Attrition Intelligence - Predictive Analytics & Exploratory Data Analysis

Human Resources are critical resources of any organiazation. Organizations spend huge amount of time and money to hire and nuture their employees. It is a huge loss for companies if employees leave, especially the key resources. So if HR can predict whether employees are at risk for leaving the company, it will allow them to identify the attrition risks and help understand and provie necessary support to retain those employees or do preventive hiring to minimize the impact to the orgranization.

Employee Data Attributes

Welcome to the Employee Data Attributes guide! Below is a comprehensive list of the key attributes included in our employee dataset.

Basic Information:

- **EmployeeId**: Unique identifier for each employee.
- **JoiningDate**: Date when the employee joined the company.
- **ProfileName**: Job profile or designation of the employee.
- **BirthDate**: Date of birth of the employee.
- **DepartmentName**: Department in which the employee works.
- **GradeCode**: Grade code or level of the employee.
- **Gender**: Gender of the employee.

Educational Background:

- **PassingYear**: Year of passing highest qualification.
- **QualificationName**: Highest qualification attained by the employee.
- **SpecializationName**: Specialization or major field of study.
- **CollegeName**: Name of the college/university attended by the employee.

Employment Details:

- **LastWorkingDate**: Date when the employee left the company (if applicable).
- **Status**: Employment status (e.g., Active, Inactive, Resigned).

Geographic Information:

- **Region**: Geographic region where the employee is based.

Performance Metrics:

- **Overall LQ Score**: Learning Quotient (LQ) score, indicating the employee's learning ability.

Sample Data Entry:

Below is a sample entry from the dataset, showcasing how the attributes are structured:

EmployeeId	JoiningDate	ProfileName	BirthDate	DepartmentName	GradeCode	Gender	PassingYear	QualificationName	SpecializationName	CollegeName	LastWorkingDate	Status	Region	Overall LQ Score
0	06-11-2021	S-CSM	23-08-1994	Transaction	G3	Male	2020	Master of Commerce (M.Com.)	Commerce		14-02-2024	Inactive	West-2	4.5

Feel free to refer back to this guide whenever you need a detailed understanding of our employee dataset!

Note: LQ Score represents the Learning Quotient score, which is a measure of the employee's ability to learn and adapt.

```python
# Importing essential libraries
import pandas as pd  # For data manipulation and analysis
import numpy as np  # For numerical operations
import matplotlib.pyplot as plt  # For plotting and data visualization
import seaborn as sns  # For statistical data visualization
from datetime import datetime  # For handling date and time operations

# Step 1: Load the data from CSV file
data_path = r"C:\Users\ASUS\Downloads\Motilal Predictive Analytics\
EmpData_MotilalOswal_HRAnalytics.csv"
df = pd.read_csv(data_path)
```

Step 2: Data Overview

Why Data Overview?

Before diving into analysis or modeling, it's crucial to have a comprehensive understanding of the dataset. Conducting a data overview helps us gain insights into the structure, content, and quality of the data.

Data Overview:

Displaying the First Few Rows:

We begin by examining the first few rows of the dataset to get a glimpse of the data's structure and content. This allows us to understand the format of the data and identify any potential issues with data types or missing values.

```
# Step 2: Data Overview
print("Data Overview:")
print(df.head())
print(f"Shape: {df.shape}")
print(df.info())
```

```
Data Overview:
   EmployeeId JoiningDate ProfileName   BirthDate DepartmentName
GradeCode  \
0           0  06-11-2021       S-CSM  23-08-1994    Transaction
G3
1           1  15-06-2013         CSO  23-06-1985    CSM Affairs
G2
2           2  03-03-2014         CSO  09-04-1981    CSM Affairs
G2
3           3  25-05-2014         CSO  20-05-1978    CSM Affairs
G2
4           4  12-11-2017         CSO  24-07-1994    Transaction
G2

  Gender  PassingYear            QualificationName SpecializationName
\
0   Male       2020.0  Master of Commerce (M.Com.)           Commerce

1   Male       2008.0                          NaN                NaN

2   Male       2018.0                          HSC            Science

3   Male       2015.0  Master of Commerce (M.Com.)           Commerce

4   Male       2015.0                       Others            Science


                    CollegeName LastWorkingDate    Status   Region  \
0                           NaN      14-02-2024  Inactive  West-2
1  Ramkrishna More Collage pune      21-06-2016  Inactive     NaN
2                           NaN      02-07-2016  Inactive  West-1
3                           NaN      30-07-2016  Inactive    South
4                           NaN      29-01-2020  Inactive    South

   Overall LQ Score
0               4.5
```

```
1               NaN
2               3.0
3               3.0
4               3.0
Shape: (9851, 15)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9851 entries, 0 to 9850
Data columns (total 15 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   EmployeeId          9851 non-null   int64
 1   JoiningDate         9851 non-null   object
 2   ProfileName         9851 non-null   object
 3   BirthDate           9851 non-null   object
 4   DepartmentName      9851 non-null   object
 5   GradeCode           9851 non-null   object
 6   Gender              9851 non-null   object
 7   PassingYear         9754 non-null   float64
 8   QualificationName   9719 non-null   object
 9   SpecializationName  8985 non-null   object
 10  CollegeName         1064 non-null   object
 11  LastWorkingDate     8215 non-null   object
 12  Status              9851 non-null   object
 13  Region              9485 non-null   object
 14  Overall LQ Score    8512 non-null   float64
dtypes: float64(2), int64(1), object(12)
memory usage: 1.1+ MB
None
```

```python
# Step 3: Missing Values
print("\nMissing Values:")
print(df.isnull().sum())
```

```
Missing Values:
EmployeeId              0
JoiningDate             0
ProfileName             0
BirthDate               0
DepartmentName          0
GradeCode               0
Gender                  0
PassingYear            97
QualificationName     132
SpecializationName    866
CollegeName          8787
LastWorkingDate      1636
Status                  0
Region                366
```

```
Overall LQ Score       1339
dtype: int64

# Step 4: Numerical Data Summary
numerical_cols = df.select_dtypes(include=[np.number])
print("\nNumerical Data Summary:")
print(numerical_cols.describe())


Numerical Data Summary:
        EmployeeId  PassingYear  Overall LQ Score
count  9851.000000  9754.000000       8512.000000
mean   4925.000000  2013.600677          3.122357
std    2843.883085     5.154574          1.317912
min       0.000000  1990.000000          0.000000
25%    2462.500000  2011.000000          3.000000
50%    4925.000000  2014.000000          3.000000
75%    7387.500000  2017.000000          3.000000
max    9850.000000  2023.000000         14.000000
```

Step 5: Categorical Data Exploration

Why Categorical Data Exploration?

Categorical variables play a crucial role in data analysis as they provide insights into different categories or groups within the dataset. Exploring categorical data helps us understand the distribution and frequencies of various categories within each variable, which is essential for gaining insights and making informed decisions.

Categorical Data Exploration:

Extracting Categorical Columns:

We begin by identifying the categorical columns in the dataset using their data types. Categorical columns contain discrete values representing different categories or groups.

```
# Step 5: Categorical Data Exploration
categorical_cols = df.select_dtypes(include=[object])
print("\nCategorical Data Exploration:")
for col in categorical_cols:
    print(f"\nUnique values in '{col}':")
    print(df[col].value_counts(dropna=False))


Categorical Data Exploration:

Unique values in 'JoiningDate':
```

```
JoiningDate
28-02-2022    74
31-10-2021    56
30-12-2021    53
03-12-2017    40
29-03-2022    32
              ..
17-10-2015     1
21-01-2015     1
28-02-2014     1
31-05-2017     1
13-02-2015     1
Name: count, Length: 1668, dtype: int64

Unique values in 'ProfileName':
ProfileName
CSM       6137
CSO       2799
S-CSM      909
CSM-CF       6
Name: count, dtype: int64

Unique values in 'BirthDate':
BirthDate
10-05-1989    18
1898-01-01    17
26-04-1990    15
10-08-1985    15
01-01-1988    14
              ..
28-12-1992     1
12-02-1994     1
10-11-2000     1
15-04-1995     1
11-04-1984     1
Name: count, Length: 3210, dtype: int64

Unique values in 'DepartmentName':
DepartmentName
Transaction    7707
CSM Affairs    2144
Name: count, dtype: int64

Unique values in 'GradeCode':
GradeCode
G3    6199
G2    2977
G4     669
G7       4
G5       2
```

```
Name: count, dtype: int64

Unique values in 'Gender':
Gender
Male      9624
Female     227
Name: count, dtype: int64

Unique values in 'QualificationName':
QualificationName
Bachelor of Arts (B.A)                                  2026
Bachelor of Commerce (B.Com.)                           1961
HSC                                                     1879
MBA/PGDM                                                 763
Bachelor of Science (B.Sc.)                              654
Bachelor of Business Administration (B.B.A.)             289
Others                                                   285
Master of Commerce (M.Com.)                              280
B.Tech/B.E.                                              278
Master of Arts (M.A.)                                    241
Bachelor of Computer Applications (B.C.A.)               231
SSC                                                      227
Diploma                                                  165
NaN                                                      132
Master of Science (M.Sc.)                                 65
Diploma (12+3 yrs)                                        55
Master Of Business Administration                         41
Bachelor of Education (B.Ed.)                             40
Master of Computer Applications (M.C.A.)                  39
Master of Social Work (MSW)                               32
Bachelor of Management Studies (B.M.S.)                   30
Bachelor In Business Management (BBM)                     22
Bachelor of Engineering                                   18
PG Diploma                                                10
LLB                                                       10
Bachelors in Accounting & Finance                          7
Bachelor of Hotel Management (B.H.M.)                      6
Diploma in Engineering                                     6
Bachelor of Science Computer Application (H)               5
Bachelor’s Degree In Tourism Studies                       4
Master of Management Studies (M.M.S.)                      4
Bachelor of Architecture (B.Arch.)                         4
Master of Law (L.L.M.)                                     4
Bachelor of Literature (B.Lit)                             3
Bachelor Of Corp Secretaryship                             3
M Tech-M.E.                                                3
Master of Communication Studies                            3
Integrated PG Course                                       3
Ministry of Human Resource Development (MHRD)              3
```

```
Bachelor of Pharmacy (B.Pharma.)                                3
Master in Marketing Management (MMM)                            2
Diploma in General Nursing and Midwifery                        2
Bachelor of Information Technology (BIT)                        2
DIPLOMA IN ELECTRONICS & COMMUNICATION ENGINEERING              2
Bachelor of Law (LLB)                                           1
Master of Philosophy                                            1
Post Graduate Diploma in Management                             1
Bachelor of physical education                                  1
Bachelor of Mass Media (B.M.M.)                                 1
Bachelor of Business Studies(BBS)                               1
PG diploma in Banking                                           1
BAF                                                             1
6Sigma                                                          1
Name: count, dtype: int64

Unique values in 'SpecializationName':
SpecializationName
Commerce                 1075
Accounting & Finance     1036
Arts                      972
NaN                       866
Arts&Humanities           704
                         ...
FINAL                       1
Banking                     1
MASTER OF PHILOSOPHY        1
Bachelor of Law (LLB)       1
Finance-PGDM                1
Name: count, Length: 116, dtype: int64

Unique values in 'CollegeName':
CollegeName
NaN                                        8787
KURUKSHETRA UNIVERSITY,KURUKSHETRA           20
SIDDARTHA F G C BIDAR                         8
MADRAS UNIVERSITY                             8
ALAGAPPA UNIVERSITY                           7
                                           ...
Hirachand Nemcahnd College of Commerce        1
Lohiya Collage                                1
B.S.PATIL COLLEGE                             1
SWAMI VIVEKANANDA 1ST GRADE COLLAGE           1
SPDM College Shirpur                          1
Name: count, Length: 457, dtype: int64

Unique values in 'LastWorkingDate':
LastWorkingDate
NaN           1636
06-10-2023      39
```

```
09-04-2024      38
01-04-2024      35
04-03-2024      34
              ...
15-11-2018       1
30-12-2016       1
31-01-2016       1
07-03-2017       1
20-09-2018       1
Name: count, Length: 1562, dtype: int64

Unique values in 'Status':
Status
Inactive    8276
Active      1435
Resigned     140
Name: count, dtype: int64

Unique values in 'Region':
Region
West-1    3029
South     2664
West-2    2447
North      668
Mumbai     508
NaN        366
East       167
HO           2
Name: count, dtype: int64
```

Step 6: Date Handling

Why Date Handling?

