**Step 1: Data Understanding**

* **Packages Installation**: You’ve installed all the necessary libraries to work with data, perform analysis, and visualize results.
* **Dataset Loading**: The read.csv function loads the dataset correctly.
* **Exploration**: You’re using head(data) and str(data) which are great for inspecting the first few rows and the structure of the data. Also, summary(data) gives you an overview of the statistical summaries of the numeric variables, helping in detecting any outliers or strange values.

**Step 2: Exploratory Data Analysis (EDA)**

* You’ve done a great job in exploring basic statistics and structure. The colSums(is.na(data)) ensures that you’re aware of any missing values in the dataset.

**Step 3: Data Cleaning**

* **Handling Missing Data**: Using na.omit to remove rows with missing values is straightforward, but you might also consider imputing missing data using methods like mean/median imputation if you expect to lose a lot of data with this approach.
* **Factor Conversion**: You’ve correctly converted categorical variables into factors. This is crucial for models like logistic regression that require factors to be handled properly.

**Step 4: Exploratory Data Visualization**

* You are using ggplot2 effectively to visualize the distributions of important variables like age, income, job roles, etc. The histograms, boxplots, and bar charts will give you valuable insights into how attrition might be linked to factors like job role or income.
* **Enhancement**: For some plots (e.g., "Distribution of Job Roles by Gender"), it might be beneficial to adjust the axis labels or rotate them for readability. Consider using theme(axis.text.x = element\_text(angle = 45, hjust = 1)) to make long text labels more readable.

**Step 5: Correlation Analysis**

* Using cor() and corrplot() is a good choice to explore relationships between numerical variables. It helps in identifying which features have high correlations, allowing you to decide if certain features can be removed or combined.

**Step 6: Hypothesis Testing**

* **Chi-Square Test**: You’ve tested for independence between Job Role and Attrition, which is appropriate for categorical variables.
* **T-Test**: Using the T-test between Monthly Income and Attrition is valid, as you are comparing the means of two groups (attrition vs. no attrition).
* **Recommendation**: You could include more hypothesis tests, like testing whether there’s a significant difference in other categorical variables (e.g., Department, Gender, etc.).

**Step 7: Building Logistic Regression Model**

* **Model Building**: Your logistic regression model uses Age, MonthlyIncome, JobRole, Department, and MaritalStatus as predictors, which seems reasonable.
* **Model Summary**: It’s a good practice to evaluate the p-values and coefficients of the model summary to understand the relationship between each variable and attrition.

**Step 8: Model Evaluation**

* You are splitting the data into a training set and test set, which is good practice to prevent overfitting.
* The confusion matrix is essential for evaluating the model's classification performance (true positives, false positives, true negatives, and false negatives).
* **Improvement**: You might want to consider adding metrics like accuracy, precision, recall, or F1-score for a more comprehensive evaluation.

**Step 9: Model Performance**

* **ROC Curve**: Plotting the ROC curve is an excellent way to evaluate the trade-off between true positive rate and false positive rate. It’s good that you are using roc() to calculate and plot the curve.
* **Precision-Recall Curve**: This is especially useful when dealing with imbalanced datasets. Consider printing the AUC (Area Under the Curve) for both the ROC and Precision-Recall curves to quantify the model’s performance.

**Output: R Markdown Report**

* Once your analysis is complete, you can use R Markdown to create an automated report with clear explanations, visualizations, and insights. This will provide a clean and professional presentation of your findings.

**Overall Project Flow**

* **Data Understanding and Cleaning**: You start by understanding the dataset’s structure and cleaning it for any missing values or type issues.
* **Exploratory Data Analysis**: You visualize the data to discover patterns and potential relationships between features and attrition.
* **Hypothesis Testing**: You conduct statistical tests to explore the significance of certain features on attrition.
* **Logistic Regression Model**: You build a predictive model to estimate the likelihood of attrition based on key features.
* **Model Evaluation**: You evaluate the performance of your model using confusion matrix, ROC, and precision-recall curves.

