where adding more clusters no longer significantly reduced inertia, indicating diminishing returns (Huddar, 2024).

As shown in the figure below, the most significant reductions in inertia occurred between k = 1 and k = 5. After k = 5, the rate of decrease in inertia flattened relatively, indicating diminishing returns. Therefore, k = 5 was selected as the optimal number of clusters. Having 5 clusters means there are 5 different groups of respondents in how they respond to mental health in the technology sector.

Figure 10: Elbow-method - optimal 'k'



After selecting k = 5 from the elbow curve as the optimal number of clusters, k-means clustering was applied to the PCA-reduced dataset, grouping similar data points for further analysis. Figure 11 displays the clusters along with their respective sizes. The dataset was divided into 5 clusters - 0, 1, 2, 3 and 4 - with Cluster 4 being the largest, containing 403 respondents, while Cluster 0 as the smallest, with 230 respondents.

Figure 11: The 5 clusters and their points

```
print(dummy['Cluster'].value_counts())
4    403
3    275
2    273
1    252
0    230
Name: Cluster, dtype: int64
```

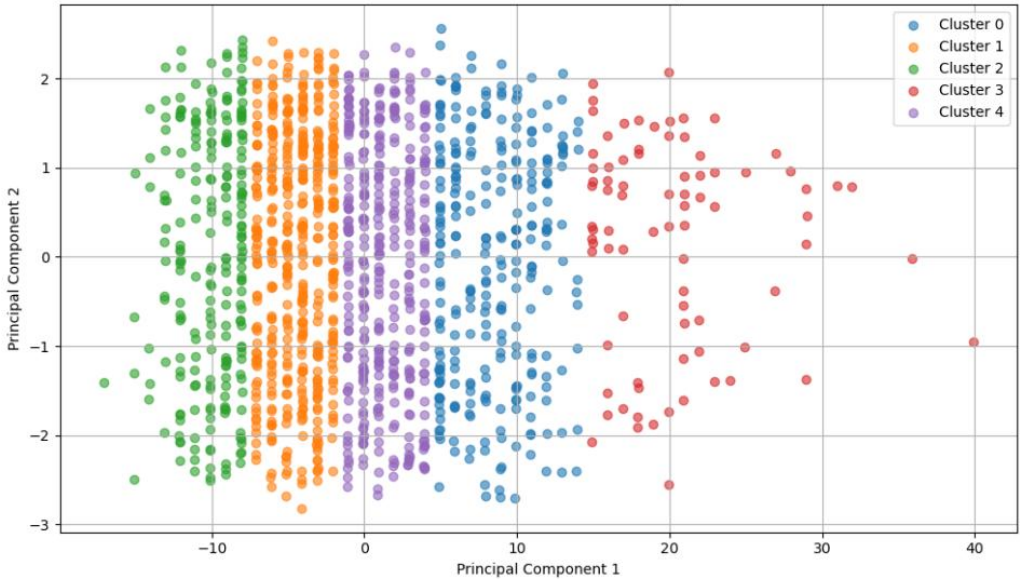
5. Data visualization & analyses

5.1 Visualizing clusters with t-SNE plot

The t-Distributed Stochastic Neighbor Embedding (t-SNE) was chosen over PCA for plotting the 5 K-means clusters because PCA primarily preserves global structure, which can sometimes fail to highlight intricate local relationships within the data (GeeksforGeeks, 2025). Since the dataset still contained 106 other principal components, a standard PCA plot could not effectively separate clusters when reduced to 2D - as it uses only 2 PCs: PC1 and PC2. Conversely, t-SNE is great at clustering similar points by preserving local neighborhoods, thus it was used as the better option for visualizing complex patterns within high-dimensional data (GeeksforGeeks, 2025).

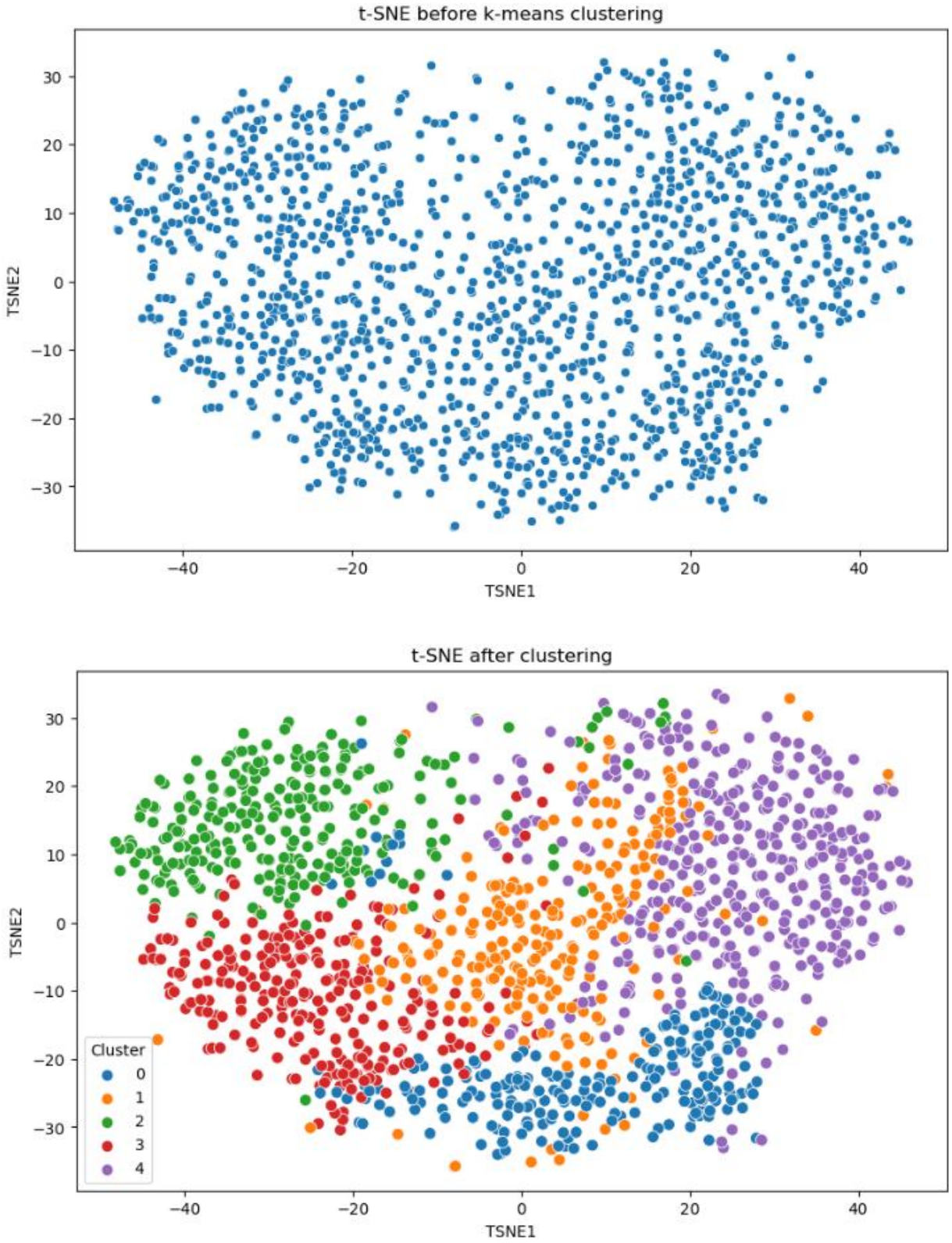
Figure 12 below shows PCA plotting, which was used in one of the earlier versions of TECH.ipynb (NOT included in the current TECH.ipynb).

Figure 12: PCA plotting of the 5 K-means clusters (from earlier version of TECH.ipynb)



Now, compare that with the figure 13 of the t-SNE plot, which highlights the in-depth relationships between data points and reveals the 5 clusters that may not have been as distinguishable in the PCA visualization.