Dates are often crucial in data analysis, providing valuable insights into temporal patterns and trends. Date handling involves converting date columns to a consistent format and extracting meaningful features such as tenure and age. This step ensures that we can leverage temporal information effectively in our analysis and modeling.

Date Handling:

Converting Date Columns:

We start by converting the date columns in the dataset to a standardized format using the `pd.to_datetime()` function. This ensures uniformity in date representations and facilitates further analysis based on temporal information.

```python
# Step 6: Date Handling
date_columns = ['JoiningDate', 'BirthDate', 'LastWorkingDate']
for col in date_columns:
    df[col] = pd.to_datetime(df[col], format='%d-%m-%Y', 
errors='coerce')

# Calculate Tenure and Age
df['Tenure'] = (df['LastWorkingDate'].fillna(datetime.now()) - 
df['JoiningDate']).dt.days / 365
df['Age'] = (datetime.now() - df['BirthDate']).dt.days / 365

# Step 7: Handle Missing Values
df['QualificationName'].fillna('Unknown', inplace=True)
```

```
C:\Users\ASUS\AppData\Local\Temp\ipykernel_17232\1240223219.py:2: 
FutureWarning: A value is trying to be set on a copy of a DataFrame or 
Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never 
work because the intermediate object on which we are setting values 
always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try 
using 'df.method({col: value}, inplace=True)' or df[col] = 
df[col].method(value) instead, to perform the operation inplace on the 
original object.


  df['QualificationName'].fillna('Unknown', inplace=True)
```

Step 8: Encode Categorical Variables

Why Encode Categorical Variables?

In many machine learning algorithms, categorical variables need to be converted into numerical representations for model training. Categorical encoding transforms categorical variables into a format that can be understood by machine learning models, enabling them to effectively learn from the data.

Categorical Variable Encoding:

Selecting Categorical Columns:

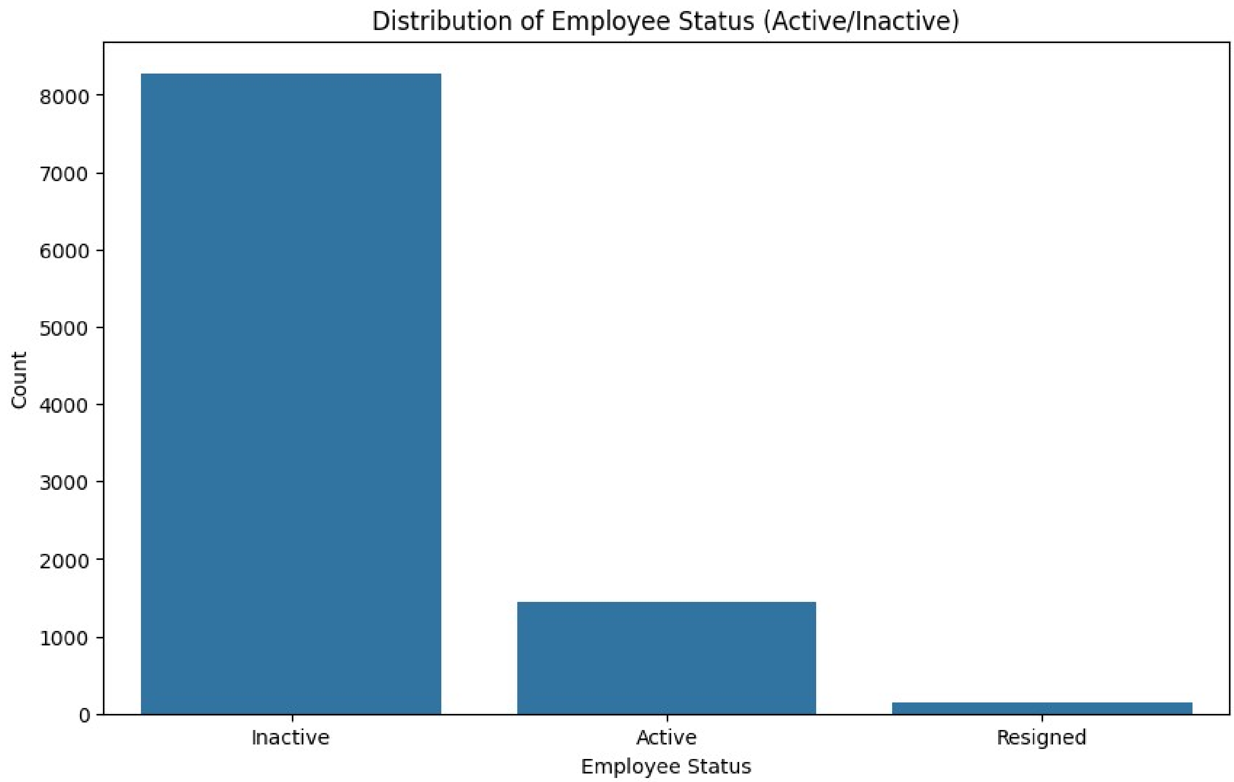
We begin by identifying the categorical columns in the dataset that need to be encoded into numerical representations. These columns represent categorical attributes such as job profiles, departments, grades, genders, qualifications, specializations, colleges, and regions.

```
# Step 8: Encode Categorical Variables
categorical_cols = ['ProfileName', 'DepartmentName', 'GradeCode',
'Gender',
                   'QualificationName', 'SpecializationName',
'CollegeName', 'Region']
df_encoded = pd.get_dummies(df, columns=categorical_cols,
drop_first=True)

# Convert Status to binary (Attrition = 1 if Inactive)
df_encoded['Attrition'] = df['Status'].apply(lambda x: 1 if x ==
'Inactive' else 0)
```

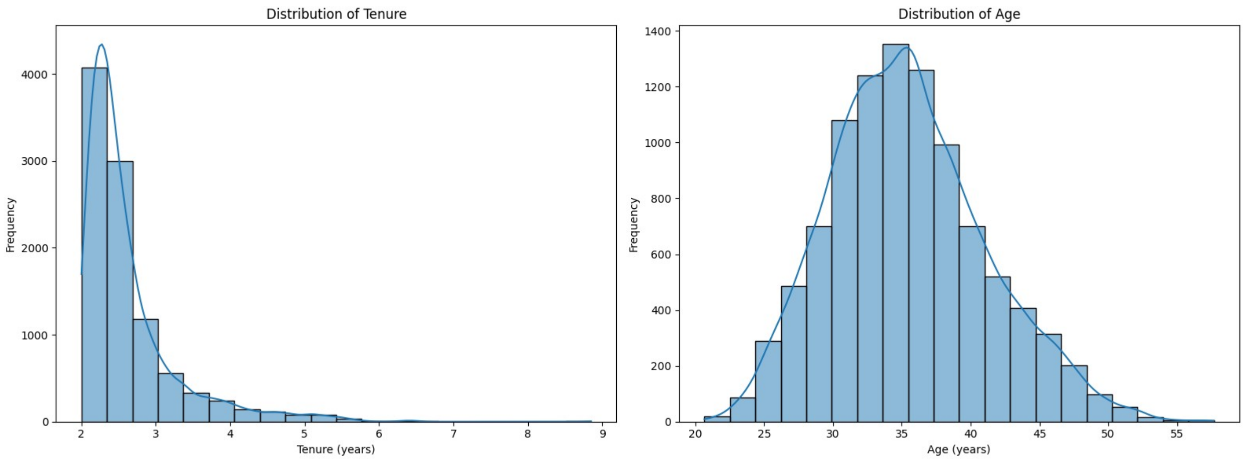
We use one-hot encoding, a popular technique for categorical encoding, to convert categorical variables into binary vectors. Each category within a categorical variable is represented by a binary feature, with a value of 1 indicating the presence of the category and 0 indicating absence

We convert the 'Status' column to a binary variable representing attrition, where 1 indicates that the employee is inactive and 0 indicates active status. This transformation enables us to create a target variable for predictive modeling tasks related to employee attrition prediction

