Employee Exit Survey Analysis

Project Report

1. Project Objective

To analyze employee exit survey data, find key reasons for employee departures, and use statistical & predictive methods to help HR improve retention.

You're basically:

- Cleaning HR survey data,
- Exploring trends (EDA),
- Performing statistical tests,
- And building a model to predict voluntary vs. involuntary exits.

2. Import Libraries & Load Dataset

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import chi2_contingency, f_oneway
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, roc_auc_score
```

Explanation:

- `pandas` & `numpy`: for data manipulation and cleaning
- `matplotlib` & `seaborn`: for visualization
- `scipy.stats`: for statistical tests (chi-square & ANOVA)
- `sklearn`: for splitting data, encoding, and building a machine learning model

Then:

```
df = pd.read_csv("employee_exit_survey.csv")
df.head()
```

Loads your dataset into a DataFrame for analysis.

3. Data Cleaning and Preprocessing

Here you handled missing values, data types, and new feature creation.

```
num_cols = ['satisfaction_score','manager_rating','last_promotion_years_ago']
for c in num_cols:
    df[c] = (
        df.groupby('department', observed=False)[c]
        .transform(lambda x: x.fillna(x.median()))
    )
    df[c] = df[c].fillna(df[c].median())
```

What this does:

- For each numeric column:
  - It fills missing values using the **median** of that column **within each department** (because satisfaction or ratings can differ by department).
  - If there's still any missing value left, it fills with the overall median.

Then:

```
df['department'] = df['department'].astype('category')
df['reason_for_leaving'] = df['reason_for_leaving'].astype('category')
```

Converts those columns to **categorical type** (more efficient, and helpful for modeling).

Derived Features :

```
df['is_senior_tenure'] = (df['tenure_years'] >= 5).astype(int)
df['exit_date'] = pd.to_datetime(df['exit_date'])
df['exit_month'] = df['exit_date'].dt.to_period('M').astype(str)
```

You're adding:

- `is_senior_tenure`: a binary feature showing if an employee worked ≥ 5 years
- `exit_month`: the month they left — useful for trend analysis

4. Exploratory Data Analysis (EDA)

(a) Count of reasons for leaving

```
sns.countplot(y='reason_for_leaving', data=df,
order=df['reason_for_leaving'].value_counts().index)
```

→ Visualizes **which reasons appear most often** — "Better pay", "Career growth", etc.

(b) Reasons by department

```
sns.countplot(x='department', data=df, hue='reason_for_leaving')
```

→ Shows which departments are losing employees to which reasons.
For example, if Sales has many "Better pay" exits — that's an actionable insight.

(c) Satisfaction vs Tenure

```
sns.scatterplot(x='tenure_years', y='satisfaction_score', data=df, hue='department')
```

→ Helps you see if long-tenured employees are generally happier or not.

5. Statistical Analysis

This step gives HR **statistical evidence** behind patterns.

(a) Chi-Square Test

```
ct = pd.crosstab(df['department'], df['reason_for_leaving'])
chi2, p, dof, _ = chi2_contingency(ct)
```

Purpose:
To test if **department** and **reason for leaving** are related.

- If **p < 0.05**, the relationship is **statistically significant**.
  → Meaning exit reasons depend on the department.

(b) ANOVA Test

```
groups = [group['satisfaction_score'].values for _, group in
df.groupby('reason_for_leaving') if len(group) >= 3]
f, p_anova = f_oneway(*groups)
```

Purpose:
Checks if **average satisfaction scores differ** between reasons for leaving.

- If **p < 0.05**, at least one group's mean satisfaction is significantly different.
  → For instance, employees leaving due to "Work-life balance" might have much lower satisfaction.

6. Predictive Modeling

Now you're using ML to predict **who is likely to leave voluntarily** (resigned) vs **involuntarily** (fired, terminated).

(a) Target Variable :

```
df['voluntary'] = df['reason_for_leaving'].apply(lambda r: 0 if 'termination' in
r.lower() else 1)
```

- 1 → Voluntary (employee chose to leave)
- 0 → Involuntary (company terminated)

(b) Feature Engineering :

```
X = df[['tenure_years','satisfaction_score','manager_rating','is_senior_tenure']]
```

These are your **numeric predictors**.

Then, categorical encoding:

```
ohe = OneHotEncoder(sparse_output=False, drop='first')
dept_encoded = ohe.fit_transform(df[['department']])
dept_df = pd.DataFrame(dept_encoded, columns=[f"dept_{c}" for c in
ohe.categories_[0][1:]])
X = pd.concat([X.reset_index(drop=True), dept_df.reset_index(drop=True)], axis=1)
```

Explanation:

- `OneHotEncoder` turns "department" (e.g., HR, Sales, Engineering) into dummy variables (`dept_Sales`, `dept_Engineering`, etc.)
- `drop='first'` avoids multicollinearity (so one category becomes the baseline).

Finally, you define:

```
y = df['voluntary']
```

(c) Train-Test Split :

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,
random_state=42)
```

- 75% of data → training
- 25% → testing
- `random_state=42` ensures reproducibility

(d) Model Training :

```
model = LogisticRegression(max_iter=500)
model.fit(X_train, y_train)
```

You're using **Logistic Regression**, ideal for binary classification (0 or 1).

(e) Evaluation :

```
y_pred = model.predict(X_test)
y_proba = model.predict_proba(X_test)[:,1]

print(classification_report(y_test, y_pred))
print("ROC AUC:", roc_auc_score(y_test, y_proba))
```

Outputs:

- **Precision / Recall / F1-score:** Model's performance for predicting voluntary vs. involuntary exits.
- **ROC AUC:** How well the model separates the two classes (higher = better).

(f) Feature Importance :

```
coef_df = pd.DataFrame({'feature': X.columns, 'coef':
model.coef_[0]}).sort_values(by='coef', key=abs, ascending=False)
sns.barplot(x='coef', y='feature', data=coef_df)
```

Shows:
Which features (e.g., satisfaction, manager rating, department) influence voluntary exit probability most strongly.

7. Insights & Recommendations

This is the business interpretation section.

Example findings you might note:

- Most exits were due to **Better pay** or **Work-life balance**.
- **Engineering** and **Sales** have the highest voluntary turnover.
- **Satisfaction score** and **Manager rating** are strong predictors of voluntary exits.
- **Statistical tests** confirm department and reason for exit are significantly related.

Recommendations:

- Review **compensation policies** for high-turnover departments.
- Introduce **career growth programs**.
- Improve **manager engagement and feedback systems**.
- Promote **flexible work schedules**.

Summary of Key Learnings

Step	Skill	What You Did
Data Cleaning	pandas	Filled missing data smartly by department
EDA	seaborn	Found top exit reasons and trends
Statistics	scipy	Proved relationships with significance tests
ML	scikit-learn	Predicted voluntary exits using logistic regression
Insights	HR Analytics	Converted data into actionable business advice