Figure 13: t-SNE visualization of the 5 K-means clusters



The t-SNE plot reveals five distinct clusters with relatively clear separation and minimal overlap, indicating that the K-Means clustering was effective in capturing the underlying structure of the data (Budu, 2024). This supports the selection of k = 5 as determined by the elbow method. While t-SNE effectively reduced the high-dimensional space into two

dimensions, it is important to acknowledge that it can sometimes distort distances, making clusters appear more or less defined than they actually are (Wattenberg et al., 2016). The overlaps in the t-SNE plot are due to clusters sharing some similar PCs, even when limited to only the top 3 contributing PCs. Nonetheless, the visualization shows that the clusters occupy separate regions in the 2D space, suggesting that meaningful differences still exist between them in the original high-dimensional feature space, dummy.

5.2 Cluster profiling

5.2.1 Formatting clusters into a readable output

As Patil (2021) puts it, profiling is a crucial step in the clustering process because it allows us to understand what the clusters actually represent in terms of respondent's behavior and define them in a way that is useful for HR applications. To improve interpretability, dominant features in each cluster were extracted by filtering centroid values above 0.5 and refining feature labels for clarity. This allowed for a more structured presentation of cluster characteristics.

Figure 14: Filtered features with centroid values above 0.5 - for cluster 0

```
Cluster 0:
 - Dominant traits:
   • self_empl_flag
   • tech_comp_flag
   • prev_employers_flag
   • mh_sought_proffes_treatm
   • comp_no_empl_26-100
   • mh_coverage_flag_Yes
   • mh_coverage_awareness_flag_No
   • mh_employer_discussion_No
   • mh_resources_provided_No
   • mh_medical_leave_Somewhat easy
   • ph_discussion_neg_impact_No
   • mh_discussion_supervis_Yes
   • prev_mh_benefits_awareness_N/A (not currently aware)
   • prev_mh_discussion_None did
   • prev_mh_discuss_neg_conseq_Some of them
   • prev_ph_discuss_neg_conseq_Some of them
   • prev_mh_discussion_cowork_Some of my previous employers
   • prev_mh_discussion_supervisor_Some of my previous employers
   • prev_mh_importance_employer_None did
   • future_mh_specification_No
   • mh_family_hist_Yes
   • mh_disorder_past_Yes
   • mh_disorder_current_Yes
   • mh_diagnos_proffesional_Yes
   • mh_not_eff_treat_impact_on_work_Often
   • sex_Male
   • Country_live_USA
   • live_us_teritory_California
   • Country_work_USA
   • work_us_teritory_California
   • work_position_Other
```

Furthermore, to contextualize the clusters within the dataset, an overall profile (overall_profile) was created by calculating the mean of all features. This profile was merged with the transposed cluster profile (cluster_profile.T), enabling direct comparisons between cluster-specific trends and general dataset trends (under df_profile). This allows for meaningful insights into how clusters differ from overall data trends (Patil, 2021; GeeksforGeeks, 2024).

Figure 15: Snippet of overall dataset profile

	0	1	2	3	4	Overall Dataset
self_empl_flag	0.956522	0.000000	0.073260	0.170909	0.000000	2.002791e-01
tech_comp_flag	0.986957	0.722222	0.765568	0.843636	0.794045	8.164689e-01
prev_employers_flag	0.865217	0.884921	0.886447	0.836364	0.918114	8.820656e-01
mh_sought_proffes_treatm	0.800000	0.797619	0.120879	0.145455	0.945409	5.854850e-01
age	0.488802	-0.142357	-0.024217	-0.250867	-0.002360	3.718821e-17
comp_no_empl_100-500	0.004348	0.194444	0.190476	0.170909	0.245658	1.730635e-01
comp_no_empl_26-100	0.986957	0.246032	0.315018	0.385455	0.243176	4.040475e-01
comp_no_empl_500-1000	0.000000	0.043651	0.058608	0.080000	0.076923	5.582694e-02
comp_no_empl_6-25	0.000000	0.190476	0.175824	0.189091	0.153846	1.465457e-01
comp_no_empl_>1000	0.004348	0.246032	0.216117	0.123636	0.248139	1.786462e-01
mh_coverage_flag_No	0.004348	0.353175	0.087912	0.272727	0.059553	1.486392e-01
mh_coverage_flag_Not eligible for coverage / N/A	0.000000	0.150794	0.025641	0.094545	0.029777	5.792045e-02
mh_coverage_flag_Yes	0.986957	0.230159	0.509158	0.360000	0.732010	5.708304e-01
mh_coverage_awareness_flag_No	0.986957	0.595238	0.494505	0.661818	0.198511	5.401256e-01
mh_coverage_awareness_flag_Yes	0.008696	0.150794	0.164835	0.094545	0.486352	2.142359e-01
mh_employer_discussion_No	0.991304	0.829365	0.695971	0.749091	0.662531	7.676204e-01
mh_employer_discussion_Yes	0.008696	0.126984	0.161172	0.192727	0.245658	1.605024e-01
mh_resources_provided_No	0.991304	0.674603	0.399267	0.661818	0.320099	5.708304e-01
mh_resources_provided_Yes	0.004348	0.130952	0.241758	0.134545	0.392060	2.058618e-01
mh_anonimity_flag_No	0.008696	0.154762	0.040293	0.072727	0.029777	5.861828e-02
mh_anonimity_flag_Yes	0.017391	0.119048	0.260073	0.210909	0.389578	2.233077e-01

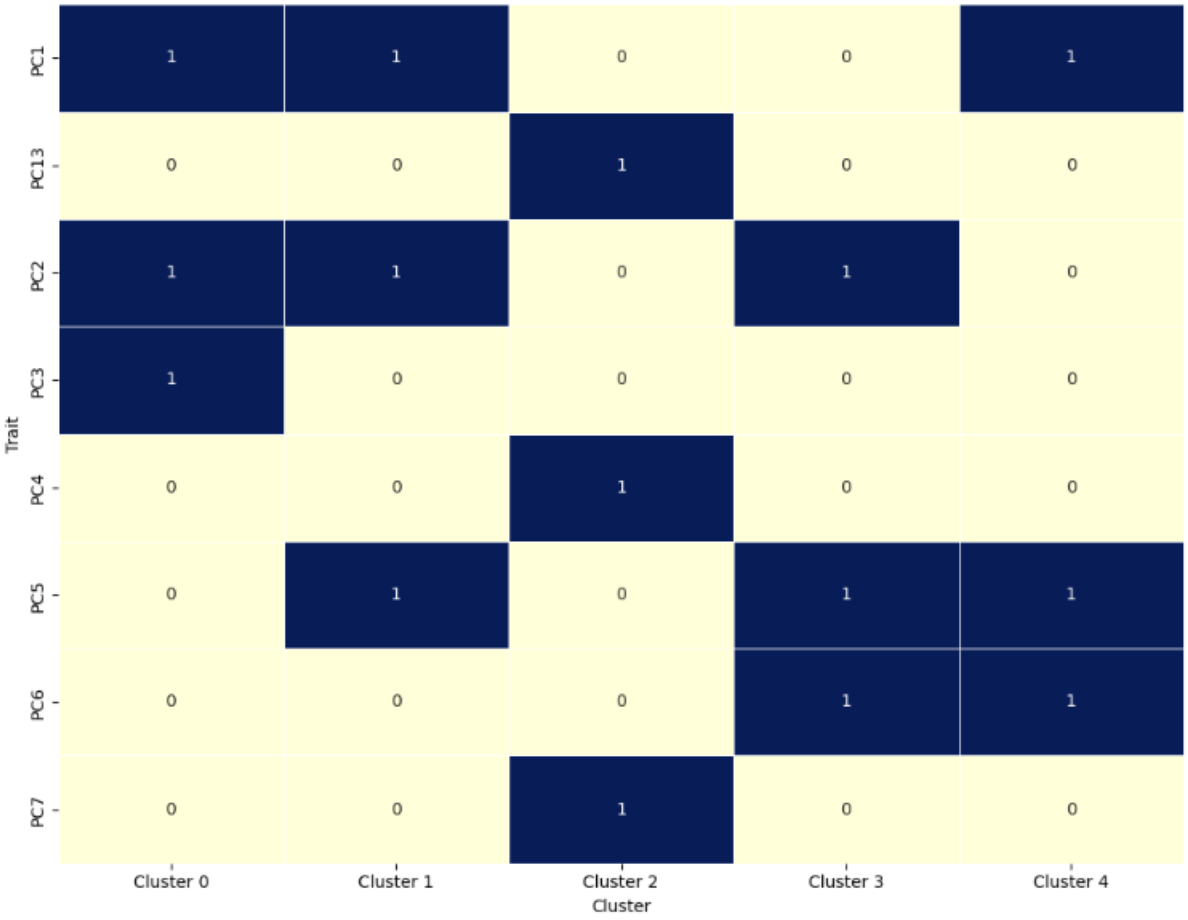
This method provided a quick and computationally efficient approach to identify the top contributing features for each cluster, allowing for an initial profiling of cluster characteristics - specifically, the 5 types of respondents in relation to mental health in technology. However, a more interpretable approach involved selecting the top-n features per cluster and using them to characterize the clusters more effectively, which is discussed in the following section.