**Step 1: Data Understanding**

**Install and Load Libraries**

# Install necessary packages

install.packages("corrplot")

install.packages("caret")

install.packages("ROCR")

install.packages("ggplot2")

# Load necessary libraries

library(tidyverse)

library(corrplot)

library(caret)

library(ROCR)

* **Installing Packages**: The install.packages() function is used to download libraries from CRAN (R’s package repository). These libraries contain pre-built functions that make tasks easier.
  + corrplot: Used for visualizing correlation matrices.
  + caret: Provides a comprehensive set of tools for data pre-processing, model training, and evaluation.
  + ROCR: Helps to evaluate the performance of classification models (like logistic regression) using ROC curves and other metrics.
  + ggplot2: A powerful visualization package for creating plots.
* **Loading Libraries**: The library() function loads the installed packages so you can use their functions in the rest of the analysis.

**Reading the Data**

data <- read.csv("DA Career/Projects/5 Employee Attrition Analysis/dataset/IBM\_HR\_Employee\_Attrition.csv")

head(data)

* **read.csv()**: This reads your dataset from the specified file path into R as a data frame. A data frame is a table-like structure in R, where each column can contain different types of data (e.g., numeric, character).
* **head(data)**: Shows the first six rows of the dataset so you can quickly check its structure. This is important for getting a sense of the types of variables and their values.

**Step 2: Exploratory Data Analysis (EDA)**

**Summary Statistics**

summary(data)

* **summary(data)**: Provides summary statistics for the numeric columns in your dataset (e.g., min, max, mean, median, quartiles). This is helpful for detecting outliers, skewness, or unusual data patterns. For instance, you can see if the average Age is around the mid-30s, or if there are unusually high values for MonthlyIncome.

**Checking for Missing Values**

colSums(is.na(data))

* **is.na(data)**: Creates a logical matrix where TRUE means a value is missing (NA), and FALSE means it is not missing.
* **colSums()**: Adds up all the TRUE values in each column to give the total number of missing values per column. This step helps to detect whether any variables have missing values that need to be addressed (either by removing or imputing missing data).

**Checking Data Structure**

str(data)

* **str(data)**: Displays the structure of the data frame. It shows the number of rows and columns, and for each column, the variable type (e.g., integer, factor, numeric). This helps you identify whether categorical variables (like Attrition, Gender) need to be converted to factors for analysis.

**Step 3: Data Cleaning**

**Handling Missing Data**

data <- na.omit(data)

* **na.omit(data)**: Removes any rows that contain missing values (NA). In some cases, missing data can affect model accuracy or cause errors, so this function ensures that the analysis uses only complete cases. However, if you have a lot of missing data, you may want to impute the missing values instead of removing them.

**Converting Variables to Factors**

data$Attrition <- as.factor(data$Attrition)

data$Gender <- as.factor(data$Gender)

data$Department <- as.factor(data$Department)

data$JobRole <- as.factor(data$JobRole)

data$MaritalStatus <- as.factor(data$MaritalStatus)

data$OverTime <- as.factor(data$OverTime)

* **as.factor()**: Converts these columns to categorical variables (factors). This is important for machine learning algorithms, as they handle factors differently from continuous numeric variables. Factors are treated as discrete groups with no inherent order.
  + For example, Attrition (Yes/No) should be a factor, not numeric, because it's a categorical outcome (whether an employee left or stayed).
  + Similarly, Gender, Department, JobRole, etc., are all categorical and should be factors.

**Verifying Changes**

str(data)

* After conversion, the str(data) function helps verify that the variables have been properly converted to factors.

**Step 4: Exploratory Data Visualization**

**Age Distribution**

ggplot(data, aes(x = Age)) +

geom\_histogram(binwidth = 2, fill = "blue", color = "black") +

theme\_minimal() +

labs(title = "Distribution of Employee Age", x = "Age", y = "Count")

* **ggplot()**: A core function of ggplot2 to create plots. The aes() function defines the aesthetic mappings, i.e., what variables to plot on which axes.
  + In this case, aes(x = Age) maps Age to the x-axis.
* **geom\_histogram()**: Creates a histogram of Age to show the distribution of employees' ages. The binwidth = 2 argument controls the width of the histogram bins (each bar will represent a 2-year age range).
* **theme\_minimal()**: Applies a minimal theme to the plot, which removes unnecessary gridlines and keeps the focus on the data.
* **labs()**: Adds a title and axis labels to the plot. This helps make the plot easier to understand and interpret.