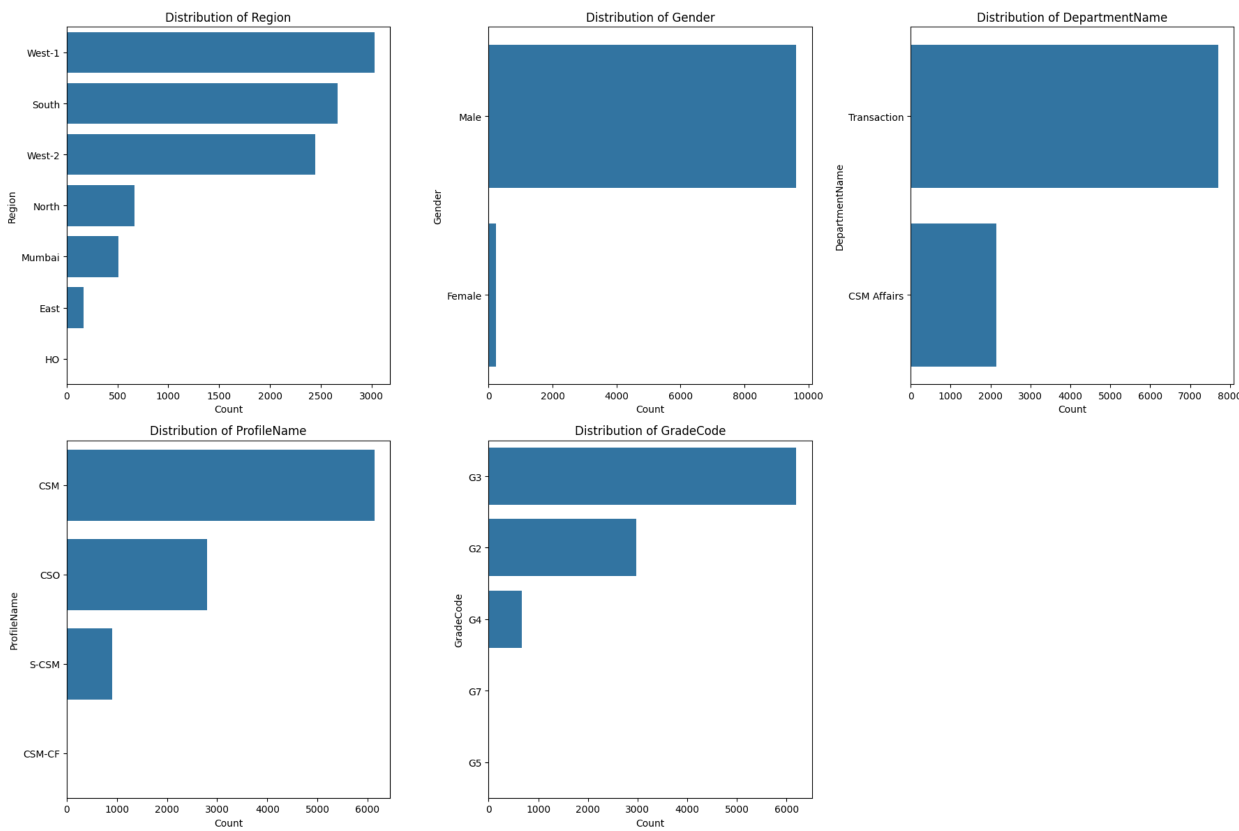
```
# Step 9: Visualizations
# Distribution of Attrition
plt.figure(figsize=(10, 6))
sns.countplot(x='Status', data=df)
plt.xlabel('Employee Status')
plt.ylabel('Count')
plt.title('Distribution of Employee Status (Active/Inactive)')
plt.show()
```




```python
# Distribution of Tenure and Age
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(16, 6))

sns.histplot(df['Tenure'].dropna(), bins=20, kde=True, ax=ax1)
ax1.set_xlabel('Tenure (years)')
ax1.set_ylabel('Frequency')
ax1.set_title('Distribution of Tenure')

sns.histplot(df['Age'].dropna(), bins=20, kde=True, ax=ax2)
ax2.set_xlabel('Age (years)')
ax2.set_ylabel('Frequency')
ax2.set_title('Distribution of Age')

plt.tight_layout()
plt.show()
```




```python
# Step 3: Distribution of Employees Over Categorical Attributes

# List of categorical columns to plot
categorical_columns = ['Region', 'Gender', 'DepartmentName',
'ProfileName', 'GradeCode']

# Create subplots for each categorical column
fig, axes = plt.subplots(2, 3, figsize=(18, 12))
axes = axes.flatten()

for i, col in enumerate(categorical_columns):
    sns.countplot(y=col, data=df, ax=axes[i],
order=df[col].value_counts().index)
    axes[i].set_title(f'Distribution of {col}')
    axes[i].set_xlabel('Count')
    axes[i].set_ylabel(col)

# Remove the extra subplot
fig.delaxes(axes[-1])

plt.tight_layout()
plt.show()
```



Step 10: Correlation Analysis

Why Correlation Analysis?

Correlation analysis is a crucial step in understanding the relationships between variables in a dataset. It helps identify patterns, dependencies, and associations between different attributes, providing insights into how changes in one variable may affect another. In the context of employee data, correlation analysis can reveal relationships between various factors such as job performance, demographics, and attrition.

Correlation Analysis:

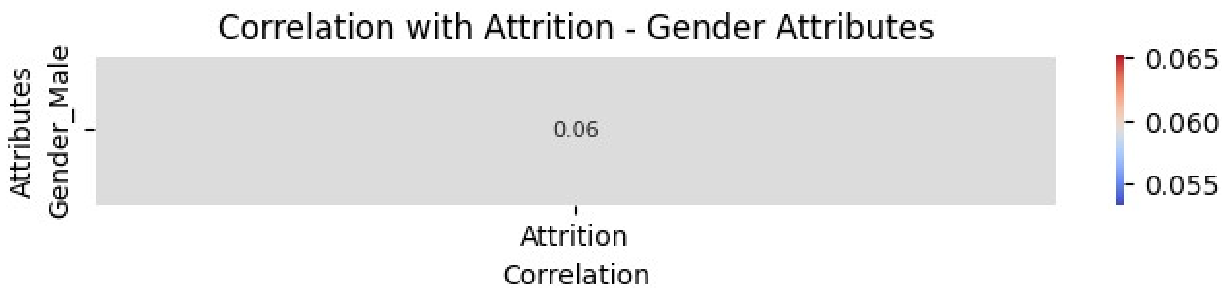
Calculating the Correlation Matrix:

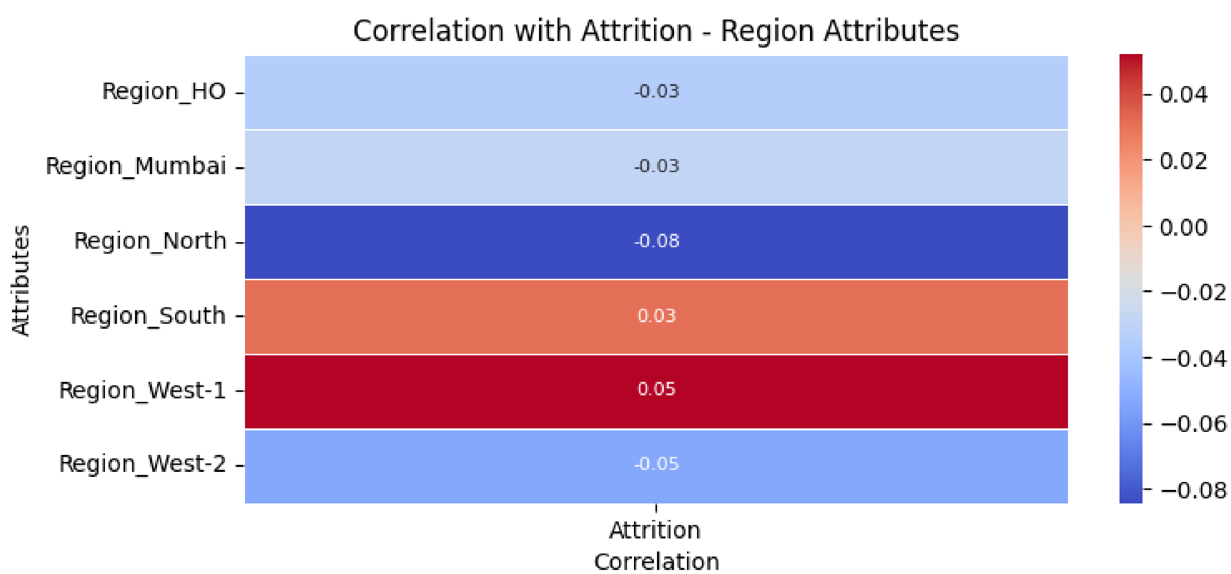
We compute the correlation matrix to quantify the strength and direction of linear relationships between numerical variables in the dataset. The correlation matrix provides a comprehensive overview of pairwise correlations between all pairs of numerical attributes.

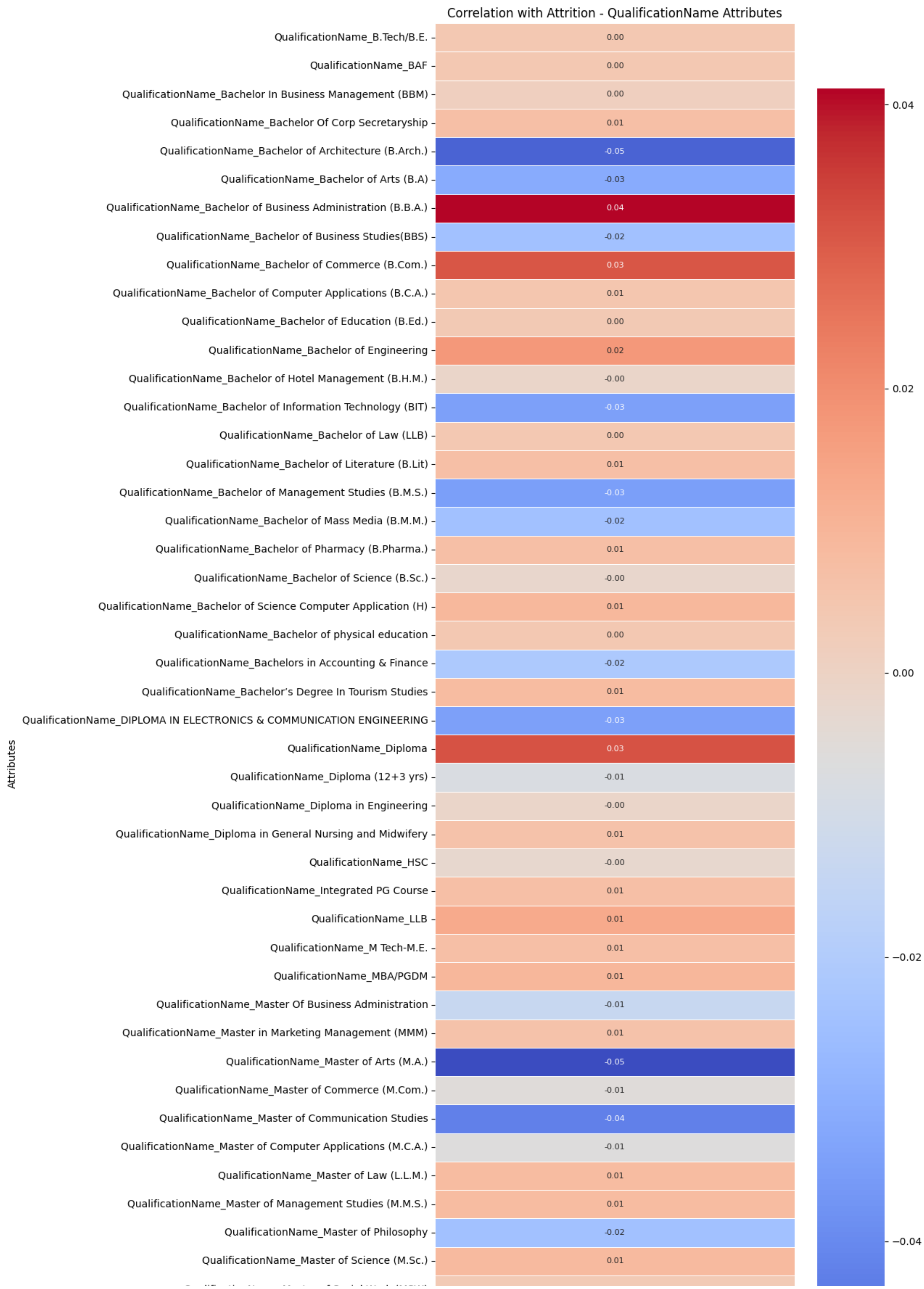
```
# Step 10: Correlation Analysis
# Calculate the correlation matrix
correlation_matrix = df_encoded.corr(numeric_only=True)
```

```
# List of categorical columns
categorical_columns = ['Gender', 'Region', 'QualificationName',
'CollegeName', 'SpecializationName',
                       'DepartmentName', 'GradeCode', 'ProfileName']

# Perform one-hot encoding on categorical columns
df_encoded = pd.get_dummies(df, columns=categorical_columns,
drop_first=True)

# Group columns based on original categorical columns
group_mapping = {}
for col in categorical_columns:
    group_mapping[col] = [c for c in df_encoded.columns if
c.startswith(col)]

# List of categorical columns
categorical_columns = ['Gender', 'Region', 'QualificationName',
'CollegeName', 'SpecializationName',
                       'DepartmentName', 'GradeCode', 'ProfileName']

# Perform one-hot encoding on categorical columns
df_encoded = pd.get_dummies(df, columns=categorical_columns,
drop_first=True)

# Create the Attrition column based on Status values
df_encoded['Attrition'] = df['Status'].apply(lambda x: 1 if x in
['Inactive', 'Resigned'] else 0)

# Group columns based on original categorical columns
group_mapping = {}
for col in categorical_columns:
    group_mapping[col] = [c for c in df_encoded.columns if
c.startswith(col)]

# Function to calculate and plot the correlation matrix for each group
with Attrition
def plot_group_correlations(df, group_mapping,
target_column='Attrition'):
    for group_name, columns in group_mapping.items():
        # Filter columns that are present in the dataframe
        relevant_columns = [col for col in columns if col in
df.columns]
        # Add the target column to the list of columns
        relevant_columns.append(target_column)

        # Calculate the correlation matrix
        correlation_matrix = df[relevant_columns].corr()

        # Extract the correlation with the target column
        correlation_with_target =
```

```
correlation_matrix[[target_column]].drop(target_column)

        # Plot the correlation matrix
        plt.figure(figsize=(8, len(relevant_columns) * 0.5))
        sns.heatmap(correlation_with_target, annot=True,
cmap='coolwarm', fmt='.2f', cbar=True, linewidths=0.5,
annot_kws={"size": 8})
        plt.title(f'Correlation with {target_column} - {group_name}
Attributes')
        plt.xlabel('Correlation')
        plt.ylabel('Attributes')
        plt.show()

# Call the function with the encoded DataFrame and group mapping
plot_group_correlations(df_encoded, group_mapping)
```