5.2.2 Binary heatmap and PC loadings

To interpret the results of the PCA and clustering process, the PCA-reduced array (X_pca_reduced) was first converted into the labeled dataframe reduced_data_df, with each column representing a principal component (e.g., PC1, PC2, etc.). This allowed for easier reference and interpretation of the transformed features.

After clustering was performed, the cluster centroids from the K-Means model were extracted and organized into the dataframe, centroid_df, using the same principal component names. For each cluster, the top 3 most dominant components - with highest centroid values - were identified to understand which dimensions strongly influenced the formation of clusters. As shown in the figure below, the dominant components were compiled into a binary matrix and visualized using a heatmap to illustrate the dominant PCs across clusters.

Figure 16: Heatmap of the 3 dominant traits/PC per cluster



To gain deeper insight into the meaning of each principal component, the PCA loadings were examined by identifying the original features that contributed most to each component. This makes it simple to interpret the PCs in a meaningful way from the pre-PCA dataset (dummy). Using cluster 0 as an example, the 3 dominant PC traits are PC1, PC2 and PC3. The figure below shows the top feature contributors for each of the 3 PCs. Only the top 5 features (based on absolute loading values) were selected for interpretability.

Figure 17: Strongest contributing features for PCs (cluster 0)

```
**For PC1!**
mh_diagnos_proffesional_Yes                          0.282426
mh_disorder_past_Yes                                 0.274402
mh_sought_proffes_treatm                             0.272608
mh_eff_treat_impact_on_work_Not applicable to me     0.272392
mh_not_eff_treat_impact_on_work_Not applicable to me 0.262361
Name: PC1, dtype: float64
time: 78 ms (started: 2025-06-10 09:22:43 +11:00)
```

```
top_features_pc = loadings['PC2'].abs().sort_values(ascending=False).head(5)
print("**For PC2!**")
print(top_features_pc)
```

```
**For PC2!**
live_us_teritory_California    0.253946
country_live_USA               0.252159
work_us_teritory_California    0.251543
country_work_USA               0.249462
mh_resources_provided_No       0.231100
Name: PC2, dtype: float64
time: 218 ms (started: 2025-06-10 09:22:45 +11:00)
```

```
top_features_pc = loadings['PC3'].abs().sort_values(ascending=False).head(5)
print("**For PC3!**")
print(top_features_pc)
```

```
**For PC3!**
age                              0.573054
mh_discussion_supervis_Yes       0.295738
mh_medical_leave_Somewhat easy   0.258015
self_empl_flag                   0.251969
comp_no_empl_26-100              0.228395
Name: PC3, dtype: float64
time: 109 ms (started: 2025-06-10 09:22:47 +11:00)
```

5.2.3 "Un-scaling" age

The reason for loading the PCs was to describe the 5 clusters (0, 1, 2, 3, and 4) as they are still not labeled. However, the PC loadings do not specify an exact age for each feature but rather indicate the degree to which the 'age' feature influences each principal component. To confirm if a cluster tend to skew older or younger, the following table was generated (figure 18).

Figure 18: Statistics table for age (scaled)

```
#the following table is created to be able to identify how old the "age" feature is for each of the PCs.
dummy['cluster'] = cluster_labels
dummy.groupby('cluster')['age'].describe()
```

cluster	count	mean	std	min	25%	50%	75%	max
0	230.0	0.488802	1.242928	-1.864753	-0.503726	0.238653	1.228491	4.940383
1	252.0	-0.142357	0.968926	-1.741024	-0.874915	-0.256267	0.362382	3.950545
2	273.0	-0.024217	0.947295	-1.864753	-0.751186	-0.132537	0.486112	3.331897
3	275.0	-0.250867	0.802051	-2.112213	-0.813050	-0.379996	0.362382	2.589518
4	403.0	-0.002360	0.927345	-1.741024	-0.627456	-0.132537	0.547977	3.579356

The table still only presented the distribution of scaled age values, without providing the actual ages. To address this, the statistical properties of age were first obtained **before** scaling (figure 19). Since the table contained scaled age values (S_A), the original age values were recalculated using the Z-score formula. By applying the mean $(\bar{x})$ and standard deviation (σ) of the age column **prior to scaling**, the actual age (A) was calculated, as shown below (StandardScaler, n.d.).

$$A = (S_A \times \sigma) + \bar{x}$$

Figure 19: Summary statistics of 'age' before scaling

```
#the mean and std of 'age' before being scaled but after cleanising
df['age'].describe()
```

```
count    1433.000000
mean       34.071179
std         8.084951
min        17.000000
25%        28.000000
50%        33.000000
75%        38.000000
max        74.000000
Name: age, dtype: float64
```

From figure 19, $\bar{x} = 34$ and $\sigma = 8$.

$$A = (S_A \times \sigma) + \bar{x}$$

$$\therefore A = (S_A \times 8) + 34$$

From figure 18, using cluster 0 as an example, use the scaled mean for S_A.

$$\therefore A = (0.488802 \times 8) + 34$$

$$\therefore A = 37.910416$$

$$\therefore A \approx 38$$

This equation was extended to calculate the rest of the scaled statistics in figure 18, displayed in figure 20 below. However, to transform the standard deviation column, a different method was used in TECH.ipynb:

Figure 20: Unscaled age stats

cluster	count	mean	std	min	25%	50%	75%	max
0	230.0	37.910415	9.943424	19.081973	29.970192	35.909220	43.827925	73.523068
1	252.0	32.861147	7.751409	20.071811	27.000677	31.949868	36.899058	65.604363
2	273.0	33.806267	7.578359	19.081973	27.990515	32.939706	37.888896	60.655172
3	275.0	31.993061	6.416410	17.102296	27.495596	30.960030	36.899058	54.716144
4	403.0	33.981124	7.418762	20.071811	28.980354	32.939706	38.383815	62.634849

Now the age statistics are more easily interpretable. For instance, cluster 0 has an average age of approximately 38 years old, with a standard deviation of 9.94 years, indicating the highest spread in age of all the 5 clusters. The oldest respondent in cluster 0 is around 73.5 years old, while the youngest is about 19 years old. The 25th percentile age in cluster 0 is about 29.97, meaning 25% of the respondents are younger than 30. Conversely, the 75th percentile is approximately 43.83, which implies that 75% of the respondents in this cluster are younger than 44 years old. In comparison cluster 3, which has the youngest average age of all clusters at around 32 years old but also the smallest spread with a standard deviation of just 6.4 years. Therefore, it can be deduced that respondents in cluster 3 are more tightly grouped around their average age.

Figure 20 gives a clearer view than the scaled version in figure 18. For example, instead of doing mental gymnastics by saying 75% of respondents are "1.23 standard deviations above the mean", it can be said that 75% of respondents in Cluster 0 are under 44, which is more intuitive.

