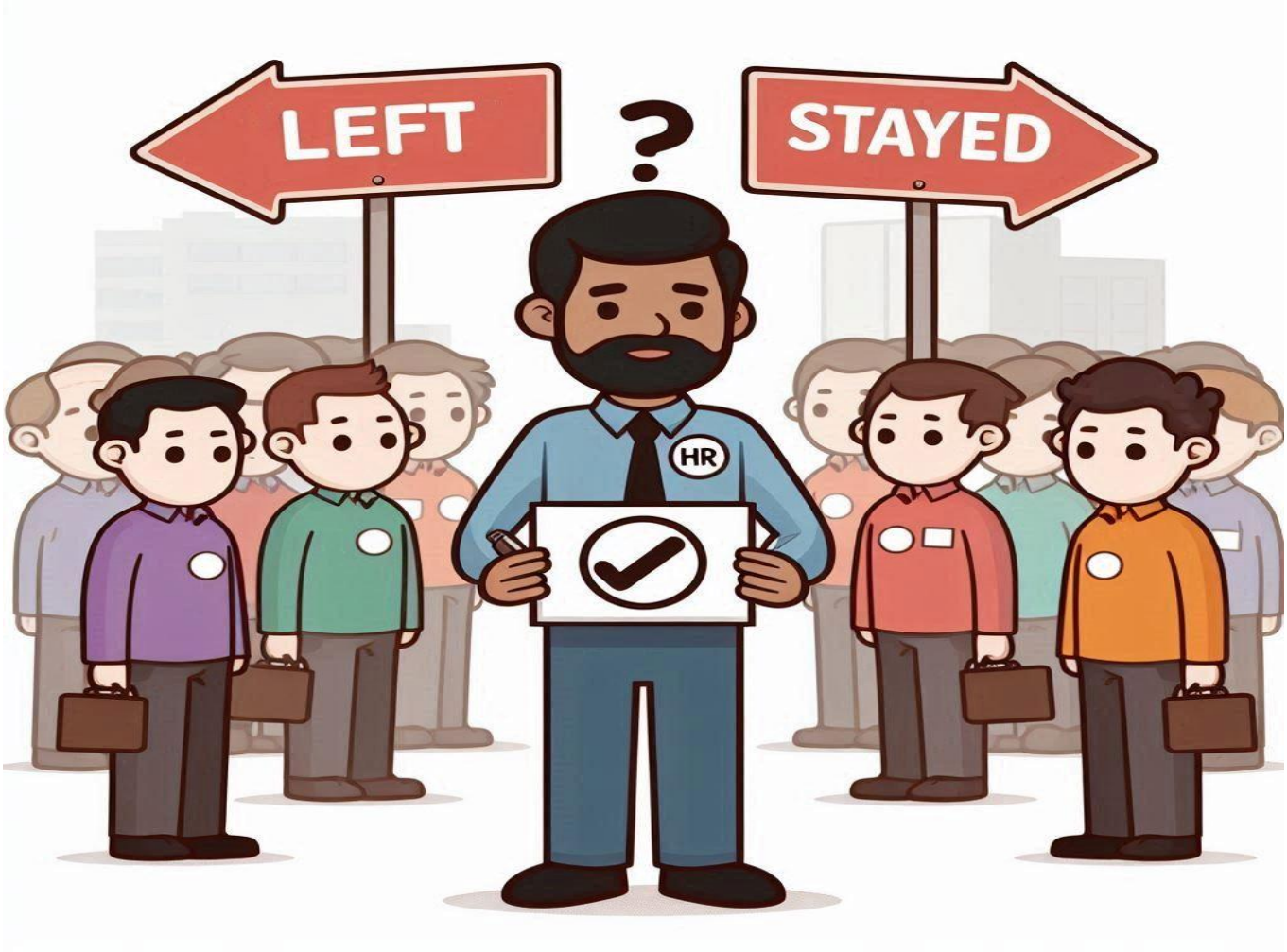
Data Science – Project Overview

Employee Attrition Analysis

Kfir Tayar



What is Employee Attrition?

