TABLE OF CONTENTS

IBM Attrition Dataset Study and Prediction Report

Dataset Description

Dataset Used: *IBM HR Analytics Employee Attrition & Performance from kaggle.*

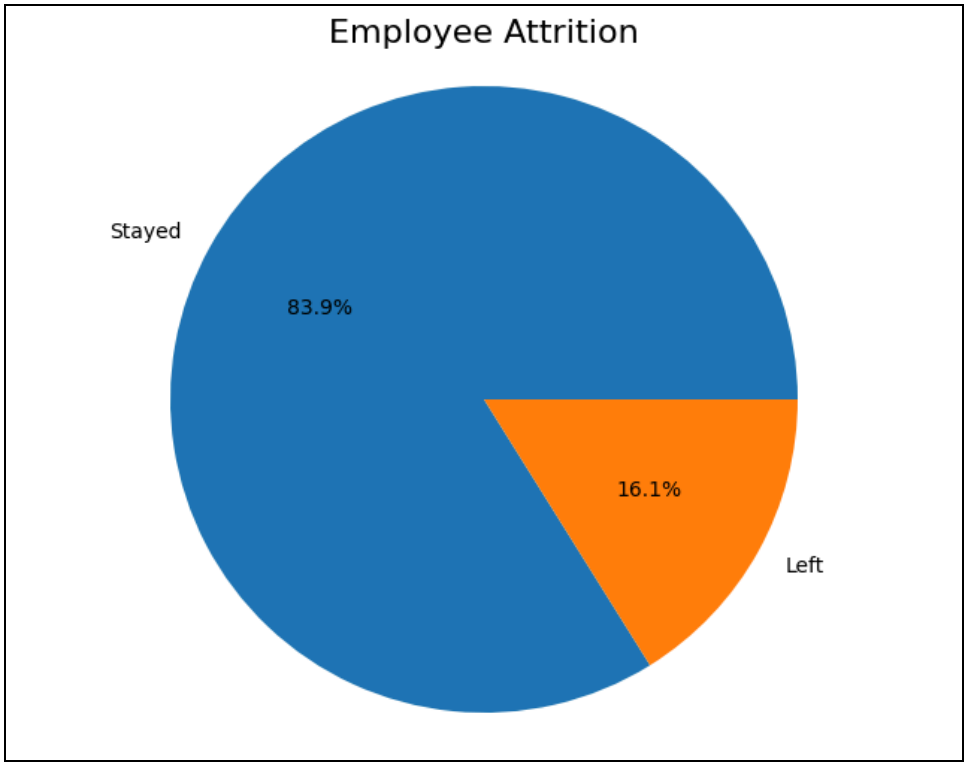
Description: The IBM HR Analytics Employee Attrition & Performance dataset offers a comprehensive view of various factors influencing employee dynamics, including demographic information, job roles, performance metrics, satisfaction levels, and reasons for attrition. Analyzing this rich dataset allows organizations to delve deeper into understanding the root causes behind attrition, whether it be related to job dissatisfaction, lack of career advancement opportunities, or issues within the organizational culture.
Moreover, by correlating performance data with attrition rates, companies can identify traits and behaviors associated with high-performing employees, enabling them to refine recruitment processes and talent development strategies. This holistic approach to workforce analytics not only helps in retaining top talent but also enhances overall organizational efficiency and competitiveness. Additionally, by leveraging advanced analytics techniques such as machine learning and predictive modeling, HR professionals can forecast future attrition trends and proactively implement interventions to mitigate potential risks, thereby ensuring sustained business success and employee satisfaction.

Dataset Analysis

Through dataset analysis, organizations can harness the power of data visualization techniques to uncover intricate patterns and insights hidden within vast volumes of information. Data visualization serves as a powerful tool for transforming complex datasets into easily understandable visual representations, allowing stakeholders to gain valuable insights at a glance. By creating visually appealing graphs, charts, and dashboards, analysts can effectively communicate trends, correlations, and anomalies within the data, facilitating informed decision-making processes across various departments within the organization. These visualizations not only enhance comprehension but also enable stakeholders to identify outliers or irregularities that may require further investigation or intervention.