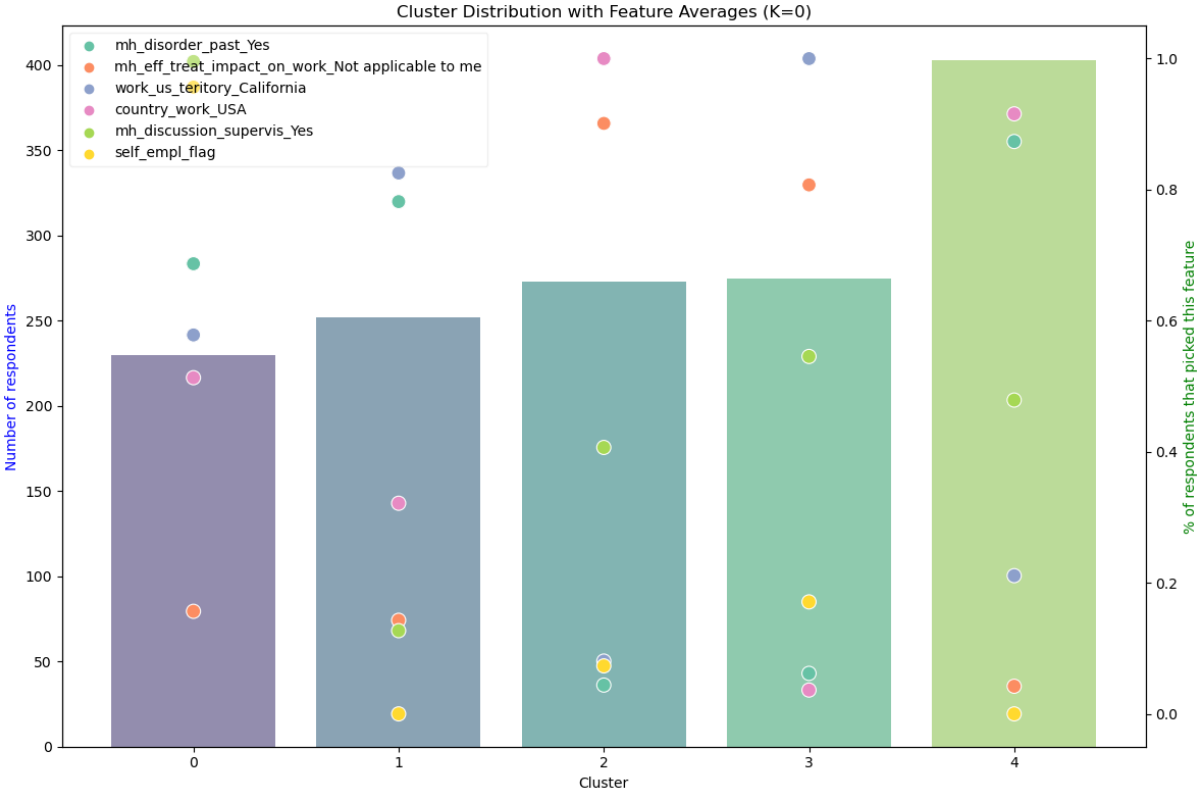
5.2.4 Profiling the clusters with supporting visuals

Now that the age problem had been dealt with, profiling the clusters can commence. Tables 1 to 5 profile each of the top 3 contributing PCs for the 5 clusters using the same loading method as displayed in figure 17 and the un-scaled ages in figure 20 respectively. Additionally, supportive visuals are provided in the form of a combo chart, modified from TrentDoesMath (2024). Each combo chart highlights the top contributing features specific to its respective cluster, while also including data from other clusters to enable direct comparisons - even if those clusters have different dominant features.

Table 1: Profiling PCs (Cluster 0)

Cluster 0	
PC	**Description**
PC1	Figure 17 shows PC1 reflect the mental health history of respondents and impact on work. PC1 captures diagnosis by professionals, past diagnoses, seeking treatment, and perceived impact on work. As seen in figure 21, around 20% of respondents for cluster 0 experience no mental health impact on work.
PC2	PC2 is mostly about location and access to support mental health issues among respondents. High values on this component mainly shows US-based respondents with no mental health support provided by the institution.
PC3	PC3 captures the demographics and workplace openness. A high score of age implies that the age feature contributed significantly to PC3. Thanks to figure 20, it can be said they are roughly 38 years old on average. In addition, PC3 represents respondents who are self-employed or in a small/mid-sized company and who finds it easier to discuss mental health with their superiors or access support. This is supported by figure 21 below, where roughly 98% of respondents in cluster 0 are self-employed with 99% saying they can have open mental health discussions with their supervisors. A pattern can be seen between the clusters where more respondents who are NOT self-employed experience limited mental health discussions with their supervisors.

Figure 21

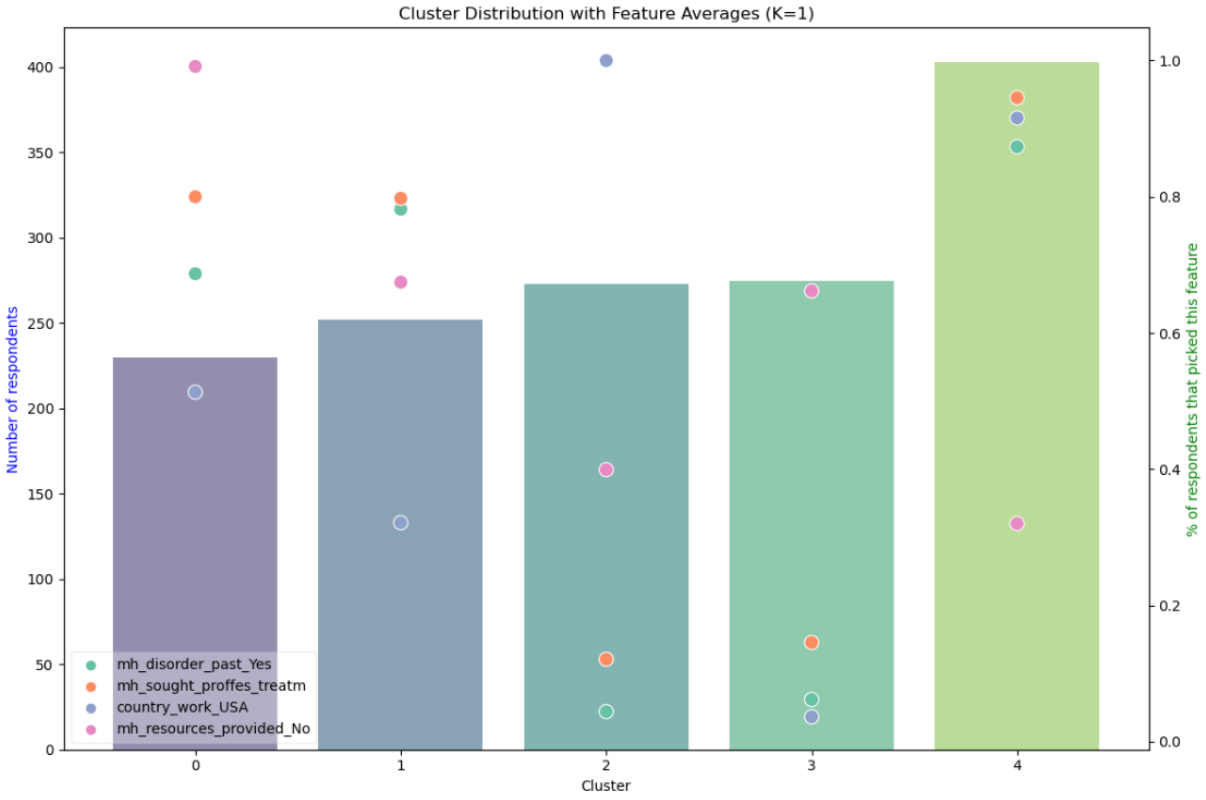


- **Mental health openness:** Figure 21 indicates that cluster 0 has the highest proportion of self-employed individuals who report openly discussing mental health with their superiors - approaching nearly 100%. This suggests a workplace culture that is significantly more open and accepting of mental health conversations.
- **Mental health history & openness:** cluster 0 also ranks among the highest in terms of respondents with a history of mental health disorders. Interestingly, while clusters 2 and 3 have fewer respondents with a mental health history, they still report relatively high levels of open discussion with superiors. This suggests that openness in mental health dialogue may not be solely dependent on personal experience with mental illness, but could also reflect broader cultural or organizational norms that encourage transparency and support. In contrast, clusters 1 and 4 contain the highest number of respondents with past mental health disorders, yet they report significantly lower levels of open discussion with superiors. This disparity may point to environments where stigma remains prevalent, or where individuals feel unsupported in sharing mental health concerns.
- **Treatment impact:** Regarding the impact of mental health treatment on work, 40 individuals in cluster 0 indicated that treatment was "not applicable" to them. This implies that the remaining 190 individuals in that cluster likely received some form of mental health treatment, and yet still felt comfortable discussing their mental health with superiors - further reinforcing the idea that cluster 0 represents a more open and supportive workplace culture.