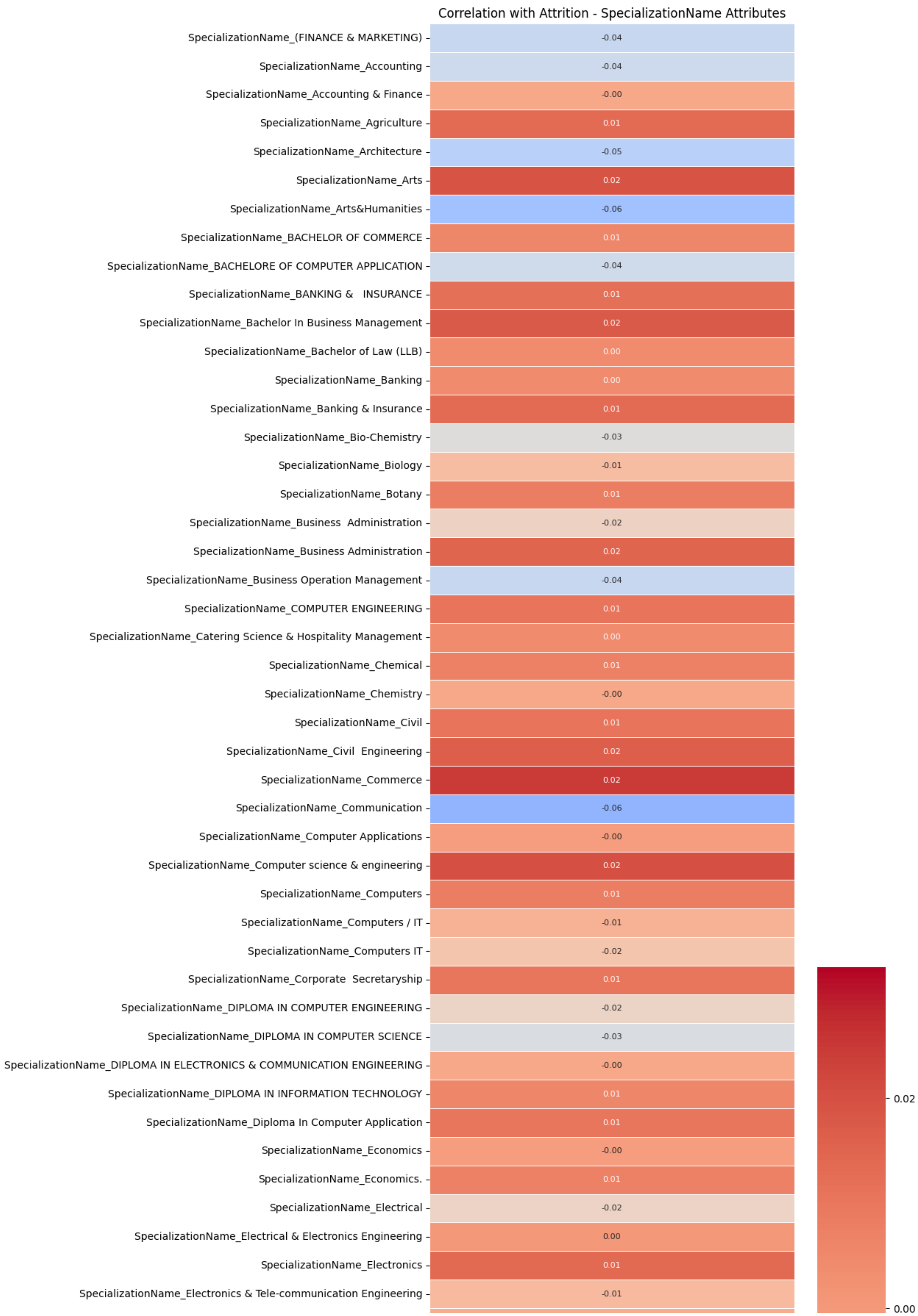


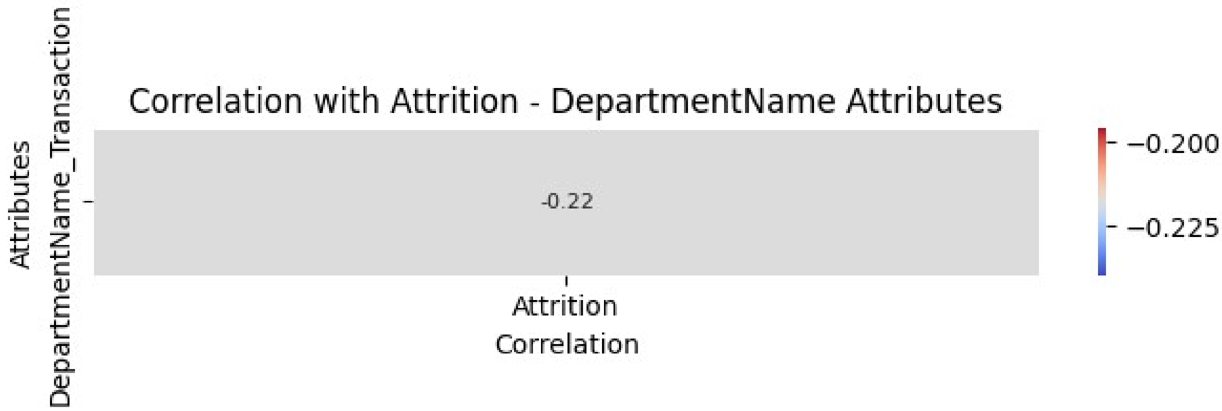


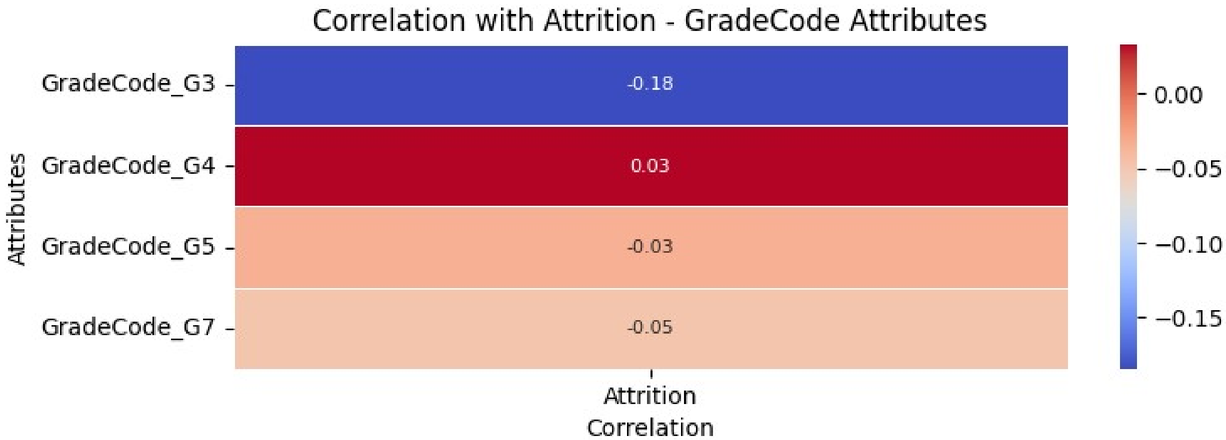


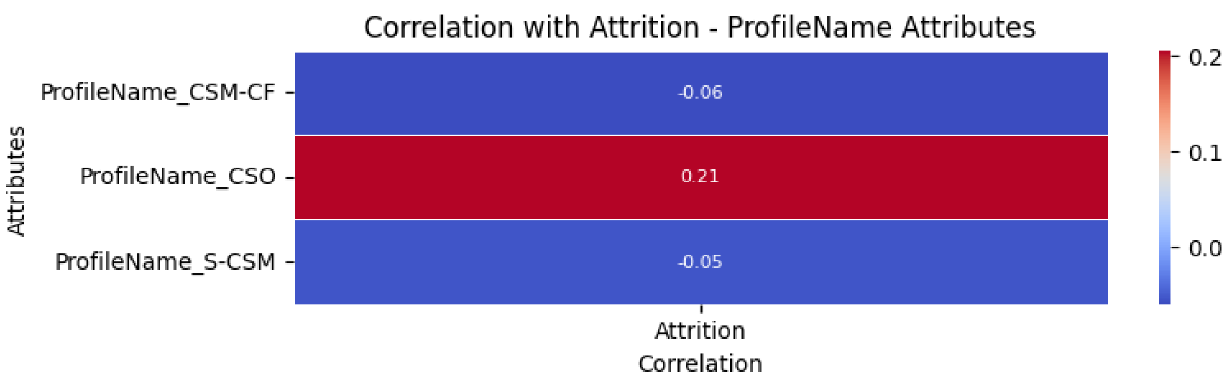
Correlation with Attrition - QualificationName Attributes
Attributes
QualificationName_B.Tech/B.E.
0.00
QualificationName_BAF
0.00
QualificationName_Bachelor In Business Management (BBM)
0.00
QualificationName_Bachelor Of Corp Secretaryship
0.01
QualificationName_Bachelor of Architecture (B.Arch.)
-0.05
QualificationName_Bachelor of Arts (B.A)
-0.03
QualificationName_Bachelor of Business Administration (B.B.A.)
0.04
QualificationName_Bachelor of Business Studies(BBS)
-0.02
QualificationName_Bachelor of Commerce (B.Com.)
0.03
QualificationName_Bachelor of Computer Applications (B.C.A.)
0.01
QualificationName_Bachelor of Education (B.Ed.)
0.00
QualificationName_Bachelor of Engineering
0.02
QualificationName_Bachelor of Hotel Management (B.H.M.)
-0.00
QualificationName_Bachelor of Information Technology (BIT)
-0.03
QualificationName_Bachelor of Law (LLB)
0.00
QualificationName_Bachelor of Literature (B.Lit)
0.01
QualificationName_Bachelor of Management Studies (B.M.S.)
-0.03
QualificationName_Bachelor of Mass Media (B.M.M.)
-0.02
QualificationName_Bachelor of Pharmacy (B.Pharma.)
0.01
QualificationName_Bachelor of Science (B.Sc.)
-0.00
QualificationName_Bachelor of Science Computer Application (H)
0.01
QualificationName_Bachelor of physical education
0.00
QualificationName_Bachelors in Accounting & Finance
-0.02
QualificationName_Bachelor’s Degree In Tourism Studies
0.01
QualificationName_DIPLOMA IN ELECTRONICS & COMMUNICATION ENGINEERING
-0.03
QualificationName_Diploma
0.03
QualificationName_Diploma (12+3 yrs)
-0.01
QualificationName_Diploma in Engineering
-0.00
QualificationName_Diploma in General Nursing and Midwifery
0.01
QualificationName_HSC
-0.00
QualificationName_Integrated PG Course
0.01
QualificationName_LLB
0.01
QualificationName_M Tech-M.E.
0.01
QualificationName_MBA/PGDM
0.01
QualificationName_Master Of Business Administration
-0.01
QualificationName_Master in Marketing Management (MMM)
0.01
QualificationName_Master of Arts (M.A.)
-0.05
QualificationName_Master of Commerce (M.Com.)
-0.01
QualificationName_Master of Communication Studies
-0.04
QualificationName_Master of Computer Applications (M.C.A.)
-0.01
QualificationName_Master of Law (L.L.M.)
0.01
QualificationName_Master of Management Studies (M.M.S.)
0.01
QualificationName_Master of Philosophy
-0.02
QualificationName_Master of Science (M.Sc.)
0.01
0.04
0.02
0.00
−0.02
−0.04


Correlation with Attrition - CollegeName Attributes
CollegeName_A. D. institutes of management 0.00
CollegeName_AADHARSH COLLEGE 0.01
CollegeName_ACHARAYA TULASI NATIONAL COLLAGE OF COMMERCE 0.01
CollegeName_ACS College Chandrapur 0.01
CollegeName_All INDIA 0.01
CollegeName_AIISM-DWD 0.01
CollegeName_ALAGAPPA UNIVERSITY 0.01
CollegeName_ANJUMAN ARTS, SCIENCE &COMMERCE COLLEGE BHATKAL 0.00
CollegeName_ANNA UNIVERSITY 0.01
CollegeName_ANNAMALAI UNIVERSITY 0.01
CollegeName_ANNASAHEB MAGAR COLLEGE MANJRI HADAPSAR 0.01
CollegeName_ARG COLLEAGE 0.01
CollegeName_ARG College 0.01
CollegeName_ART'S , SCIENCE AND COMMERCE COLLEGE HADAPSAR 0.01
CollegeName_ARTS AND COMMERCE COLLEGE LANJA 0.01
CollegeName_ARTS& COMMERRCE COLLEGE MENDRDA 0.01
CollegeName_Ahmednagar College Ahmednagar 0.00
CollegeName_Allagadda Institute of management sciences 0.01
CollegeName_Allahabad University 0.01
CollegeName_Alvas college ,moodabidre 0.01
CollegeName_Art,Sci. &Comm. College,Chikaldara 0.01
CollegeName_Asmita foundation commerce college 0.01
CollegeName_B P BRAHAMBHATT ARTS & COMMERS COLLAGE 0.00
CollegeName_B.J.S.RAMPURIYA JAIN COLLEGE,BIKANER 0.01
CollegeName_B.S.PATIL COLLEGE 0.00
CollegeName_B.S.R college Alwar 0.00
CollegeName_B.V.Bhoomreddy College 0.00
CollegeName_BABASAHEB AMBEDKAR COLLEGE 0.00
CollegeName_BAHAUDDIN ARTS COLLAGE 0.00
CollegeName_BALWANT COLLEGE VITA 0.00
CollegeName_BARMER 0.01
