# Technical Document

**Problem Statement 1**: The company wants to understand what contributes to employees’ resignation, and how we can help get ahead of the attrition, and what actions can company take to prevent the voluntary termination.

● Please help the business compare the characteristics of the employees that leave voluntarily to the population that does not.

● Identify 2-3 turnover drivers that business can act on to prevent future attrition.

**Assumptions made:**

* Due to the uneven distribution of employees, Company C, D and E are considered as one company
* Terminating employees are automatically moved to HR department by the HRIS system, hence active employees have departments while terminated employees all show HR as department
* Involuntary terminations can have various reasons; hence they are not considered in any analysis

Methodology

1. **EDA & data-processing:**

First, transformed the dataset into a clean and logically consistent data source ready for analysis:

1. **Data-Type Conversion**: Converted key columns into the most appropriate Pandas data types (datetimes, categorical, Booleans) to reduce memory usage and improve processing speed.

A screenshot of a computer code

AI-generated content may be incorrect.

1. **Comprehensive Consistency Validation**  
   Implementing business-rule checks, from date order and activity-status matching to age range and duplicate detection, to systematically identify all rows with contradictory, missing, or out-of-range values. First, we inspect sample violations, then remove every row that failed any rule, yielding a final dataset guaranteed free of the identified inconsistencies.

A screenshot of a computer program

AI-generated content may be incorrect.

1. **Data Cleansing**: The dataset contains ~90k rows of HR records spanning ~9000 unique employees. Each row is either a monthly Snapshot (tracking headcount and status at month-end) or an Activity (a discrete Hire or Termination event). As a result, every person appears dozens of times, once per month plus, if applicable, a hire or termination row, making direct analysis of each employee’s status or attrition patterns cumbersome. To streamline this, the preprocessing pipeline:

* Collapses all history into a single “latest” row per employee by sorting on recorddate\_key (newest first) and dropping duplicates.
* Filters out involuntary departures (layoffs, firings, etc.), retaining only active employees and voluntary leavers. Involuntary exits are removed because they reflect employer-driven decisions rather than the personal choices we need to analyse when modelling voluntary attrition.
* Flags voluntary exits with a binary column, voluntary\_termination, set to 1 for voluntary departures and 0 otherwise.

The result is a clean, one-row-per-employee table with a clear voluntary-attrition indicator, ideal for reporting.

**A screenshot of a computer code

AI-generated content may be incorrect.**

**Performing analysis:**

1. **EDA using visualizations**: To uncover how voluntary attrition varies across key demographic and work-related factors, I built Python visualizations using Pandas for aggregation and Matplotlib/Seaborn for plotting. Each chart followed the two-step pattern:
2. Computing Attrition Rate
   * Group the cleaned dataset by a given feature (e.g. age, gender\_key, ethnicity, length\_of\_service, promotions, manager\_role, etc.).
   * Calculate the attrition rate for individual features.
   * Also capture raw counts of employees and voluntary exits for context.
3. Plotting Bar Charts
   * Dual‐bar plots showing total headcount vs. voluntary exits, so I can see both scale and turnover volume.
   * Single‐bar attrition‐rate plots positioned directly beneath, annotating each bar with its percentage value for immediate readability.
   * Consistent styling (horizontal axes for grouping variable, vertical for rates/counts, minimal gridlines, on‐bar labels) ensures that trends across demographics and work factors can be compared at a glance.

By charting each demographic (age, gender, ethnicity) and work factor (tenure, promotions, manager status, distance bands, etc.), these Python‐based EDA visuals translate raw numbers into clear, actionable insights, identifying where to prioritise retention efforts and which cohorts warrant deeper investigation.

A screenshot of a graph

AI-generated content may be incorrect.

A graph with red lines and numbers

AI-generated content may be incorrect.

Code for one such visualization ( tenure vs attrition rate):

import pandas as pd

import matplotlib.pyplot as plt

**# Attrition rate (%) at each length\_of\_service**

filtered\_df = df[df['length\_of\_service'] <= 43]

attrition\_by\_ls = ( filtered\_df .groupby('length\_of\_service')['voluntary\_termination'] .mean() .mul(100) .sort\_index())

**# DEFINE TENURE BANDS**

bins = [-1, 1, 3, 5, 10, df['length\_of\_service'].max()]

labels = ['<1 yr', '1–3 yrs', '3–5 yrs', '5–10 yrs', '10+ yrs']

df['tenure\_band'] = pd.cut(df['length\_of\_service'], bins=bins, labels=labels)

**# CALCULATE ATTRITION RATE BY TENURE BAND**

attrition\_by\_band = ( df.groupby('tenure\_band')['voluntary\_termination'].mean().mul(100).reindex(labels) # preserve order)