Table 2: Profiling PCs (Custer 1)

Cluster 1	
PC	**Description**
PC1	Figure 17 shows PC1 reflect the mental health history of respondents and impact on work. PC1 captures diagnosis by professionals, past diagnoses, seeking treatment, and perceived impact on work. As seen in figure 21, around 20% of respondents for cluster 0 experience no mental health impact on work.
PC2	PC2 is mostly about location and access to support mental health issues among respondents. High values on this component mainly shows US-based respondents with no mental health support provided by the institution.
PC5	Loading PC5 shows the top contributing features being (in order): Age, work and living in US-California. As figure 20 shows, cluster 1 has a mean age of 33 years old. .

Figure 22



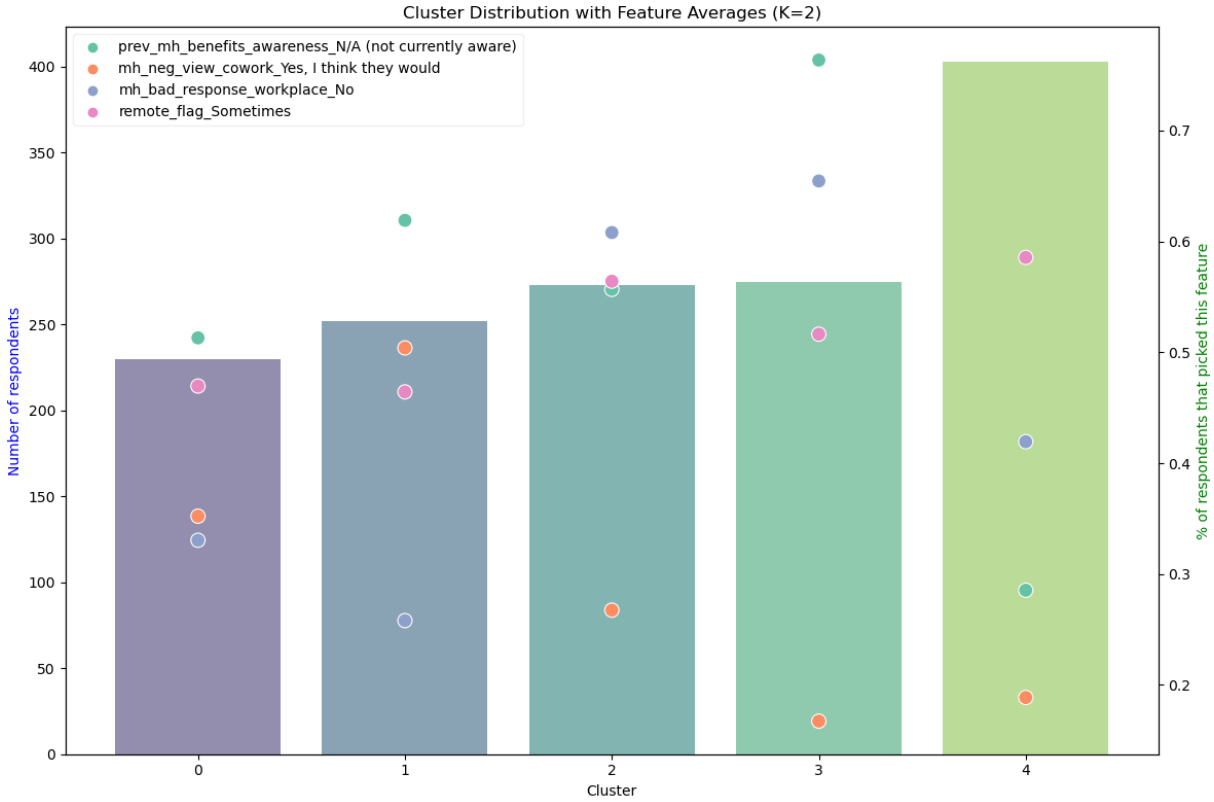
- **Sought professional treatment:** Figure 22 shows that respondents with a past mental health disorder sought professional treatment across all clusters, with the highest incidence in cluster 4. In cluster 1, there is a near 1:1 ratio between individuals with a mental health history and those who pursued professional treatment - approximately 80% of the respondents (202 individuals). This suggests a high level of treatment-seeking behavior, possibly reflecting better personal recognition of symptoms or stronger cultural attitudes toward mental health support within this group.
- **Gaps in workplace mental health resources:** When asked about workplace support, nearly 70% of respondents in cluster 1 reported that no mental health resources were

available to them. This suggests a significant disconnect between personal help-seeking behavior and organizational support systems - employees may be proactive about their well-being but are navigating these challenges without sufficient backing from their employers. In cluster 0, a full 100% of respondents (230) indicated the absence of mental health resources, highlighting a stark lack of workplace support, despite the openness in mental health discussions observed earlier. In contrast, cluster 4 - which had the highest professional treatment rate - reported the lowest absence of resources at 30% (121 respondents), implying that access to support may correlate with higher treatment engagement.

Table 3: Profiling PCs (Cluster 2)

Cluster 2	
PC	**Description**
PC7	PC7 heavily emphasizes previous inaction and very limited engagement with mental health resources. This include not discussing mental health and employers ranging from not valuing to somewhat valuing mental health. Furthermore, PC7 captures respondents who are not aware of any mental health benefits provided by their employer.
PC4	Age has the strongest influence, with the mean being 34 years old. PC4 is also loaded with respondents' beliefs like "mental health is harming career progression" and thinking they get "negative views from coworkers". But when asked if they experienced such beliefs in real life, respondents say "discussing mental health does not negatively impact them" but they still say they do not have discussion with coworkers.
PC13	Reflects "remote_work_sometimes" and respondents' views on how ineffective treatment affects work. Although, PC13 also captures positive views on workplace responses. .