CollegeName_BASAMMA COLLEGE 0.01
CollegeName_BHADRA INSTUTE OF COLLAGE 0.01
CollegeName_BHARATH COLLEGE OF SCIENCE AND MANAGEMENT 0.01
CollegeName_BHARATI VIDYAPITH SANGLI 0.00
CollegeName_BHARTHIDASAN UNIVERSITY 0.01
CollegeName_BITS 0.00
CollegeName_BRDBDPG COLLAGE BARAHAZ BAZAR 0.00
CollegeName_Bahavnagar collage 0.01
CollegeName_Balaji Institute of Technology &science 0.01
CollegeName_Basic College, Bikaner 0.01
CollegeName_Bhabuti mahavidyalaya amgaon 0.01
CollegeName_Bharathidasan University 0.01
CollegeName_Bhartiya Mahavidhyala Amravati 0.00
CollegeName_Bhavnagar 0.01
CollegeName_Bishop caldwel collage 0.00


Correlation with Attrition - SpecializationName Attributes
SpecializationName_(FINANCE & MARKETING) -0.04
SpecializationName_Accounting -0.04
SpecializationName_Accounting & Finance -0.00
SpecializationName_Agriculture 0.01
SpecializationName_Architecture -0.05
SpecializationName_Arts 0.02
SpecializationName_Arts&Humanities -0.06
SpecializationName_BACHELOR OF COMMERCE 0.01
SpecializationName_BACHELORE OF COMPUTER APPLICATION -0.04
SpecializationName_BANKING & INSURANCE 0.01
SpecializationName_Bachelor In Business Management 0.02
SpecializationName_Bachelor of Law (LLB) 0.00
SpecializationName_Banking 0.00
SpecializationName_Banking & Insurance 0.01
SpecializationName_Bio-Chemistry -0.03
SpecializationName_Biology -0.01
SpecializationName_Botany 0.01
SpecializationName_Business Administration -0.02
SpecializationName_Business Administration 0.02
SpecializationName_Business Operation Management -0.04
SpecializationName_COMPUTER ENGINEERING 0.01
SpecializationName_Catering Science & Hospitality Management 0.00
SpecializationName_Chemical 0.01
SpecializationName_Chemistry -0.00
SpecializationName_Civil 0.01
SpecializationName_Civil Engineering 0.02
SpecializationName_Commerce 0.02
SpecializationName_Communication -0.06
SpecializationName_Computer Applications -0.00
SpecializationName_Computer science & engineering 0.02
SpecializationName_Computers 0.01
SpecializationName_Computers / IT -0.01
SpecializationName_Computers IT -0.02
SpecializationName_Corporate Secretaryship 0.01
SpecializationName_DIPLOMA IN COMPUTER ENGINEERING -0.02
SpecializationName_DIPLOMA IN COMPUTER SCIENCE -0.03
SpecializationName_DIPLOMA IN ELECTRONICS & COMMUNICATION ENGINEERING -0.00
SpecializationName_DIPLOMA IN INFORMATION TECHNOLOGY 0.01
SpecializationName_Diploma In Computer Application 0.01
SpecializationName_Economics -0.00
SpecializationName_Economics. 0.01
SpecializationName_Electrical -0.02
SpecializationName_Electrical & Electronics Engineering 0.00
SpecializationName_Electronics 0.01
SpecializationName_Electronics & Tele-communication Engineering -0.01
0.02
0.00








```
# Print top correlations with Attrition
print("\nTop Correlations with Attrition:")
print(correlation_matrix['Attrition'].sort_values(ascending=False))
```

```
Top Correlations with Attrition:
Attrition                      1.000000
ProfileName_CSO                0.217107
Age                            0.189261
```

```
Gender_Male                     0.056686
Region_West-1                   0.055392
                                  ...
SpecializationName_Others      -0.102512
PassingYear                    -0.140054
GradeCode_G3                   -0.193767
DepartmentName_Transaction     -0.230091
Overall LQ Score               -0.404390
Name: Attrition, Length: 642, dtype: float64
```

Data Preprocessing

Handling Dates

Converted date columns to datetime format to facilitate analysis.

Calculating Tenure and Age

Calculated tenure (years employed) and age (years old) of employees.

Handling Missing Values

Filled missing values in 'CollegeName' with 'Unknown' and 'Overall LQ Score' with median.

Creating Target Variable

Created 'Attrition' variable based on 'Status', where 1 indicates leaving and 0 indicates not.

Handling Missing Values in Categorical Columns

Filled missing values in categorical columns with 'Unknown'.