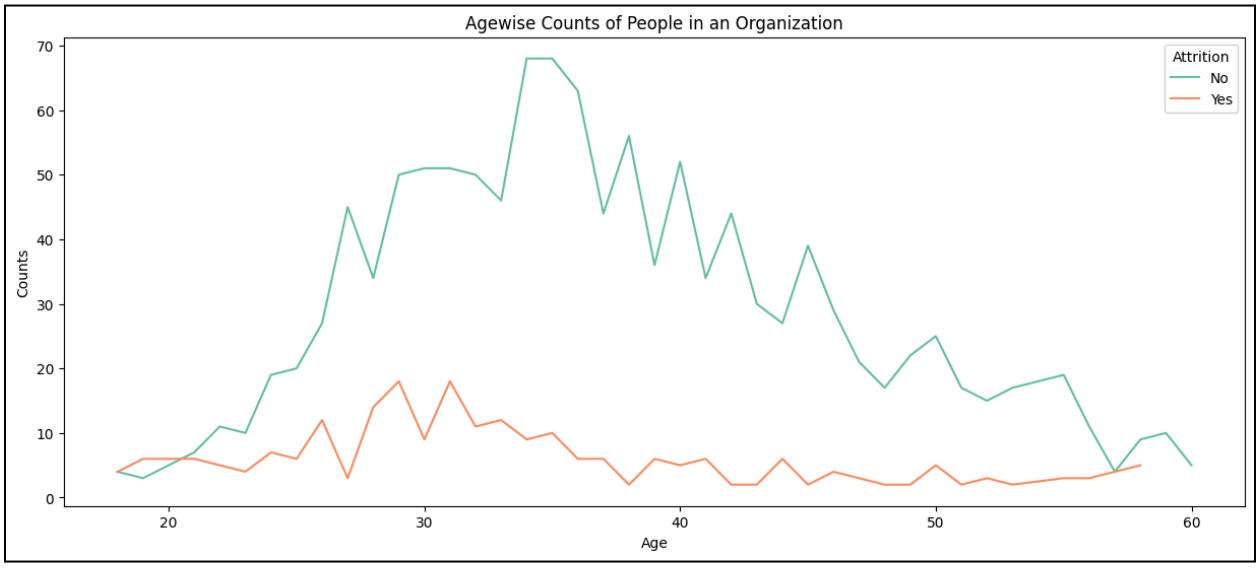
Various Plots Obtained:

1. **Pie Chart to show overall Employee Attrition**



Inference: This pie chart shows that 83.9% of the people that worked with the company chose to stay and continue the work while 16.1% people left the company.