Employee attrition (also called employee churn) is when employees leave a company over time. It is measured as a percentage of the total workforce that leaves in a given year. Employees may leave voluntarily (like retiring or finding a new job) or involuntarily (such as company layoffs or mergers).

The standard formula to calculate rate of employee attrition is:

$$\left(\frac{\text{Number of employees who left}}{\text{Total employees in the same period}}\right) \times 100$$

A healthy attrition rate is typically 10% or lower. If it is higher, companies need to analyze why employees are leaving and take action to retain them.

Types of Employee Attrition:

Employee attrition can be categorized into different types based on the nature of employee departures:

- **Voluntary Attrition**: Employees leave by choice due to better job opportunities, retirement, personal reasons, or dissatisfaction with the workplace.
- **Involuntary Attrition**: Employees are let go due to company layoffs, restructuring, or performance-related dismissals.
- **Internal Attrition**: Employees transfer to different roles or departments within the same organization, potentially creating skill gaps in specific teams.
- **Demographic-Specific Attrition**: Certain employee groups, such as those of a particular age, gender, or background, leave at higher rates, indicating potential inclusivity or workplace issues.

The common factors that lead to Attrition:

Understanding the reasons behind employee attrition is crucial for organizations aiming to improve retention. Key factors contributing to attrition include:

- **Inadequate Compensation and Benefits**: Employees often seek better pay and benefits elsewhere when they feel their current compensation doesn't align with their skills or industry standards.
- **Limited Career Development Opportunities**: A lack of clear paths for advancement or professional growth can lead employees to pursue opportunities with better prospects.
- **Poor Work-Life Balance**: Rigid schedules and excessive workloads can cause burnout, prompting employees to leave in search of more flexible arrangements.
- **Unsatisfactory Working Conditions**: An unhealthy or unsafe work environment can diminish job satisfaction and drive employees away.
- **Lack of Recognition**: When employees feel their efforts go unnoticed, their motivation decreases, increasing the likelihood of departure.
- **Management Issues**: Poor leadership, inadequate communication, and unresolved conflicts with supervisors can lead to dissatisfaction and attrition.
- **Personal Reasons**: Factors such as health issues, family commitments, or relocation can also contribute to an employee's decision to leave.

Addressing these factors proactively can help organizations reduce attrition rates and foster a more engaged and stable workforce.

The Importance of Preventing Employee Attrition:

While some attrition is inevitable, high attrition rates can harm a business by increasing costs, causing talent loss, and damaging the company’s reputation. Managing attrition effectively is crucial for maintaining workplace stability, employee well-being, and overall business success.

Sources:

https://web.hr/glossary/attrition

https://www.legislate.ai/blog/what-is-employee-attrition-and-how-to-prevent-it

https://techrseries.com/featured/biggest-factors-that-lead-to-employee-attrition/

Is an employee likely to leave the company?

One of the key questions in workforce management is: Is an employee likely to leave the company? Understanding and predicting employee attrition can help businesses take proactive steps to retain talent.

What I aim to predict is whether an employee will stay or leave based on an existing database of employee features. This model may help companies estimate attrition rates for a given period.

Let's step forward to the process.

Dataset:

The dataset is sourced from Kaggle and includes various features related to employee demographics, work experience, and job satisfaction.

- The dataset is synthetic, yet it presents realistic employee profiles.
- The dataset contains 74,498 samples.
- The dataset is split into training and testing sets.
- The dataset goal is to analyze the factors influencing attrition and predict which employees are at risk of leaving.

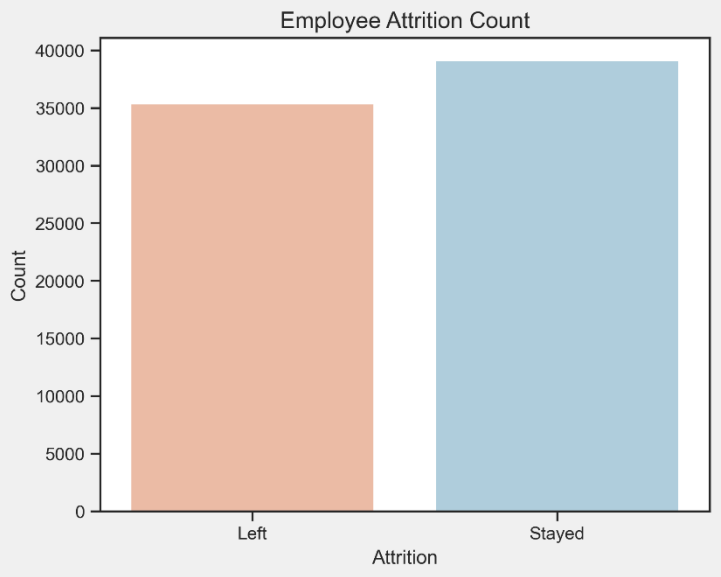
Date prep:

In this section, I combine the training and testing sets into a single DataFrame. I also clean the 'Education Level' feature by removing punctuation, transform the 'Age' feature into an 'Age Group', and create a new feature called 'Start Age', which represents the age at which an employee began working at the company. Additionally, I drop unnecessary features, such as 'Company Tenure', which doesn't align well with the data (as over 50% of its values exceed the employee's age).

```
len(df[df['Company Tenure'] > df['Age']]) # -> 52902
```

EDA:

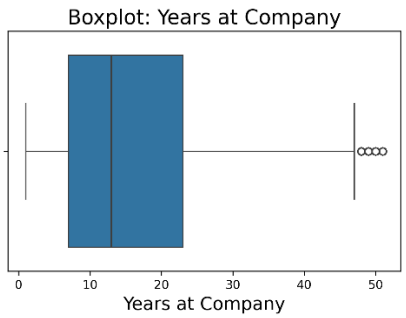
Once the data preparation file was completed, I began the exploratory data analysis (EDA) process. I generated data protocol files and an AutoViz report, examined the skewness of continuous features, and visualized the numerical and categorical features using histograms and count plots. After these visualizations, and since all continuous features were normally distributed, I conducted Pearson correlation analysis, an ANOVA test, and a Chi-Square test.

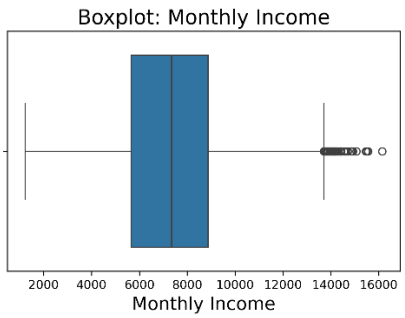


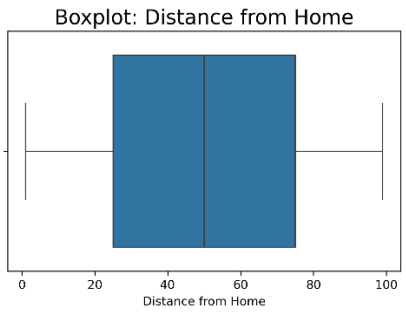
This count plot represents the ratio of the Attrition (target feature).

Data Cleansing:

After completing the EDA, I found no missing values in the DataFrame. Next, I checked for outliers using the IQR method and identified outliers in two features: 'Years at Company' and 'Monthly Income'. The following box plots illustrate this:



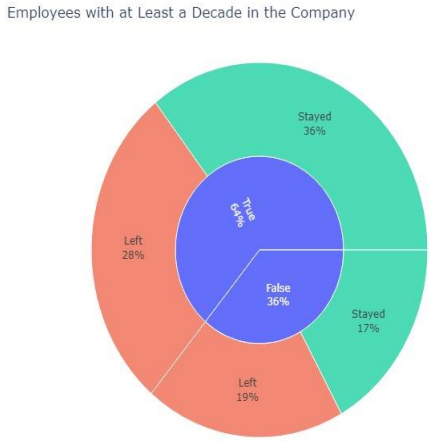


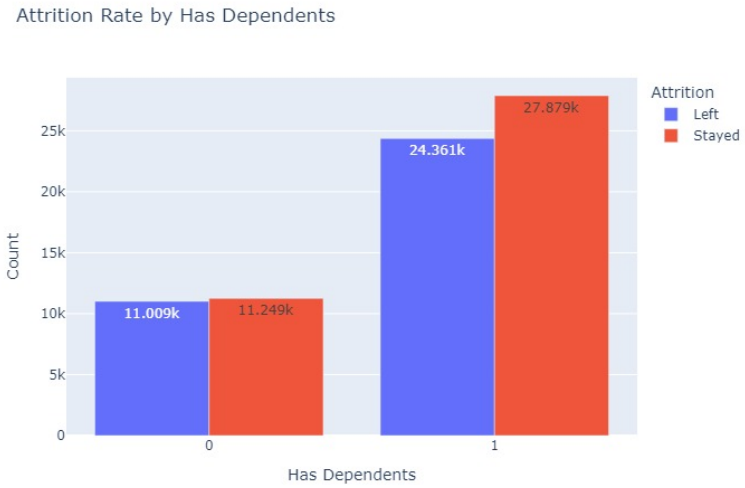


I replaced the outliers with NaN values and examined whether imputation was necessary. Since the distribution and correlations remained unchanged with and without the outliers, I concluded that imputation was not needed. As a result, I decided to retain the outliers in the DataFrame.

Feature Engineering:

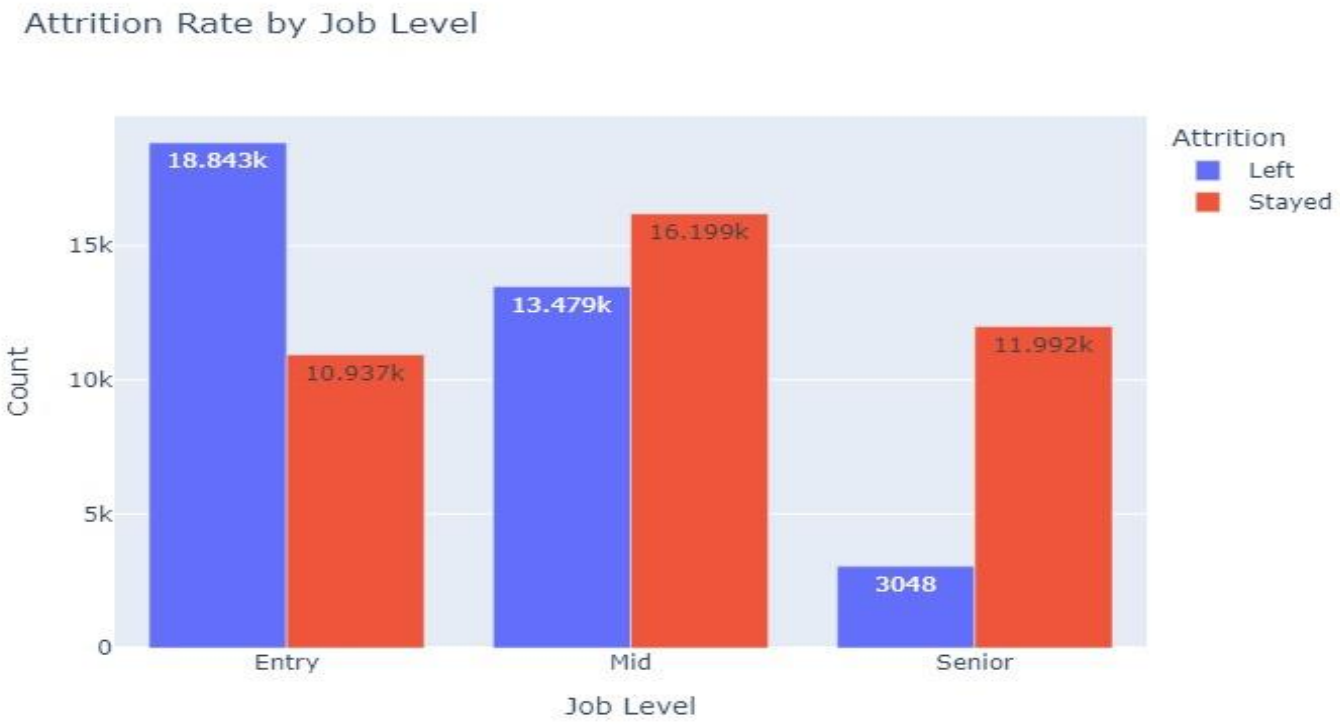
In this section, I added new features derived from the internal data and explored them. The following plots help illustrate the main patterns.