By completing these steps, we ensured the dataset was clean and ready for analysis.

```
import pandas as pd
import numpy as np
from datetime import datetime

# Load the data from CSV file
data_path = r"C:\Users\ASUS\Downloads\Motilal Predictive Analytics\
EmpData_MotilalOswal_HRAnalytics.csv"
data = pd.read_csv(data_path)

# Handle dates first
date_columns = ['JoiningDate', 'BirthDate', 'LastWorkingDate']
for col in date_columns:
```

```
    data[col] = pd.to_datetime(data[col], format='%d-%m-%Y',
errors='coerce')

# Drop rows with missing critical dates
data.dropna(subset=['JoiningDate', 'BirthDate'], inplace=True)

# Calculate tenure and age
data['Tenure'] = (pd.Timestamp.now() - data['JoiningDate']).dt.days /
365.25
data['Age'] = (pd.Timestamp.now() - data['BirthDate']).dt.days /
365.25

# Handle missing values
data['CollegeName'].fillna('Unknown', inplace=True)
data['Overall LQ Score'].fillna(data['Overall LQ Score'].median(),
inplace=True)

# Create the target variable 'Attrition' based on 'Status' values
data['Attrition'] = data['Status'].apply(lambda x: 1 if x in
['Inactive', 'Resigned'] else 0)

# Handle missing values in categorical columns
categorical_columns = ['ProfileName', 'DepartmentName', 'Gender',
'QualificationName', 'SpecializationName', 'Region']
for col in categorical_columns:
    data[col].fillna('Unknown', inplace=True)
```

```
C:\Users\ASUS\AppData\Local\Temp\ipykernel_17232\1825930221.py:22:
FutureWarning: A value is trying to be set on a copy of a DataFrame or
Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never
work because the intermediate object on which we are setting values
always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try
using 'df.method({col: value}, inplace=True)' or df[col] =
df[col].method(value) instead, to perform the operation inplace on the
original object.


  data['CollegeName'].fillna('Unknown', inplace=True)
C:\Users\ASUS\AppData\Local\Temp\ipykernel_17232\1825930221.py:23:
FutureWarning: A value is trying to be set on a copy of a DataFrame or
Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never
work because the intermediate object on which we are setting values
always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try
using 'df.method({col: value}, inplace=True)' or df[col] =
```

```
df[col].method(value) instead, to perform the operation inplace on the
original object.


  data['Overall LQ Score'].fillna(data['Overall LQ Score'].median(),
inplace=True)
C:\Users\ASUS\AppData\Local\Temp\ipykernel_17232\1825930221.py:31:
FutureWarning: A value is trying to be set on a copy of a DataFrame or
Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never
work because the intermediate object on which we are setting values
always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try
using 'df.method({col: value}, inplace=True)' or df[col] =
df[col].method(value) instead, to perform the operation inplace on the
original object.


  data[col].fillna('Unknown', inplace=True)

# Create pivot tables for each categorical attribute to summarize
attrition
pivot_tables = {}
for col in categorical_columns:
    pivot = data.pivot_table(index=col, values='Attrition',
aggfunc='sum').reset_index()
    pivot.columns = [col, 'Attrition Count']

    # Calculate percentage of grand total
    pivot['Percentage of Grand Total'] = pivot['Attrition Count'] /
pivot['Attrition Count'].sum() * 100

    # Sort pivot table by attrition count in descending order
    pivot = pivot.sort_values(by='Attrition Count', ascending=False)

    pivot_tables[col] = pivot

# Display the pivot tables in a readable format
for col, pivot in pivot_tables.items():
    print(f"\nAttrition Counts by {col}:")
    print(pivot.to_string(index=False))

# Additionally, you can save these pivot tables to CSV files if needed
for col, pivot in pivot_tables.items():
    pivot.to_csv(f"{col}_attrition_counts.csv", index=False)


Attrition Counts by ProfileName:
ProfileName  Attrition Count  Percentage of Grand Total
```

```
       CSM            4954                       59.046484
       CSO            2714                       32.348033
     S-CSM             722                        8.605483
    CSM-CF               0                        0.000000

Attrition Counts by DepartmentName:
DepartmentName  Attrition Count  Percentage of Grand Total
   Transaction             6246                  74.445769
   CSM Affairs             2144                  25.554231

Attrition Counts by Gender:
Gender  Attrition Count  Percentage of Grand Total
  Male             8227                  98.057211
Female              163                   1.942789

Attrition Counts by QualificationName:
                                      QualificationName  Attrition Count
Percentage of Grand Total
                         Bachelor of Commerce (B.Com.)             1715
20.441001
                                Bachelor of Arts (B.A)             1682
20.047676
                                                   HSC             1599
19.058403
                                              MBA/PGDM              658
7.842670
                           Bachelor of Science (B.Sc.)              555
6.615018
      Bachelor of Business Administration (B.B.A.)                 271
3.230036
                                                Others              250
2.979738
                                           B.Tech/B.E.              240
2.860548
                           Master of Commerce (M.Com.)              236
2.812872
        Bachelor of Computer Applications (B.C.A.)                  196
2.336114
                                 Master of Arts (M.A.)              176
2.097735
                                                   SSC              170
2.026222
                                               Diploma              152
1.811681
                                               Unknown              125
1.489869
                            Master of Science (M.Sc.)                58
0.691299
                                   Diploma (12+3 yrs)                45
0.536353
```

Bachelor of Education (B.Ed.)	35	0.417163
Master of Computer Applications (M.C.A.)	32	0.381406
Master Of Business Administration	32	0.381406
Master of Social Work (MSW)	28	0.333731
Bachelor of Management Studies (B.M.S.)	19	0.226460
Bachelor In Business Management (BBM)	19	0.226460
Bachelor of Engineering	18	0.214541
PG Diploma	10	0.119190
LLB	10	0.119190
Diploma in Engineering	5	0.059595
Bachelor of Hotel Management (B.H.M.)	5	0.059595
Bachelor of Science Computer Application (H)	5	0.059595
Master of Law (L.L.M.)	4	0.047676
Master of Management Studies (M.M.S.)	4	0.047676
Bachelor’s Degree In Tourism Studies	4	0.047676
Bachelors in Accounting & Finance	4	0.047676
Integrated PG Course	3	0.035757
Ministry of Human Resource Development (MHRD)	3	0.035757
Bachelor Of Corp Secretaryship	3	0.035757
Bachelor of Pharmacy (B.Pharma.)	3	0.035757
Bachelor of Literature (B.Lit)	3	0.035757
M Tech-M.E.	3	0.035757
Diploma in General Nursing and Midwifery	2	0.023838
Master in Marketing Management (MMM)	2	0.023838
PG diploma in Banking	1	

```
0.011919
              Post Graduate Diploma in Management                1
0.011919
                                              BAF                1
0.011919
                                           6Sigma                1
0.011919
                           Bachelor of Law (LLB)                 1
0.011919
                     Bachelor of physical education              1
0.011919
                            Master of Philosophy                 0
0.000000
                    Master of Communication Studies              0
0.000000
                 Bachelor of Architecture (B.Arch.)              0
0.000000
                  Bachelor of Business Studies(BBS)              0
0.000000
DIPLOMA IN ELECTRONICS & COMMUNICATION ENGINEERING               0
0.000000
                    Bachelor of Mass Media (B.M.M.)              0
0.000000
          Bachelor of Information Technology (BIT)               0
0.000000

Attrition Counts by SpecializationName:
                               SpecializationName  Attrition Count
Percentage of Grand Total
                                         Commerce              945
11.263409
                             Accounting & Finance              880
10.488677
                                             Arts              847
10.095352
                                          Unknown              818
9.749702
                                  Arts&Humanities              550
6.555423
                                          Science              452
5.387366
                                           Others              398
4.743743
                                        Marketing              293
3.492253
                                          Finance              275
3.277712
                                       Accounting              257
3.063170
```

Business Administration	207	2.467223
Computer Applications	199	2.371871
Vocational	168	2.002384
SSC	158	1.883194
History	158	1.883194
Chemistry	121	1.442193
Economics	119	1.418355
Sociology	108	1.287247
Political Science	108	1.287247
Computers / IT	103	1.227652
Hindi	75	0.893921
Mechanical	69	0.822408
Restructured	69	0.822408
Computers	57	0.679380
Financial Markets	47	0.560191
Business Administration	45	0.536353
Maths	44	0.524434
Business Operation Management	39	0.464839
(FINANCE & MARKETING)	39	0.464839
Electrical	36	0.429082
Physics	33	0.393325
HR / Industrial Relations	32	0.381406
English	31	0.369487
Electronics	28	0.333731
Mechanical Engineering	27	

```
0.321812
                          Management Studies            23
0.274136
                                   Fine Arts            23
0.274136
              Computer science & engineering            23
0.274136
                      Information Technology            22
0.262217
                         Banking & Insurance            21
0.250298
                                 Agriculture            21
0.250298
             Bachelor In Business Management            18
0.214541
                                 Social work            17
0.202622
                                     Zoology            17
0.202622
                                   Teaching.            16
0.190703
                          Civil  Engineering            16
0.190703
             Electronics / Telecommunication            16
0.190703
                                 [Sociology]            15
0.178784
                                 Mathematics            13
0.154946
                                    Sanskrit            12
0.143027
        Electrical & Electronics Engineering            12
0.143027
                                     Geology            12
0.143027
        Hospitality and Hotel Administration            11
0.131108
                 PGDI (FINANCIAL MANAGEMENT)            11
0.131108
                                H.S.C.PASSED            11
0.131108
                                Computers IT             9
0.107271
    Electronics and Telecommunication Enginerring        8
0.095352
                        BANKING &   INSURANCE            8
0.095352
                         Finance  Management             8
0.095352
```

H.S.C	8	0.095352
Electronics & Tele-communication Engineering	8	0.095352
DIPLOMA IN ELECTRONICS & COMMUNICATION ENGINEERING	8	0.095352
Civil	7	0.083433
Environmental science	7	0.083433
COMPUTER ENGINEERING	7	0.083433
Communication	6	0.071514
HR & Marketing	6	0.071514
Operations	6	0.071514
Corporate Secretaryship	6	0.071514
Manufacturing Science & Engineering	6	0.071514
M.A	6	0.071514
Diploma In Computer Application	6	0.071514
Masters in Social Work	5	0.059595
Bio-Chemistry	5	0.059595
Other Management	5	0.059595
Pass Course	4	0.047676
Marketing & Human Resources	4	0.047676
PR / Advertising	4	0.047676
M.SC COMPUTER SCIENCE	4	0.047676
Law	4	0.047676
Tourism studies	4	0.047676
Botany	4	0.047676
Biology	4	0.047676
Laser and Electro Optical Engineering	3	

		0.035757
Pharmacy	3	0.035757
Economics.	3	0.035757
Statistics	3	0.035757
DIPLOMA IN COMPUTER ENGINEERING	3	0.035757
Chemical	3	0.035757
BACHELORE OF COMPUTER APPLICATION	3	0.035757
Electronics Engineering	3	0.035757
Production / Industrial	2	0.023838
Psychology	2	0.023838
Vocational Course	2	0.023838
General Nursing and Midwifery	2	0.023838
DIPLOMA IN INFORMATION TECHNOLOGY	2	0.023838
BACHELOR OF COMMERCE	2	0.023838
Marketing Management	2	0.023838
LLB	2	0.023838
Journalism	2	0.023838
Information Technology	2	0.023838
Hotel Management	2	0.023838
Other Engineering	2	0.023838
Finance and Management	2	0.023838
P.G.D. (SALES & MARKETING)	1	0.011919
Bachelor of Law (LLB)	1	0.011919
Banking	1	0.011919
Catering Science & Hospitality Management	1	0.011919

```
                                          Finance-PGDM                1
0.011919
                                                 FINAL                1
0.011919
                                             Geography                1
0.011919
                                           (ECONOMICS)                1
0.011919
                         DIPLOMA IN COMPUTER SCIENCE                  0
0.000000
                                 MASTER OF PHILOSOPHY                 0
0.000000
                                            Mass Media                0
0.000000
                                          Architecture                0
0.000000

Attrition Counts by Region:
 Region  Attrition Count  Percentage of Grand Total
 West-1             2666                  31.775924
  South             2307                  27.497020
 West-2             2006                  23.909416
  North              497                   5.923719
 Mumbai              412                   4.910608
Unknown              366                   4.362336
   East              136                   1.620977
     HO                0                   0.000000
```

Predictive Analytics

Understanding Machine Learning Modules and Functions

In machine learning, various modules and functions are used to build, evaluate, and fine-tune models. Let's delve into the purpose and significance of each:

1. **train_test_split**: Splits a dataset into training and testing sets, crucial for assessing model performance.