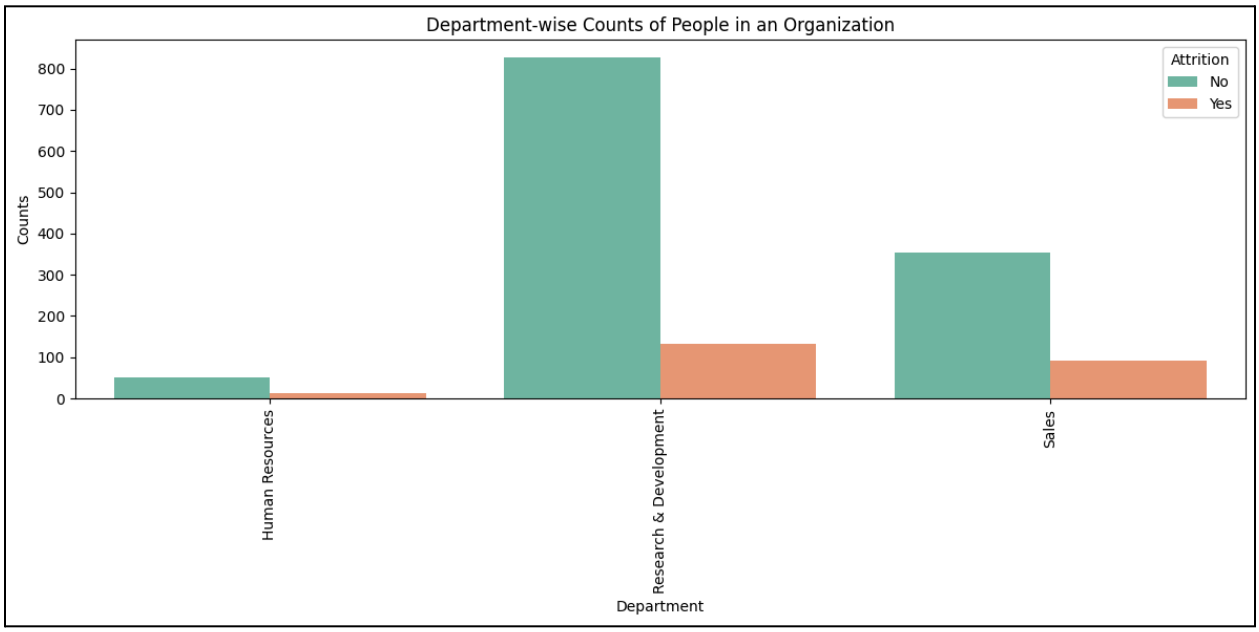
2. **Age Wise count of people**



Inference: This Line plot clearly shows that the volume of people that chose to stay in the company between the age group of 30-40 is significantly more than the people who chose to

leave the company in the same age group. With this data, it is clearly evident that people in the age group of 30-40 are finding somethings undesirable in the company which is prompting them to leave.
Also, with this line plot we may infer that most of the workforce of the company comprises of people in the age group of 20-45 after which the workforce declines.

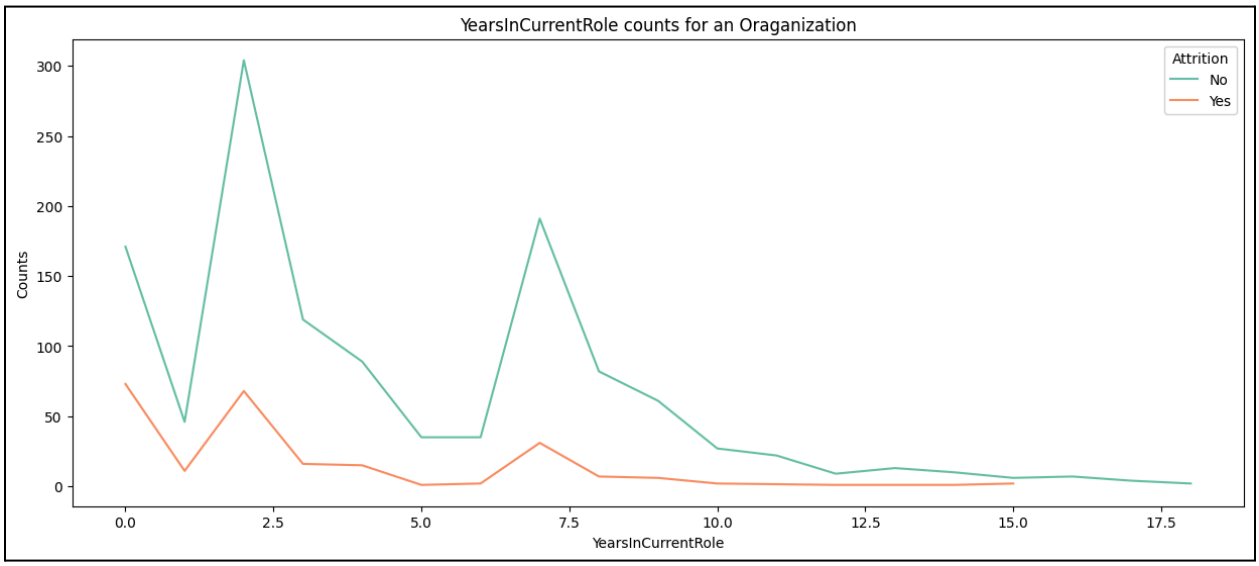
3. Department wise count



Inference:
This bar plot shows two very important things:

- First, the number of employees in the R&D department is significantly more than the number of people in the other departments like HR and Sales.
- Second, percent wise there are 20.63% employees in the sales department who departed from the company for some reason or another, this is the highest number (per-cent wise) among all the other departments.

4. **Years in Current Role Count**



Inference: This Line plot shows very clearly that in the long term very few people left the job. More precisely after working for more than 7.5 years in a role the employees are less likely to leave the job.
On the other hand though, people who worked between 0 - 2.5 years in a role are more likely to be departed from the job. This trend spiked in the neighboring areas of the 7.5 years mark.