Based on these graphs, we can conclude that the longer an employee stays with the company, the higher the likelihood they will remain. Additionally, if the employee has children, the probability of staying increases even further.

Here, we can see that employees with a low 'Job Level' (Junior) are more likely to leave compared to senior employees. This supports the insights from the 'Has Decade' feature.

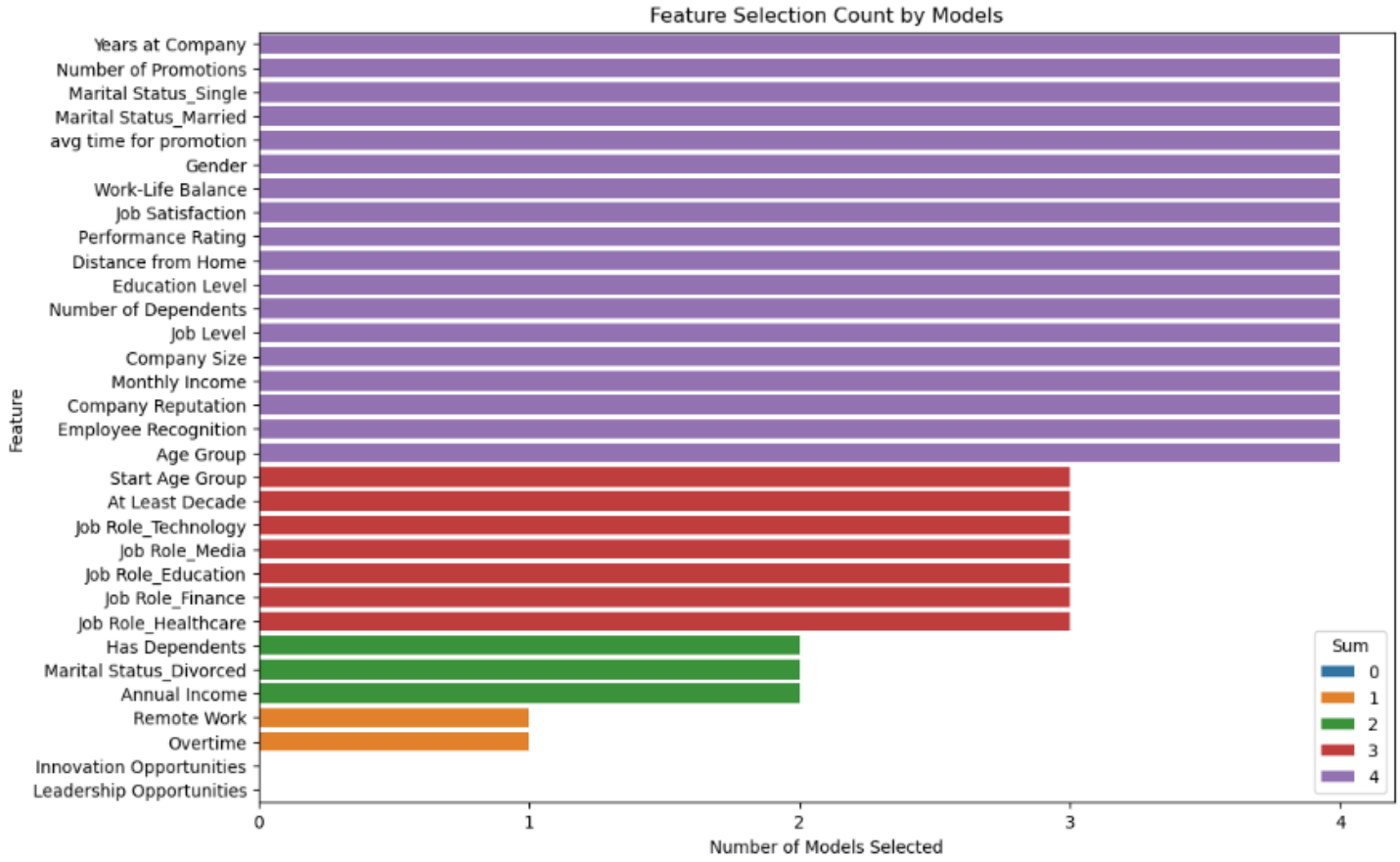


Data Encoding:

Application of data encoding techniques, such as One-Hot Encoding and Label Encoding, was performed to convert categorical variables into numerical representations. One-Hot Encoding was applied to features with no ordinal relationship, creating binary columns for each category. For features with an inherent order, Label Encoding was used, assigning a unique integer to each category. This transformation allows the model to effectively process categorical data.