2. **StratifiedKFold**: Ensures class distribution balance across folds in cross-validation, beneficial for imbalanced datasets.

3. **cross_val_score**: Performs cross-validation to evaluate model performance robustly.

4. **StandardScaler**: Standardizes features to ensure uniformity, essential for models sensitive to feature scale.

5. **LogisticRegression**: Implements logistic regression for binary classification tasks.

6. **SimpleImputer**: Imputes missing values in datasets, ensuring completeness before training.

7. **accuracy_score**: Calculates the proportion of correctly classified instances, a fundamental metric for classification.

8. **precision_recall_fscore_support**: Computes precision, recall, F1-score, and support for each class, offering insights into model performance.

9. **roc_auc_score**: Measures the area under the ROC curve, indicating model discriminative ability.

10. **classification_report**: Generates a comprehensive report with performance metrics for each class, aiding in model evaluation.

These modules and functions are indispensable for developing and evaluating machine learning models effectively.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime
from sklearn.model_selection import train_test_split, StratifiedKFold,
cross_val_score
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.impute import SimpleImputer
from sklearn.metrics import accuracy_score,
precision_recall_fscore_support, roc_auc_score, classification_report

# Load the data from CSV file
data_path = r"C:\Users\ASUS\Downloads\Motilal Predictive Analytics\
EmpData_MotilalOswal_HRAnalytics.csv"
data = pd.read_csv(data_path)

# Handle dates
date_columns = ['JoiningDate', 'BirthDate', 'LastWorkingDate']
for col in date_columns:
    data[col] = pd.to_datetime(data[col], format='%d-%m-%Y',
errors='coerce')

# Drop rows with missing critical dates
data.dropna(subset=['JoiningDate', 'BirthDate'], inplace=True)

# Calculate tenure and age
data['Tenure'] = (pd.Timestamp.now() - data['JoiningDate']).dt.days /
365.25
data['Age'] = (pd.Timestamp.now() - data['BirthDate']).dt.days /
365.25
```

```
# Handle missing values
data['CollegeName'].fillna('Unknown', inplace=True)
data['Overall LQ Score'].fillna(data['Overall LQ Score'].median(),
inplace=True)
```

```
C:\Users\ASUS\AppData\Local\Temp\ipykernel_17232\420196649.py:2:
FutureWarning: A value is trying to be set on a copy of a DataFrame or
Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never
work because the intermediate object on which we are setting values
always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try
using 'df.method({col: value}, inplace=True)' or df[col] =
df[col].method(value) instead, to perform the operation inplace on the
original object.


  data['CollegeName'].fillna('Unknown', inplace=True)
C:\Users\ASUS\AppData\Local\Temp\ipykernel_17232\420196649.py:3:
FutureWarning: A value is trying to be set on a copy of a DataFrame or
Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never
work because the intermediate object on which we are setting values
always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try
using 'df.method({col: value}, inplace=True)' or df[col] =
df[col].method(value) instead, to perform the operation inplace on the
original object.


  data['Overall LQ Score'].fillna(data['Overall LQ Score'].median(),
inplace=True)
```

```
# Create the target variable 'Attrition' based on 'Status' values
data['Attrition'] = data['Status'].apply(lambda x: 1 if x in
['Inactive', 'Resigned'] else 0)

# Handle missing values in categorical columns
categorical_columns = ['ProfileName', 'DepartmentName', 'Gender',
'QualificationName', 'SpecializationName', 'Region']
for col in categorical_columns:
    data[col].fillna('Unknown', inplace=True)
```

```
C:\Users\ASUS\AppData\Local\Temp\ipykernel_17232\1345938875.py:4:
FutureWarning: A value is trying to be set on a copy of a DataFrame or
Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never
work because the intermediate object on which we are setting values
always behaves as a copy.
```

```
For example, when doing 'df[col].method(value, inplace=True)', try
using 'df.method({col: value}, inplace=True)' or df[col] =
df[col].method(value) instead, to perform the operation inplace on the
original object.


  data[col].fillna('Unknown', inplace=True)

# One-hot encoding
data_encoded = pd.get_dummies(data, columns=categorical_columns,
drop_first=True)
```

Feature Selection: The features (independent variables) and target variable (dependent variable) are selected. Numerical columns such as 'Tenure', 'Age', and 'Overall LQ Score' are chosen along with the one-hot encoded categorical columns as features. The target variable is 'Attrition', indicating whether an employee has left the company.

```
# Select features and target
numerical_columns = ['Tenure', 'Age', 'Overall LQ Score']
X = data_encoded[numerical_columns + [col for col in
data_encoded.columns if col.startswith(tuple(categorical_columns))]]
y = data_encoded['Attrition']
```

Handling Missing Values: Any remaining missing values in the features are handled by replacing them with the median value of the respective column using SimpleImputer.

```
# Handle any remaining NaNs
if X.isna().sum().sum() > 0:
    imputer = SimpleImputer(strategy='median')
    X_imputed = pd.DataFrame(imputer.fit_transform(X),
columns=X.columns, index=X.index)
else:
    X_imputed = X
```

Data Splitting: The dataset is split into training and testing sets using the train_test_split function. The split is stratified based on the target variable 'Attrition' to maintain class balance in both sets.

```
# Split data
X_train, X_test, y_train, y_test = train_test_split(X_imputed, y,
test_size=0.2, random_state=42, stratify=y)
```

Feature Scaling: The features are scaled using StandardScaler to standardize their distribution, which can improve the performance of the logistic regression model.

```
# Scale features
scaler = StandardScaler()
```

```
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Model Training: A logistic regression model is trained on the training data using LogisticRegression with a maximum of 1000 iterations.

```
# Train logistic regression model
lr_model = LogisticRegression(max_iter=1000)
lr_model.fit(X_train_scaled, y_train)
```

```
LogisticRegression(max_iter=1000)
```

```
# Predictions
y_pred = lr_model.predict(X_test_scaled)
y_pred_proba = lr_model.predict_proba(X_test_scaled)[:, 1]
```

Model Evaluation: The trained model is used to make predictions on the test data. Performance metrics such as accuracy, precision, recall, F1-score, and AUC-ROC score are calculated to evaluate the model's performance. The classification report provides a detailed summary of precision, recall, F1-score, and support for each class.

```
# Evaluate the model
print("Logistic Regression Results:")
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Precision, Recall, F1-Score:",
precision_recall_fscore_support(y_test, y_pred, average='binary'))
print("AUC-ROC Score:", roc_auc_score(y_test, y_pred_proba))
print("Classification Report:")
print(classification_report(y_test, y_pred))
```

```
Logistic Regression Results:
Accuracy: 0.9129770992366413
Precision, Recall, F1-Score: (0.9498507462686567, 0.9481525625744934,
0.9490008947211452, None)
AUC-ROC Score: 0.9589107656784043
Classification Report:
              precision    recall  f1-score   support

           0       0.70      0.71      0.70       287
           1       0.95      0.95      0.95      1678

    accuracy                           0.91      1965
   macro avg       0.82      0.83      0.83      1965
weighted avg       0.91      0.91      0.91      1965
```

Here's how to interpret the logistic regression results:

1. **Accuracy**: The accuracy of the model is 91.30%, indicating that it correctly predicts whether an employee will leave or not approximately 91.30% of the time. It's a measure of overall correctness in classification.

2. **Precision, Recall, F1-Score**:

    - **Precision**: Precision measures the proportion of true positive predictions out of all positive predictions made by the model. In this case, the precision for class 0 (employees who stayed) is 70% and for class 1 (employees who left) is 95%. It means that out of all predicted instances of each class, 70% of those predicted to stay actually stayed, and 95% of those predicted to leave actually left.
    - **Recall (Sensitivity)**: Recall measures the proportion of true positive predictions out of all actual positives in the data. In this case, the recall for class 0 is 71% and for class 1 is 95%. It means that the model correctly identifies 71% of employees who actually stayed and 95% of employees who actually left.
    - **F1-Score**: F1-score is the harmonic mean of precision and recall, providing a balance between the two metrics. It is a measure of a test's accuracy that considers both the precision and the recall. In this case, the F1-score for class 0 is 0.70 and for class 1 is 0.95.
3. **AUC-ROC Score**: The Area Under the Receiver Operating Characteristic Curve (AUC-ROC) measures the model's ability to distinguish between the positive and negative classes. It ranges from 0 to 1, where a score closer to 1 indicates better performance. In this case, the AUC-ROC score is 0.959, indicating high discriminative ability of the model.

4. **Classification Report**: The classification report provides a summary of precision, recall, F1-score, and support (the number of actual occurrences of the class in the specified dataset) for each class (0 and 1) in the dataset. It provides a more detailed evaluation of the model's performance for each class.

In summary, these results suggest that the logistic regression model performs well in predicting employee attrition, with high accuracy, precision, recall, F1-score, and AUC-ROC score. However, it's essential to consider the specific context of the problem and the business implications of false positives and false negatives when interpreting these results.