Data Preprocessing

Below are the detailed steps which were taken when preprocessing the data:

1. **Identify Unique Values per Column**:
   - Iterate through each column in the dataset.
   - Print the name of each column along with the number of unique values it contains.

2. **Encode the 'Attrition' Column**:
   - Convert the 'Attrition' column from categorical (containing 'Yes' and 'No') to numerical by using the `factorize()` method. This method assigns a unique numeric label to each unique category in the column. The labels are based on the order of appearance of the categories in the column.

3. **Drop Irrelevant Columns**:
   - Remove columns that are deemed irrelevant or have constant values across all rows. These columns include 'EmployeeCount', 'Over18', 'StandardHours', and 'EmployeeNumber'. This step helps in reducing noise and streamlining the dataset.

4. **Identify Categorical and Numeric Data**:

- Separate the dataset into two subsets based on data type: numeric and categorical. This is done using the `select_dtypes()` method, specifying the data types 'int64' and 'float64' for numeric data, and 'object' for categorical data.

5. **Convert Categorical Variables to Dummy Variables**:
   - Identify categorical columns with fewer than 20 unique values, which are suitable for one-hot encoding.
   - Use one-hot encoding (`pd.get_dummies()`) to convert categorical variables into dummy variables. This process creates binary (0 or 1) columns for each unique category within the original categorical columns. The `drop_first=True` parameter is used to drop the first category from each encoded column to avoid multicollinearity issues in subsequent analyses. The `dtype='uint8'` parameter specifies the data type of the dummy variables as unsigned 8-bit integers, which optimizes memory usage.

These preprocessing steps help prepare the dataset for further analysis, i.e., predictive modeling, by ensuring that the data is appropriately formatted and encoded for various Machine Learning algorithms to use.

Model Development

1. **Data Loading and Preparation**:
   - The dataset containing information on employee attrition and related attributes was loaded into a DataFrame named 'df'.

2. **Data Splitting**:
   - The dataset was split into training and testing sets using the `train_test_split` function from scikit-learn.
   - Features were assigned to X, excluding the target variable 'Attrition', while the 'Attrition' column was assigned to the target variable y.
   - The splitting was stratified based on the target variable to ensure that the distribution of classes ('Attrition' = Yes/No) in the training and testing sets is similar.

3. **Feature Scaling**:
   - Standardization was applied to the numerical features in both the training and testing sets using the `StandardScaler` from scikit-learn.
   - The `fit_transform` method was used to fit the scaler to the training data and transform it, ensuring that the scaling parameters are learned from the training set and applied to both sets consistently.
   - The `transform` method was applied to the testing set to scale its features based on the parameters learned from the training set.

4. **Model Training and Evaluation**:

Several classification models were selected for training, including Logistic Regression, Random Forest Classifier, Support Vector Classifier (SVC), XGBoost Classifier, Decision Tree Classifier, Gaussian Naive Bayes, and K-Nearest Neighbors Classifier.

- Each model was instantiated and trained on the scaled training data using the `fit` method.
- Predictions were made on the scaled testing data using the `predict` method.
- Accuracy and ROC AUC score were calculated for each model using the predicted values and the actual target values from the testing set.
- Model performance metrics (accuracy and ROC AUC) were stored in lists for further analysis.

5. **Model Comparison**:
   - The accuracy and ROC AUC scores obtained from each model were compiled into a DataFrame named 'model_comparison'.
   - The DataFrame was sorted in descending order based on accuracy to identify the best-performing models.

Evaluation Results and Optimizations

Model Optimization and Training Process Report

1. **Hyperparameter Tuning**:
   - For each model, hyperparameters were specified along with their possible values in a list named 'hyperparameters'.
   - Hyperparameters were tuned using the `RandomizedSearchCV` method from scikit-learn, which performs randomized search over specified parameter values.
   - A loop was used to iterate over each model and corresponding hyperparameters, applying randomized search with 10 iterations and 5-fold cross-validation to find the best combination of hyperparameters.
   - The best model, its corresponding score, and the best parameters were printed for each model.