Figure 23

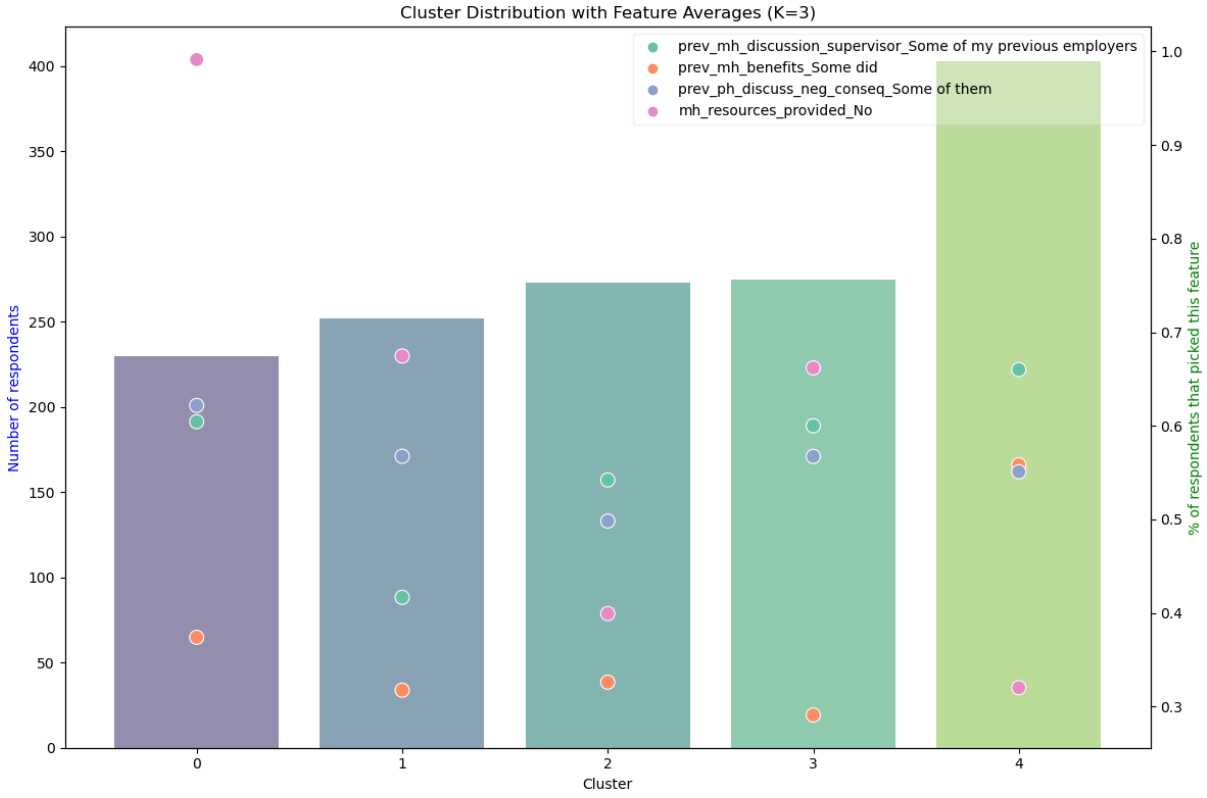


- **Supportive workplace responses:** Figure 23 shows that cluster 2 and 3 stands out with nearly 60% of respondents (165) reporting no negative workplace response to mental health disclosure, the highest among all clusters. This indicates a stronger culture of support, suggesting that policies or management practices in this cluster may already be effective. In contrast, cluster 1 reported the lowest rate of 25% (63), where roughly 45% of the respondents answered to "Yes". Although responses marked "Yes" are not displayed in the figure, they can be visualized in tech.ipynb under section 3.4 by adding the features mh_bad_response_workplace_Yes, I experienced and mh_bad_response_workplace_Yes, I observed to the feature_means variable.
- **Awareness of mental health benefits:** Despite strong workplace responses, about 56% of cluster 2 respondents remain unaware of existing mental health benefits, while clusters 3 and 1 reported the highest level of unawareness. This suggests that communication of resources is not keeping pace with the supportive environment. By comparison, cluster 4 has much lower unawareness (28%).
- **Stigma and remote work context:** Perceived coworker stigma is generally low across clusters with cluster 2 being the middle ground at 26% of respondents, though cluster 1 shows elevated concern with 50% of respondents. Regarding remote work, it appears most common in clusters 2, 3, and 4 (50–60%), which may influence how respondents are not currently aware of any mental health benefits. Furthermore, remote work appears to influence respondents' perceptions of coworker-related stigma, with higher levels of remote work associated with lower perceived stigma.
- **Key takeaway:** A key takeaway for HR is to implement targeted policies and initiatives that raise remote workers' awareness of existing mental health benefits, while also emphasizing how remote work can help reduce stigma from colleagues.

Table 4: Profiling PCs (Cluster 3)

Cluster 3	
PC	**Description**
PC6	Similar to PC7, PC6 is strongly loaded on previous mental health-related workplace experiences, such as: • having prior discussions about mental health with supervisors and coworkers. • many had employers that offered mental health benefits. • many also perceived some negative consequences tied to discussing mental at previous jobs.
PC5	Loading PC5 shows the top contributing features being (in order): Age, work and living in US-California. As figure 20 shows, cluster 1 has a mean age of 33 years old.
PC2	PC2 is mostly about location and access to support mental health issues among respondents. High values on this component mainly shows US-based respondents with no mental health support provided by the institution.

Figure 24



- **Supervisor discussions:** Cluster 3 shows moderate openness, with around 60% of respondents reporting prior mental health discussions with supervisors. This places it above cluster 1 and 2 but behind cluster 4, which remains the most open (around 66%).
- **Previous workplace mental health benefits provided:** Only about 30% of respondents in cluster 3 reported receiving mental health benefits at some of their

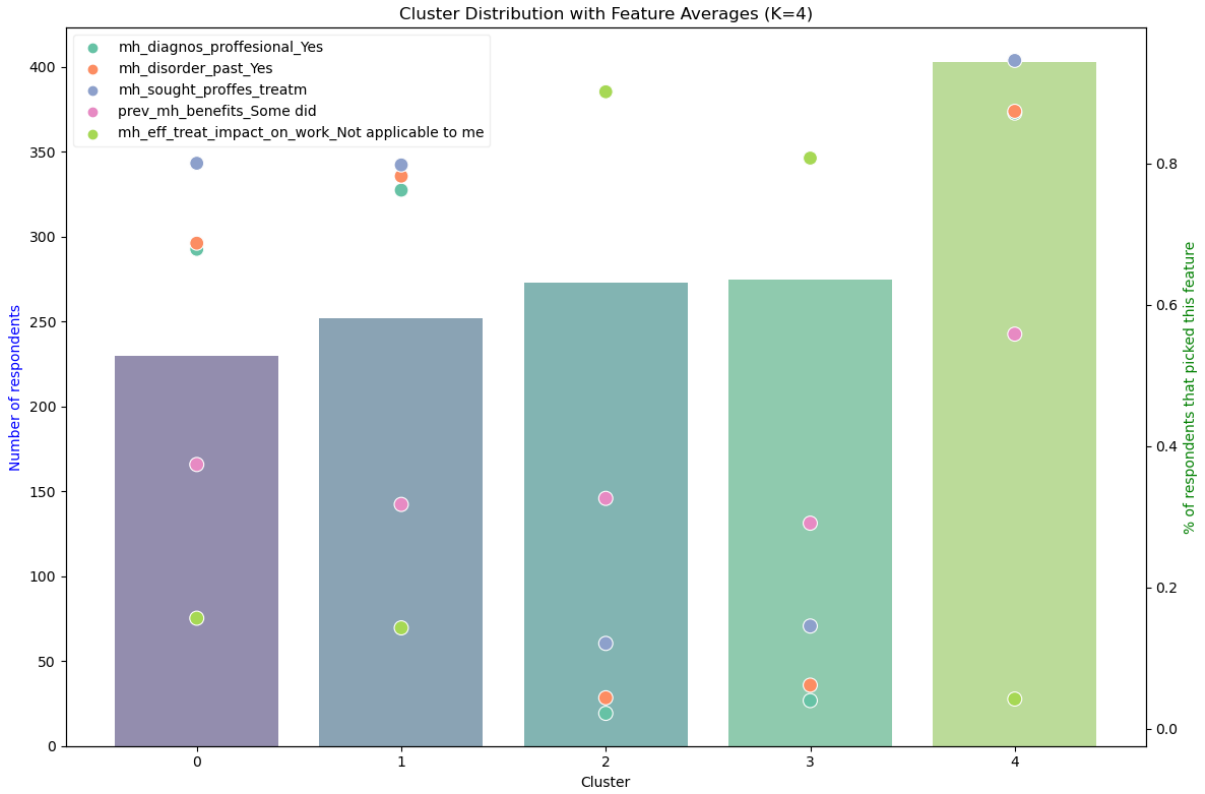
previous jobs. clusters 0 to 2 show similar patterns, ranging between 30% and 38%. In contrast, cluster 4 stands out with the highest rate at approximately 57%. This suggests that clusters 0 to 3 may have had limited access to mental health support in their prior workplaces, whereas cluster 4 likely represents individuals with greater exposure to or prioritization of mental health resources

- **Negative consequences of discussion:** Around 56% of cluster 3 respondents reported some negative consequences from discussing mental health - a rate higher than cluster 2 (the lowest at 50%) but lower than cluster 0 (the highest at 61%). This indicates a mixed environment where stigma is present but not overwhelming in all clusters.
- **Lack of resources:** Approximately 67% of respondents in Cluster 3 reported a lack of mental health resources, similar to Cluster 1. Cluster 0 stands out as the most extreme, with 100% indicating no resources were provided. In contrast, Clusters 4 and 2 show more favorable outcomes, with fewer than 40% reporting insufficient support.