**# COUNT EMPLOYEES & EXITS BY TENURE BAND**

count\_and\_exits = (df.groupby('tenure\_band').agg(

employee\_count = ('employee\_ID', 'count'),

voluntary\_exits = ('voluntary\_termination', 'sum') ) .reindex(labels))

**# PLOT 1: Attrition Rate % by exact length\_of\_service**

plt.figure(figsize=(12,4))

plt.bar(attrition\_by\_ls.index, attrition\_by\_ls.values, color='teal')

plt.title('Attrition Rate % by length\_of\_service')

plt.xlabel('length\_of\_service (years)')

plt.ylabel('Attrition Rate %')

**# annotate bars**

for x, y in zip(attrition\_by\_ls.index, attrition\_by\_ls.values):

plt.text(x, y + 0.5, f'{y:.1f}%', ha='center', va='bottom', fontsize=8)

plt.tight\_layout()

plt.show()

**# PLOT 2: Attrition Rate % by Tenure Band**

plt.figure(figsize=(6,4))

plt.bar(attrition\_by\_band.index.astype(str), attrition\_by\_band.values, color='teal')

plt.title('Attrition Rate % by Tenure Band')

plt.xlabel('Tenure Band')

plt.ylabel('Attrition Rate %')

for x, y in zip(attrition\_by\_band.index.astype(str), attrition\_by\_band.values):

plt.text(x, y + 0.5, f'{y:.1f}%', ha='center', va='bottom', fontsize=9)

plt.tight\_layout()

plt.show()

**# PLOT 3: Counts vs Voluntary Exits by Tenure Band**

plt.figure(figsize=(6,4))

idx = range(len(labels))

width = 0.35

plt.bar(idx, count\_and\_exits['employee\_count'], width, label='Stayers', color='skyblue')

plt.bar([i+width for i in idx], count\_and\_exits['voluntary\_exits'], width, label='Voluntary Exits',color='darkblue')

plt.xticks([i+width/2 for i in idx], labels)

plt.title('Count of employee\_ID and Sum of voluntary\_termination by Tenure Band')

plt.ylabel('Count')

plt.legend()

# annotate

for i, (emp, vol) in enumerate(zip(count\_and\_exits['employee\_count'],

count\_and\_exits['voluntary\_exits'])):

plt.text(i, emp + 5, str(emp), ha='center', va='bottom', fontsize=8)

plt.text(i+width, vol + 5, str(vol), ha='center', va='bottom', fontsize=8)

plt.tight\_layout()

plt.show()

**2. Statistical analysis:**

T-test and Chi-square: To know which features of our workforce truly distinguish those who stay from those who choose to leave, we applied two complementary statistical tests. First, ran independent-samples t-tests on continuous measures (length of service, number of promotions, age, commute distance) to see whether the average value differs significantly between the “Stayed” and “Voluntarily Left” groups.

Second, we used chi-square tests of independence on each categorical variable (department, job role, gender, resource-group membership), to determine whether the proportion of employees in each category varies more than we would expect by chance. By combining mean-based comparisons for numeric data with distribution-based comparisons for categories, we can pinpoint which factors really matter for voluntary attrition.

A screenshot of a computer code

AI-generated content may be incorrect.

Results: A screenshot of a computer

AI-generated content may be incorrect.

Correlation analysis: I used correlation analysis to quantify the **strength and direction** of the linear relationship between each numeric feature and voluntary attrition. While our t-tests told us *which* drivers showed significant differences between “Stayed” and “Left” groups, correlation reveals *how strongly* those factors move in tandem with quit behavior.

By computing Pearson’s *r* for every continuous (and Boolean) variable against the binary attrition flag:

* A Single Ranked View: Easily see which predictors (e.g. tenure, promotions) have the largest absolute correlations, and which are essentially uncorrelated (e.g. age, top-talent).
* Directionality: Confirm negative or positive correlations
* Complement to Hypothesis Tests: T-tests and chi-square assess *statistical significance* of group differences; correlation tells us *practical significance*, how much variation in attrition can be linearly explained by each feature.
* Feature Prioritization for Modeling: Variables with near-zero correlation can be deprioritized or dropped when building predictive models, focusing instead on the handful of features that truly move the needle.
* The VIF results reveal which predictors carry redundant information (i.e. are highly collinear) and which stand relatively on their own

A screenshot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

3.Running Logistic Regression: It helps to get the impact of each feature and a clear ranking of what to focus on. Also, each feature gets an odds ratio, which tells by how much, exactly the drivers matter.

* I started with feature engineered bands (age\_band, tenure\_band, etc.)

A screenshot of a computer program

AI-generated content may be incorrect.

* **Train/Test Split** holds out 20 % of the data for final evaluation, train on the remaining 80 %.

A computer code with text

AI-generated content may be incorrect.