Cross-Validation: Cross-validation is performed using StratifiedKFold with 5 folds to assess the model's robustness and generalization performance. F1-scores are calculated for each fold, and the mean F1-score is reported as an overall measure of the model's performance.

```
# Cross-validation to evaluate model performance
sss = StratifiedKFold(n_splits=5, random_state=42, shuffle=True)
cross_val_scores = cross_val_score(lr_model, X_imputed, y, cv=sss,
scoring='f1')

print("Cross-validated F1-Scores:", cross_val_scores)
print("Mean F1-Score:", cross_val_scores.mean())
```

```
Cross-validated F1-Scores: [0.94777448 0.94777448 0.94447761
0.94764862 0.94874074]
Mean F1-Score: 0.9472831877545259
```

Predicting the actual Attrition based on the past data

```python
# Predict attrition for the entire dataset
y_pred_all = lr_model.predict(X_imputed)
data_encoded['Attrition_Predicted'] = y_pred_all

# Display the updated dataframe with predicted attrition
print(data_encoded[['EmployeeId', 'Status', 'Attrition_Predicted']])
```

```
      EmployeeId    Status  Attrition_Predicted
0              0  Inactive                    1
1              1  Inactive                    1
2              2  Inactive                    1
3              3  Inactive                    1
4              4  Inactive                    1
...          ...       ...                  ...
9846        9846  Inactive                    1
9847        9847  Inactive                    1
9848        9848  Inactive                    1
9849        9849  Inactive                    1
9850        9850  Inactive                    1

[9825 rows x 3 columns]

c:\Users\ASUS\AppData\Local\Programs\Python\Python312\Lib\site-
packages\sklearn\base.py:486: UserWarning: X has feature names, but
LogisticRegression was fitted without feature names
  warnings.warn(
```

```python
# Save the dataframe to a CSV file
data_encoded[['EmployeeId', 'Status',
'Attrition_Predicted']].to_csv('predicted_attrition1.csv',
index=False)
```

Imabalance Checking

```python
from imblearn.over_sampling import SMOTE

# Instantiate SMOTE
smote = SMOTE(random_state=42)

# Apply SMOTE to the data
```

```
X_resampled, y_resampled = smote.fit_resample(X_imputed, y)

# Split the resampled data into training and testing sets
X_train_resampled, X_test_resampled, y_train_resampled,
y_test_resampled = train_test_split(X_resampled, y_resampled,
test_size=0.2, random_state=42)

# Train logistic regression model on resampled data
lr_model_resampled = LogisticRegression(max_iter=1000)
lr_model_resampled.fit(X_train_resampled, y_train_resampled)

# Predictions on resampled data
y_pred_resampled = lr_model_resampled.predict(X_test_resampled)

# Evaluate the model on resampled data
print("Logistic Regression Results on Resampled Data:")
print("Accuracy:", accuracy_score(y_test_resampled, y_pred_resampled))
print("Precision, Recall, F1-Score:",
precision_recall_fscore_support(y_test_resampled, y_pred_resampled,
average='binary'))
print("AUC-ROC Score:", roc_auc_score(y_test_resampled,
lr_model_resampled.predict_proba(X_test_resampled)[:, 1]))
print("Classification Report:")
print(classification_report(y_test_resampled, y_pred_resampled))
```

```
Logistic Regression Results on Resampled Data:
Accuracy: 0.932657926102503
Precision, Recall, F1-Score: (0.9585427135678392, 0.9051008303677343,
0.9310555216595485, None)
AUC-ROC Score: 0.978657276194941
Classification Report:
              precision    recall  f1-score   support

           0       0.91      0.96      0.93      1670
           1       0.96      0.91      0.93      1686

    accuracy                           0.93      3356
   macro avg       0.93      0.93      0.93      3356
weighted avg       0.93      0.93      0.93      3356
```

```
# Train logistic regression model
lr_model = LogisticRegression(max_iter=1000)
lr_model.fit(X_train_scaled, y_train)

# Assign feature names to coefficients
feature_names = X_train.columns.tolist()
lr_model.coef_ = np.zeros((1, len(feature_names)))
lr_model.coef_[0] = lr_model.coef_
lr_model.intercept_ = np.array([lr_model.intercept_])
```

```python
# Predictions
y_pred = lr_model.predict(X_test_scaled)
y_pred_proba = lr_model.predict_proba(X_test_scaled)[:, 1]

# Predict attrition for the entire dataset
y_pred_all = lr_model.predict(X_imputed)
data_encoded['Attrition_Predicted'] = y_pred_all

# Display the updated dataframe with predicted attrition
print(data_encoded[['EmployeeId', 'Status', 'Attrition_Predicted']])
```

```
      EmployeeId    Status  Attrition_Predicted
0              0  Inactive                    1
1              1  Inactive                    1
2              2  Inactive                    1
3              3  Inactive                    1
4              4  Inactive                    1
...          ...       ...                  ...
9846        9846  Inactive                    1
9847        9847  Inactive                    1
9848        9848  Inactive                    1
9849        9849  Inactive                    1
9850        9850  Inactive                    1

[9825 rows x 3 columns]

c:\Users\ASUS\AppData\Local\Programs\Python\Python312\Lib\site-
packages\sklearn\base.py:486: UserWarning: X has feature names, but
LogisticRegression was fitted without feature names
  warnings.warn(
```

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime
from sklearn.model_selection import train_test_split, StratifiedKFold,
cross_val_score
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.impute import SimpleImputer
from sklearn.metrics import accuracy_score,
precision_recall_fscore_support, roc_auc_score, classification_report
from imblearn.over_sampling import SMOTE
from imblearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestClassifier

# Load the data from CSV file
data_path = r"C:\Users\ASUS\Downloads\Motilal Predictive Analytics\
EmpData_MotilalOswal_HRAnalytics.csv"
data = pd.read_csv(data_path)
```

```
# Handle dates
date_columns = ['JoiningDate', 'BirthDate', 'LastWorkingDate']
for col in date_columns:
    data[col] = pd.to_datetime(data[col], format='%d-%m-%Y',
errors='coerce')

# Drop rows with missing critical dates
data.dropna(subset=['JoiningDate', 'BirthDate'], inplace=True)

# Calculate tenure and age
data['Tenure'] = (pd.Timestamp.now() - data['JoiningDate']).dt.days /
365.25
data['Age'] = (pd.Timestamp.now() - data['BirthDate']).dt.days /
365.25

# Create the target variable 'Attrition' based on 'Status' values
data['Attrition'] = data['Status'].apply(lambda x: 1 if x in
['Inactive', 'Resigned'] else 0)

# Check class distribution
print("Class distribution:")
print(data['Attrition'].value_counts(normalize=True))
```

```
Class distribution:
Attrition
1    0.853944
0    0.146056
Name: proportion, dtype: float64
```

```
# One-hot encoding
categorical_columns = ['ProfileName', 'DepartmentName', 'Gender',
'QualificationName', 'SpecializationName', 'Region', 'CollegeName']
data_encoded = pd.get_dummies(data, columns=categorical_columns,
drop_first=True)

# Select features and target
numerical_columns = ['Tenure', 'Age', 'Overall LQ Score']
X = data_encoded[numerical_columns + [col for col in
data_encoded.columns if col.startswith(tuple(categorical_columns))]]
y = data_encoded['Attrition']

# Impute missing values in numerical columns
imputer = SimpleImputer(strategy='median')
X_imputed = pd.DataFrame(imputer.fit_transform(X), columns=X.columns,
index=X.index)

# Split data
X_train, X_test, y_train, y_test = train_test_split(X_imputed, y,
test_size=0.2, random_state=42, stratify=y)
```

```
# Scale features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Oversample the minority class
smote = SMOTE(random_state=42)
X_train_resampled, y_train_resampled =
smote.fit_resample(X_train_scaled, y_train)

# Train Random Forest model
rf_model = RandomForestClassifier(n_estimators=100,
class_weight='balanced', random_state=42)
rf_model.fit(X_train_resampled, y_train_resampled)
```

```
RandomForestClassifier(class_weight='balanced', random_state=42)
```

```
# Predict on test set
y_pred_rf = rf_model.predict(X_test_scaled)
y_pred_proba_rf = rf_model.predict_proba(X_test_scaled)[:, 1]

# Evaluate the model
print("\nRandom Forest Results:")
print("Accuracy:", accuracy_score(y_test, y_pred_rf))
print("Precision, Recall, F1-Score:",
precision_recall_fscore_support(y_test, y_pred_rf, average='binary'))
print("AUC-ROC Score:", roc_auc_score(y_test, y_pred_proba_rf))
print("\nClassification Report:")
print(classification_report(y_test, y_pred_rf))
```

```
Random Forest Results:
Accuracy: 0.9796437659033079
Precision, Recall, F1-Score: (0.9939686369119421, 0.9821215733015495,
0.988009592326139, None)
AUC-ROC Score: 0.9970140327999568

Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.97      0.93       287
           1       0.99      0.98      0.99      1678

    accuracy                           0.98      1965
   macro avg       0.95      0.97      0.96      1965
weighted avg       0.98      0.98      0.98      1965
```

```
# Feature importances
feature_importances = pd.Series(rf_model.feature_importances_,
index=X.columns).sort_values(ascending=False)
```

```python
print("\nTop 10 features by importance:")
print(feature_importances.head(10))
```

```
Top 10 features by importance:
Tenure                           0.356951
Overall LQ Score                 0.239333
Age                              0.081275
ProfileName_CSO                  0.041817
DepartmentName_Transaction       0.040154
Region_West-1                    0.016387
ProfileName_S-CSM                0.014764
Region_West-2                    0.013243
Region_South                     0.013091
SpecializationName_Others        0.010193
dtype: float64
```

```python
# Predict attrition for the entire dataset
X_all_scaled = scaler.transform(X_imputed)
y_pred_all_rf = rf_model.predict(X_all_scaled)
data_encoded['Attrition_Predicted'] = y_pred_all_rf

# Map predictions back to original status
status_map = {0: 'Active', 1: 'Inactive'}
data_encoded['Status_Predicted'] =
data_encoded['Attrition_Predicted'].map(status_map)

# Display the updated dataframe with predicted status
print("\nSample of predictions:")
print(data_encoded[['EmployeeId', 'Status',
'Status_Predicted']].sample(20))
```

```
Sample of predictions:
      EmployeeId    Status Status_Predicted
2054        2054  Inactive         Inactive
2514        2514  Inactive         Inactive
5138        5138  Inactive         Inactive
9702        9702  Inactive         Inactive
9683        9683  Inactive         Inactive
3593        3593  Inactive         Inactive
7942        7942  Inactive         Inactive
2972        2972  Inactive         Inactive
2340        2340  Inactive         Inactive
5447        5447  Inactive         Inactive
8528        8528  Inactive         Inactive
539          539  Inactive         Inactive
8036        8036    Active           Active
2221        2221  Inactive         Inactive
9350        9350    Active           Active
2130        2130  Inactive         Inactive
```

```
7932        7932  Inactive         Inactive
4499        4499  Inactive         Inactive
6321        6321  Inactive         Inactive
7531        7531    Active           Active

# Confusion matrix for actual vs predicted status
print("\nConfusion Matrix (Actual vs Predicted):")
print(pd.crosstab(data_encoded['Status'],
data_encoded['Status_Predicted']))
```

```
Confusion Matrix (Actual vs Predicted):
Status_Predicted  Active  Inactive
Status
Active              1425        10
Inactive              23      8227
Resigned               7       133
```