Table 5: Profiling PCs (Cluster 4)

Cluster 4	
PC	**Description**
PC6	Similar to PC7, PC6 is strongly loaded on previous mental health-related workplace experiences, such as: • having prior discussions about mental health with supervisors and coworkers. • many had employers that offered mental health benefits. • many also perceived some negative consequences tied to discussing mental at previous jobs.
PC5	Loading PC5 shows the top contributing features being (in order): Age, work and living in US-California. As figure 20 shows, cluster 1 has a mean age of 33 years old.
PC1	Figure 17 shows PC1 reflect the mental health history of respondents and impact on work. PC1 captures diagnosis by professionals, past diagnoses, seeking treatment, and perceived impact on work. As seen in figure 21, around 20% of respondents for cluster 0 experience no mental health impact on work.

Figure 25



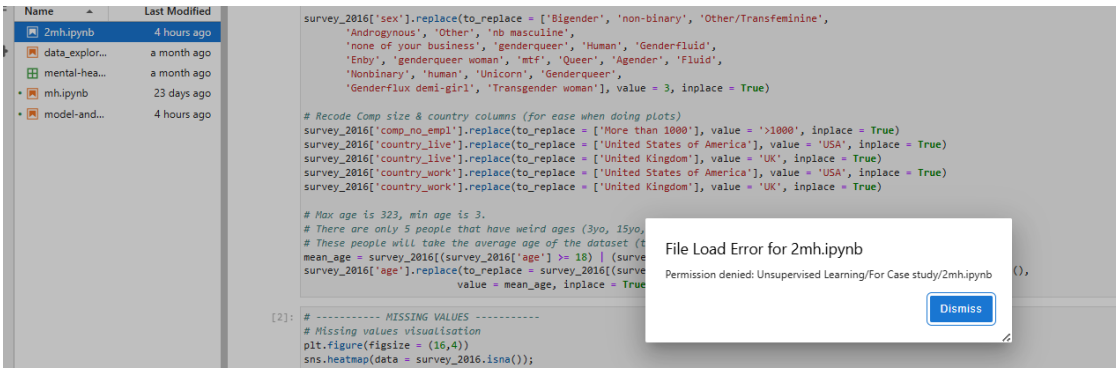
- **Professional diagnosis and past disorder:** Cluster 4 shows the highest rates of professional diagnosis (89%) and past mental health disorder (89% - backed by figure 21). Across all clusters, a trend can be seen where professional diagnosis and reporting a past disorder are closely aligned, and in cluster 4 they occur at the exact same rate of 89%. This strong link suggests that a past mental health disorder is most often identified through a professional diagnosis. However, these features are not entirely redundant. While most people recognize a disorder after being diagnosed, some may self-identify symptoms without formal assessment, therefore distinguishing between “professional diagnosis” and “self-reported past disorder” remains important.
- **Seeking professional treatment:** Cluster 4 has the highest rate of seeking professional treatment at 95% (see figure 22). Clusters 2 and 3 appear much lower, with fewer than 20% seeking treatment. However, since these clusters also report very low rates of professional diagnosis and past disorder, their treatment levels are relatively proportional. This suggests clusters 2 and 3 face fewer challenges in this area.
- **Impact of treatment on work**: around 5% in cluster 4 reported that treatment impact on work was “not applicable,” suggesting that the vast majority may experience some level of impact from treatment on their work. clusters 0 and 1 follow a similar pattern. In contrast, clusters 2 and 3 stand out, with over 80% indicating treatment impact was not applicable – likely because very few in these clusters sought or received treatment in the first place. This highlights a key divide: clusters 0, 1, and especially 4 are more engaged with treatment and thus more likely to report its effects on work, while clusters 2 and 3 remain largely outside this experience.

6. Trials & triumphs

Throughout the case study, several challenges emerged. These are outlined in the table below, along with the corresponding solutions that highlight how each obstacle was successfully overcome.

Trials	Triumphs
During the data cleaning phase, the original notebook file 2mh.ipynb had loading issues (as seen in figure 26) and there was NO backup file manually saved. Causes: Checking the file in windows file manager shows 0 bytes, which means the file was corrupted.	Luckily, the issue was solved due to Jupyter automatically saving backups as checkpoints. To ensure peace of mind for later studies, a backup file will be created.
Initially, the elbow method steps were followed from online sources under the assumption that the same approach applied to this dataset. However, those sources did not include dimensionality reduction before applying k-means, whereas PCA had already been performed in this case. As a result, an unnecessary fit_transform step was applied again, leading to an inaccurate elbow plot and making it difficult to determine the optimal number of clusters.	The issue was identified during a review of the dataframes used in each cell, which revealed that the elbow method was referencing reduced_pca, while the actual PCA-reduced dataset was named X_pca_reduced. This discrepancy stemmed from an earlier duplicated version of TECH.ipynb, where the dataframe name had not been updated. From this, it became clear that the redundant processing steps had already been completed beforehand, allowing them to be removed. This correction ensured that the clustering method was applied properly without unnecessary repetition, leading to a more accurate elbow plot.
During the cluster profiling, I was stuck on how to print the mean values per feature for each cluster while keeping the original names for the features. Initially, they were printed with just their principial component number “PC1, PC2, PC3” etc. This was because the dataset used was X_pca_reduced, from after applying the PCAs, which is a numpy array.	This was fixed by simply assigning the cluster labels when applying the k-means clustering to the dummy one-hot encoded dataset before applying PCA.

Figure 26: Corrupted notebook file



7. Limitation & mitigations

The table outlines the limitations that affect this case study analysis, along with the mitigation strategies that can be applied to prevent or reduce similar issues in future studies.

Limitations	Mitigations
Dataset **misrepresented column data types**, classifying some numeric or mixed-type columns as object. This led to confusion when selecting columns with specific data types for analysis. Attempts to inspect only object columns using select_dtypes() produced unexpected results, seemingly processing all columns regardless of their actual type. Excessive output from print() statements triggered the "IOPub data rate exceeded" error in Jupyter Notebook, temporarily disrupting the communication between the server and interface. These complicated the debugging process and required significant effort to detect (Python Pool, 2021).	Future datasets should have correct **datatypes assigned** to columns.
Library Version Compatibility Issue. Numpy was updated to version 2.2.6 for a separate notebook related to a work project. However, this update affected the current notebook that relies on pandas 1.5.3, which had compatibility concerns with the newer numpy version. This led to potential instability when performing operations involving both libraries. To resolve this, numpy was downgraded to 1.24.4, ensuring compatibility with pandas 1.5.3.	Use **virtual environments** to Isolate different projects so updates in one notebook don't affect another.
While **feature engineering** was performed in the data preprocessing via imputing missing values, outliers, encoding and scaling, not performing any **feature selection** before PCA hindered accurate k-means clustering of the 5 clusters as the t-SNE plot shows. This resulted in the wide scatters and overlaps that shows slight ambiguity in cluster separation.	Perform **Feature selection** before PCA.
Sometimes **missing values** say more about the dataset than present values. Additionally, removing **outliers** can be bad as often times in research, outliers lead to breakthroughs (Infinite codes, 2025).	Something to keep in mind for future endeavors.
Perplexity and **learning rate** were not set for the t-sne plot which could have potentially led to the overlaps between clusters as showcased in figure 13.	For a better cluster separation, perplexity and learning rate must be experimented with iteratively and set. Could also use **UMAP** instead of t-sne (Dobilas, 2021).
In section 5.2.4, the supporting combo charts are limited in the sense that visualizing every feature for each PC per cluster would become to	Alternate visual methods like **Radar Charts** could be employed for better comparison of a feature between clusters.

cluttered and messy. Consequently, some PCs had to be spared.	
Further feature engineering is needed after one-hot encoding. For example, in the dummy dataset (one-hot encoded), multiple variables were created such as: • prev_mh_benefits_Some did, • prev_mh_benefits_Yes, they all did, • prev_mh_benefits_awareness_N/A, • prev_mh_benefits_awareness_No, I only became aware later, and • prev_mh_benefits_awareness_Yes, I was aware of all of them. This resulted in fragmented categories where similar responses (e.g., different forms of "Yes" or "No") were split into separate variables, creating redundancy and reducing clarity.	Similar categories should be consolidated into single variables during preprocessing to reduce redundancy, simplify the dataset, and improve model interpretability.