**Boxplot of Monthly Income vs Attrition**

ggplot(data, aes(x = Attrition, y = MonthlyIncome, fill = Attrition)) +

geom\_boxplot() +

theme\_minimal() +

labs(title = "Monthly Income by Attrition", x = "Attrition", y = "Monthly Income")

* **Boxplot**: Shows the distribution of MonthlyIncome for both groups of Attrition (Yes and No). Boxplots help identify the central tendency (median), spread (interquartile range), and any potential outliers in income for employees who stayed versus left.
* **fill = Attrition**: Colors the boxplot differently for employees who left and those who stayed.

**Step 5: Correlation Analysis**

**Correlation Matrix**

corr\_matrix <- cor(data %>% select(Age, MonthlyIncome, JobLevel, NumCompaniesWorked, TotalWorkingYears))

corrplot(corr\_matrix, method = "number")

* **select()**: Selects the relevant numerical columns for correlation analysis. Here, you're examining how Age, MonthlyIncome, JobLevel, NumCompaniesWorked, and TotalWorkingYears relate to each other.
* **cor()**: Computes the correlation matrix, which measures the strength of the linear relationship between each pair of selected variables. The correlation value ranges from -1 (perfect negative correlation) to +1 (perfect positive correlation). A value of 0 means no correlation.
* **corrplot()**: Visualizes the correlation matrix, where each cell represents the correlation between two variables. This helps quickly identify strong correlations (positive or negative) and informs feature selection for the model.

**Step 6: Hypothesis Testing**

**Chi-Square Test between Job Role and Attrition**

chisq\_test\_jobrole <- chisq.test(table(data$JobRole, data$Attrition))

* **chisq.test()**: Conducts a Chi-Square test of independence between two categorical variables (JobRole and Attrition). The null hypothesis is that there's no relationship between job role and attrition (i.e., job role does not influence whether an employee leaves). A p-value less than 0.05 typically indicates a significant relationship.

**T-Test between Monthly Income and Attrition**

t\_test\_income <- t.test(MonthlyIncome ~ Attrition, data = data)

* **t.test()**: Performs an independent two-sample t-test to compare the means of MonthlyIncome for employees who left (Attrition = Yes) versus those who stayed (Attrition = No).
  + The null hypothesis here is that there is no difference in mean income between the two groups.
  + A low p-value suggests that income significantly impacts whether an employee stays or leaves.

**Step 7: Building Logistic Regression Model**

**Splitting the Data**

train\_indices <- sample(1:nrow(data), 0.8 \* nrow(data))

train\_data <- data[train\_indices, ]

test\_data <- data[-train\_indices, ]

* **sample()**: Randomly splits the data into two parts. Here, 80% of the data is used for training the model, and the remaining 20% is used for testing. This ensures the model is trained on one set and evaluated on a separate set to avoid overfitting.

**Logistic Regression Model**

model <- glm(Attrition ~ Age + MonthlyIncome + JobRole + Department + MaritalStatus,

family = "binomial", data = train\_data)

* **glm()**: Fits a generalized linear model (GLM), specifically logistic regression, with Attrition as the dependent variable and Age, MonthlyIncome, JobRole, Department, and MaritalStatus as independent variables. The family = "binomial" specifies that this is a binary outcome (attrition: Yes or No).
* This model helps predict the probability of an employee leaving (attrition) based on the selected features.

**Step 8: Model Evaluation**

**Confusion Matrix**

conf\_matrix <- table(test\_data$PredictedAttrition, test\_data$Attrition)

* **table()**: Creates a confusion matrix, comparing the predicted attrition status (PredictedAttrition) to the actual values in the test data (Attrition). The matrix helps assess model performance by showing the number of true positives, true negatives, false positives, and false negatives.