* ColumnTransformer isolates our categorical features (cat\_cols) for one-hot encoding, and drops everything else.
* OneHotEncoder(drop="first") creates 0/1 dummies but omits one level per variable (to avoid perfect multicollinearity).
* class\_weight="balanced" automatically up-weights the rare “quit” class so the model doesn’t ignore it.

**A computer screen shot of a code

AI-generated content may be incorrect.**

* **Training & Prediction**

**A screen shot of a computer code

AI-generated content may be incorrect.**

A screenshot of a computer screen

AI-generated content may be incorrect.

* We convert the logistic coefficients into ordered odds-ratios, this step produces two actionable lists:
  + Top risk factors: the employee attributes that most strongly increase quit-odds, exactly the groups we should prioritize for retention efforts.
  + Protective factors: the characteristics that most reduce quit-odds, those we can double-down on to keep employees engaged and on board.

A screenshot of a computer program

AI-generated content may be incorrect.

A graph of a logistic regression

AI-generated content may be incorrect.

* **Random Forest:** It captures non-linear effects and interactions automatically, and often gives higher predictive accuracy than a single linear model.
* Like the logistic model, we split/test and fit the model, evaluate Gini drivers for feature importance, along with multiple scores

A screenshot of a computer program

AI-generated content may be incorrect.

A graph with text and numbers

AI-generated content may be incorrect.

**Problem Statement 2**: The company wants to understand what some attributes are that employees share amongst themselves, and that information can help the company design its talent program that are targeted to the different groups.

● Please help the talent team segment the firm's total employee base into smaller segments for a targeted talent strategy.

● Prioritize one of these targeted segments and identify 2-3 unique insights and propose actions that a Talent Partner can take to more effectively engage this group than the general population

Methodology:

Since nearly every driver is categorical (binned tenure, age-band, promotion, ERG membership, manager\_role, top\_talent, gender, ethnicity, marital\_status), K-modes is a natural distance = simple matching clustering that handles arbitrary categories without forcing dummy-expansion or Euclidean distortions.

* **Data preparation**

A screenshot of a computer

AI-generated content may be incorrect.

* **Optimal K search**

A screenshot of a computer program

AI-generated content may be incorrect.

A graph with a line

AI-generated content may be incorrect.

A screen shot of a computer code

AI-generated content may be incorrect.

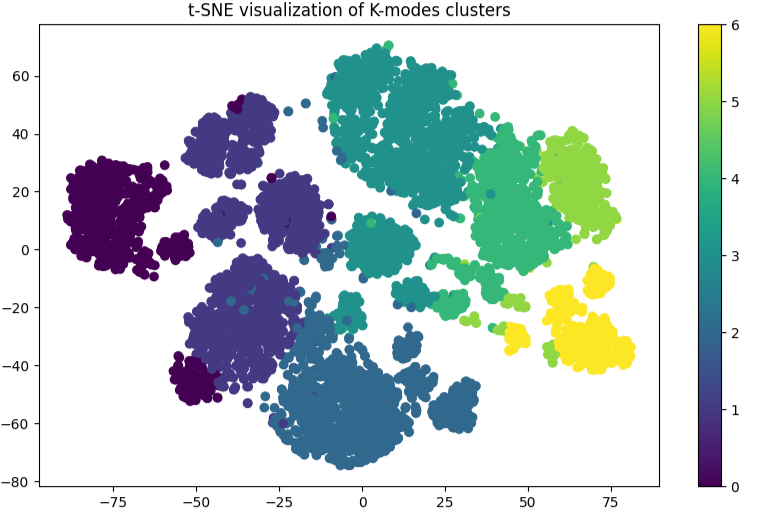
* Fit K-modes & attach segment labels: From the elbow method, we tried k=4 and 7, 7 showed more clear clusters, hence, it was chosen. The two built-in options in the kmodes package are:
  + **Cao** – tries to pick high-density points first (good for very skewed data)
  + **Huang** – picks initial modes by sampling categories in proportion to their marginal frequencies.

In practice, **Huang** often gives a stable starting point for purely categorical data, and so is a common default.

* **Visualizing Clusters using tSNE**: t-Distributed Stochastic Neighbor Embedding (t-SNE) is a non-linear dimension-reduction algorithm designed to preserve local neighborhood structure. By projecting high-dimensional one-hot matrix down to **2 dimensions**, we get a scatterplot where points that are “close” (i.e. share most categories) stay close in the plot.

A screen shot of a computer code

AI-generated content may be incorrect.



*Note: t-SNE visualisation of the 7 k-modes clusters. While this demonstrates clean cluster separation, relative distance between clusters is not meaningful due to dimensionality reduction*

* **Performed EDA on the different clusters for Profiling:**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer screen

AI-generated content may be incorrect.**