8. Conclusion

In summary, this project focused on analyzing employees in tech related job survey data to uncover patterns in workplace mental health. The project followed a structured pipeline: data preprocessing, dimensionality reduction, k means clustering, analysis visuals, and critical reflection. The coding work was carried out in Jupyter lab, in the notebook file TECH.ipynb.

The raw dataset was cleansed of missing values, standardized, one hot encoded, and the “age” feature was scaled during data preprocessing. The result was the dummy dataset with 310 features and 1433 responses. PCA was then applied to reduce the dimensionality of the feature space to 108 features with the same number of responses, while retaining 95% of the dataset’s variance. A scree plot (figure 8) was used to identify the optimal number of components by showcasing both the retained variance and diminishing returns.

The reduced dataset was then used in k means clustering to group employees into distinct profiles based on their survey responses. The optimal number of clusters was determined using the elbow method, where 5 clusters (0, 1, 2, 3, and 4) were selected, each representing a unique employee profile with specific workplace mental health awareness, benefit accessibility, and demographic characteristics.

Clusters were visualized using PCA reduced plots and further described with summary statistics and feature comparisons. The age feature was unscaled using the z-score method, making age values interpretable rather than relying on their scaled version. A heatmap (figure 16) was used to identify the top 3 most dominant PCs for each cluster. Each PC’s top 5 contributing features were then identified through PCA loadings, allowing meaningful interpretation of the clusters. The analysis, with the help of the combo charts, revealed insights such as varying levels of awareness of mental health benefits, differences in organizational support, and patterns linked to prior exposure to workplace benefits.

HR now has 5 clusters detailing groups of employees and their workplace mental health experiences, supported by in depth analysis and visuals to guide interpretation. With reliable, cleaned data, HR can use this report to design actions and policies that improve the work environment for mental health

9. Link to GitHub repository

https://github.com/JediPriAmo/Unsupervised-Learning---Mental-health-in-tech-jobs

10. Bibliography

1. Learn with Ankith. (2023). *Data cleaning/data preprocessing before building a model - A comprehensive guide* [Video]. YouTube. https://www.youtube.com/watch?v=GP-2634exqA&t=311s

2. Olteanu, A. (2020). Model and Visualize Mental Health in Tech [Notebook].Kaggle. Model and Visualize Mental Health in Tech

3. Visually Explained. (2021). Principal component analysis (PCA) [Video]. YouTube. https://www.youtube.com/watch?v=FD4DeN81ODY

4. Naik, K. (2018). Principle component analysis (PCA) using sklearn and python [Video]. YouTube. https://www.youtube.com/watch?v=QdBy02ExhGI

5. DataCamp. (2024). t-SNE high-dimensional data visualization | Python tutorial [Video]. YouTube. https://www.youtube.com/watch?v=D9bdJm1GYFY

6. Mcdonald, A. (2021). K-Means clustering algorithm with Python tutorial [Video]. YouTube. https://www.youtube.com/watch?v=iNlZ3IU5Ffw

7. Chen, J. (2017). Data visualization with Python Seaborn. Kaggle. Retrieved from https://www.kaggle.com/code/jchen2186/data-visualization-with-python-seaborn#Cleaning-the-Data

8. Juneja, H. (2018). Preprocessing and random forest with 87% accuracy. Kaggle. Retrieved from https://www.kaggle.com/code/h1rshit/preprocessing-and-random-forest-with-87-accuracy

9. Catherine. (2020). Mental health: EDA, important features. Kaggle. Retrieved from https://www.kaggle.com/code/ptfrwrd/mental-health-eda-important-features#Feature-Selection

10. codebasics. (2019). Machine learning tutorial Python - 13: K Means clustering algorithm [Video]. YouTube. https://www.youtube.com/watch?v=EItlUEPCIzM

11. GeeksforGeeks. (2025, May 19). Difference between PCA vs t-SNE. Retrieved from https://www.geeksforgeeks.org/difference-between-pca-vs-t-sne/

12. GeeksforGeeks. (2024, August 14). Managing high-dimensional data in machine learning. Retrieved from https://www.geeksforgeeks.org/managing-high-dimensional-data-in-machine-learning/

13. Wattenberg, M., et al. (2016). How to use t-SNE effectively. Distill. https://doi.org/10.23915/distill.00002

14. Huddar, M. (2024). Elbow method | Silhouette coefficient method in K-means clustering solved example. YouTube. Retrieved from https://www.youtube.com/watch?v=wW1tgWtkj4I&t=242s

15. W3Schools. (n.d.). Machine learning - K-means. Retrieved from https://www.w3schools.com/python/python_ml_k-means.asp

16. Patil, V. (2021). Clustering and profiling customers using K-means. Medium. Retrieved from https://medium.com/analytics-vidhya/clustering-and-profiling-customers-using-k-means-9afa4277427

17. Wongoutong, C. (2024). The impact of neglecting feature scaling in k-means clustering. PLOS ONE, 19(12), e0310839. https://doi.org/10.1371/journal.pone.0310839

18. Infinite Codes. (2025, March 18). 30 machine learning facts most people get wrong [Video]. YouTube. https://www.youtube.com/watch?v=uEpEEQVGYyQ

19. GeeksforGeeks. (2024, June 26). K-means clustering with SciPy. Retrieved from https://www.geeksforgeeks.org/k-means-clustering-with-scipy/

20. Budu, E. (2024, June 17). *How to interpret a t-SNE plot?* Baeldung. Retrieved from https://www.baeldung.com/cs/t-distributed-stochastic-neighbor-embedding

21. Dobilas, S. (2021, October 25). UMAP dimensionality reduction - An incredibly robust machine learning algorithm. Towards Data Science. Retrieved from https://towardsdatascience.com/umap-dimensionality-reduction-an-incredibly-robust-machine-learning-algorithm-b5acb01de568

22. Scikit-learn Developers. (n.d.). StandardScaler. Scikit-learn. Retrieved June 14, 2025, from https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

23. Sarah, M. (2025). Cluster analysis: A comprehensive guide to cluster analysis-applications, best practices, and resources. Displayr. Retrieved June 16, 2025, from https://www.displayr.com/understanding-cluster-analysis-a-comprehensive-guide

24. TrentDoesMath. (2024, September 1). Hands on data science project: Understand customers with KMeans clustering in Python [Video]. YouTube. https://www.youtube.com/watch?v=afPJeQuVeuY&t=6288s

25. Altius Group. (2025, May 28). 5 strategies to reduce mental health stigma in your workplace. Altius. https://altius.au/news-and-research/5-strategies-to-reduce-mental-health-stigma-in-your-workplace

26. Python Pool. (2021, May 5). What Causes “iopub data rate exceeded” Problem and How to Fix it. Python Pool. https://www.pythonpool.com/iopub-data-rate-exceeded/

27. GeeksforGeeks. (2025, July 23). Implementing PCA in Python with scikit-learn. https://www.geeksforgeeks.org/machine-learning/implementing-pca-in-python-with-scikit-learn/

28. Soriano, P. V., & Kebabci, C. (n.d.). Scree plot for PCA explained. Statistics Globe. https://statisticsglobe.com/scree-plot-pca

29. N.N. (2023). Machine learning - unsupervised learning and feature engineering (Version No. 001-2023-1016). IU Internationale Hochschule GmbH.