**Step 9: Model Performance**

**ROC Curve**

roc\_curve <- roc(test\_data$Attrition, as.numeric(predictions))

plot(roc\_curve, main = "ROC Curve")

* **roc()**: Computes the ROC (Receiver Operating Characteristic) curve, which evaluates the performance of a classification model at various thresholds. The ROC curve helps you understand the trade-off between sensitivity (True Positive Rate) and specificity (False Positive Rate).

**Precision-Recall Curve**

pr\_curve <- performance(prediction(predictions, test\_data$Attrition), "prec", "rec")

plot(pr\_curve, main = "Precision-Recall Curve")

* **performance()**: Computes precision and recall at various thresholds and plots the Precision-Recall curve. This is useful for understanding how well the model performs when dealing with imbalanced classes (e.g., more employees staying than leaving).

Here's a summary of the key findings from the data analysis:

**1. Exploratory Data Analysis (EDA) Insights:**

* **Age Distribution**: The age of employees in the dataset ranges from 18 to 60, with a mean age of 36.92 years. The histogram of employee age would show the distribution, where most employees are clustered around the 30-40 age range.
* **Attrition vs Monthly Income**: The boxplot comparing Monthly Income and Attrition reveals that employees who leave the company (Attrition = Yes) tend to have lower monthly incomes than those who stay.
* **Attrition by Job Role**: There is a noticeable difference in attrition rates across different job roles. Some roles might have a higher proportion of employees leaving.
* **Gender and Job Role**: The distribution of job roles by gender shows how male and female employees are distributed across different roles. It's useful to see if there are gender-based patterns in specific job roles.
* **Job Satisfaction and Attrition**: There’s an indication that employees with lower job satisfaction are more likely to leave. The bar plot shows the correlation between job satisfaction levels and attrition rates.

**2. Statistical Tests:**

* **Chi-Square Test**: A Pearson’s Chi-Square test between Job Role and Attrition shows a significant relationship (p-value = 2.752e-15), meaning that job roles significantly influence employee attrition.
* **T-Test**: The T-test comparing Monthly Income by Attrition shows a significant difference in monthly income between employees who stayed and those who left. The average income of employees who stayed (mean = 6832.74) is significantly higher than those who left (mean = 4787.09). The p-value is very low (4.434e-13), confirming this result.

**3. Logistic Regression Model:**

* The logistic regression model predicts attrition based on variables such as Age, Monthly Income, Job Role, Department, and Marital Status. Here are some key results:
  + **Age**: Age has a significant negative effect on attrition (p-value = 0.00112). Older employees are less likely to leave.
  + **Monthly Income**: Monthly income does not significantly influence attrition in this model (p-value = 0.47422).
  + **Job Roles**:
    - **Laboratory Technician**: Employees in the Laboratory Technician role are more likely to leave compared to other roles (p-value = 0.00397).
    - Other job roles like Human Resources, Manufacturing Director, and Research Scientist don’t show significant effects on attrition.
  + **Department**: Employees in the Research & Development and Sales departments also don't show significant differences in attrition rates.
  + **Marital Status**: Marital status does not seem to significantly affect attrition either.

**4. Visualizations:**

* **Job Role Distribution**: The bar chart of job roles shows the distribution across different roles. A significant number of employees are in roles like Laboratory Technicians and Research Scientists.
* **Department Distribution**: The distribution of departments shows that the majority of employees work in Research & Development and Sales.
* **Attrition by Job Level**: Attrition varies with job level, with lower levels showing higher attrition rates.

**Conclusion:**

* **Age** and **Job Role** are significant factors in predicting employee attrition.
* The **Monthly Income** difference between employees who leave and stay is significant, but the logistic model doesn't find a strong predictive power for Monthly Income.
* The Logistic Regression model shows that older employees tend to stay longer, while certain job roles, like Laboratory Technicians, have a higher risk of attrition.