The hyperparameters used for tuning each model are as follows:

- **Logistic Regression**:
  - `**penalty**`: Specifies the norm used in the penalization ('l1' for Lasso, 'l2' for Ridge).
  - `**C**`: Regularization parameter. Controls the inverse of regularization strength.

- **Random Forest Classifier**:
  - `**n_estimators**`: Number of trees in the forest.
  - `**max_features**`: Number of features to consider when looking for the best split.

- **Support Vector Classifier (SVC)**:

- `**kernel**`: Specifies the kernel type to be used in the algorithm ('linear', 'poly', 'rbf', 'sigmoid').
- `**C**`: Regularization parameter. Controls the trade-off between a smooth decision boundary and classifying the training points correctly.

- **XGBoost Classifier**:
  - `**max_depth**`: Maximum depth of a tree.
  - `**learning_rate**`: Boosting learning rate.

- **Decision Tree Classifier**:
  - `**criterion**`: Function to measure the quality of a split ('gini' for Gini impurity, 'entropy' for information gain).
  - `**max_depth**`: Maximum depth of the tree.

- **K-Nearest Neighbors Classifier**:
  - `**n_neighbors**`: Number of neighbors to consider.
  - `**weights**`: Weight function used in prediction ('uniform' for all points in each neighborhood to have equal weight, 'distance' for points weighted by the inverse of their distance).

2. **Model Training and Evaluation**:
- After tuning the hyperparameters, models were trained using the best parameter settings obtained from the tuning process.
- The accuracy, ROC AUC score, and F1 score were calculated for each model using the testing set.
- Additionally, confusion matrices were generated to visualize the performance of each model in predicting employee attrition.

Summary

Summary Report on Employee Attrition Analysis

Findings:

1. **Data Insights**:
- A considerable portion of the workforce resides within a 10-mile radius of the company, suggesting a preference for shorter commutes.
- Approximately 67% of employees hold Bachelor's and Master's degrees, indicating a highly educated workforce.
- A significant 75% of employees demonstrate high levels of job involvement, highlighting strong engagement with their roles.
- Job satisfaction is reported by 60% of employees, indicating overall contentment within the workforce.
- Around one-third of employees have only worked for one company, implying a tendency towards long-term commitment.

- Work-life balance is rated positively, with 60% of employees rating it as 3.0 and a notable portion rating it as 4.0.
- A decline in employee numbers is observed as years since the last promotion increase, with 73% receiving a promotion within the last two years.

Insights Gained:

1. **Predictive Modeling**:
   - Multiple classification models were trained and evaluated to predict employee attrition.
   - Models were optimized using hyperparameter tuning techniques to improve performance.
   - Evaluation metrics such as accuracy, ROC AUC score, and F1 score were utilized to assess model effectiveness.

Challenges Encountered:

1. **Data Quality Issues**:
   - Incomplete or inaccurate data may have impacted model performance.
   - Imbalanced class distribution in the target variable 'Attrition' may have influenced model training and evaluation.

Recommendations for Reducing Employee Attrition:

1. **Enhance Employee Engagement**:
   - Foster a supportive work environment to increase job satisfaction and job involvement.
   - Provide opportunities for skill development and career advancement to promote employee growth and retention.

2. **Improve Work-Life Balance**:
   - Implement policies and initiatives to support a healthy work-life balance for employees.
   - Offer flexible work arrangements and wellness programs to reduce stress and burnout.

3. **Address Compensation and Benefits**:
   - Conduct regular salary reviews and ensure competitive compensation packages to retain top talent.
   - Provide attractive benefits such as healthcare, retirement plans, and bonuses to incentivize employee loyalty.

5. **Invest in Talent Development**:
   - Provide opportunities for continuous learning and skill enhancement through training programs and workshops.
   - Recognize and reward employee achievements to motivate and retain valuable talent within the organization.

6. **Monitor and Analyze Attrition Trends**:
   - Regularly analyze employee attrition data to identify patterns, trends, and potential risk factors.

- Use predictive analytics and machine learning techniques to forecast attrition and proactively implement retention strategies.

By implementing these recommendations, organizations can effectively reduce employee attrition rates, enhance employee satisfaction and retention, and foster a positive and productive work culture.

Thank you!