Feature Selection & Normalization:

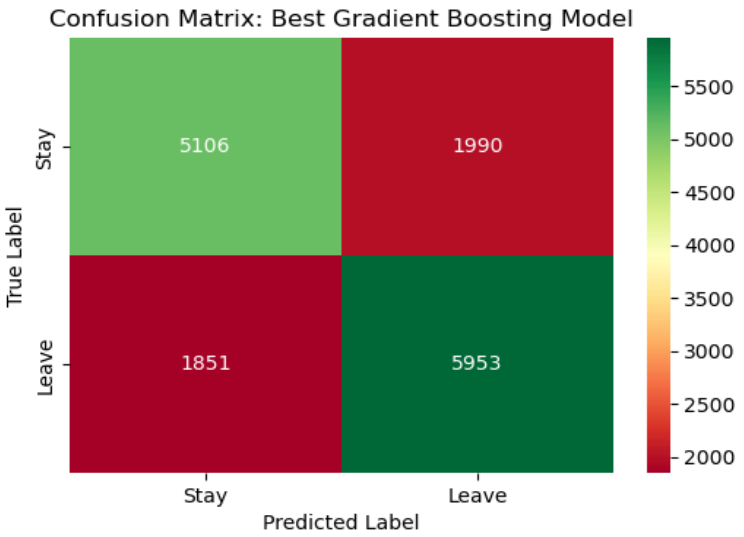
To select the most important features, I began by normalizing the continuous features using MinMax normalization. Then, I applied multiple models: **Lasso**, **Ridge**, **GradientBoostingClassifier**, **LogisticRegression**, and **XGBClassifier** to evaluate feature importance. The following plot illustrates the number of models that selected each feature:



I decided to set a threshold of 4, resulting in 19 features that were retained as the most relevant predictors for the model.

Model Selection & Fine Tuning:

In the final section, I split the DataFrame into training and testing sets and began evaluating classification models. After running the base models, I selected the Gradient Boosting model as it produced the best performance for my dataset. Next, I performed fine-tuning on the model using GridSearchCV with cross-validation. The final results are as follows:




```
Best Gradient Boosting Model Performance:
Accuracy: 0.7422147651006712
F1 Score: 0.7560805232742744
Recall: 0.7628139415684264
Precision: 0.749464937680977
```

Deployment:

The deployment of this model can significantly assist the HR team in various ways:

1. **Maintaining a Healthy Attrition Rate**: The HR team can continuously monitor employee factors and update the model to assess the likelihood of an employee leaving the company. By identifying high-risk employees early, they can take proactive measures to retain valuable talent and reduce unwanted attrition.
2. **Evaluating New Candidates**: The talent acquisition team can leverage the model to assess the potential risk of a new hire leaving the company prematurely. By evaluating the candidate's profile against the model, the team can better understand if the candidate is likely to stay long-term, saving the company from investing resources in high-risk hires.

This model can provide actionable insights that help HR teams make informed decisions, ultimately contributing to a more stable and cost-effective workforce.

In conclusion, this model provides valuable insights for HR teams to predict employee attrition and make informed decisions regarding retention and recruitment. By leveraging predictive analytics, organizations can proactively address potential issues, reduce turnover costs, and maintain a stable workforce.

Thank you

Kfir Tayar

Project Notebooks:

01_Employee_Attrition_Data_Preparation.ipynb

02_Employee_Attrition_Exploratory_Data_Analysis.ipynb

03_Employee_Attrition_Data_Cleansing.ipynb

04_Employee_Attrition_Feature_Engineering.ipynb

05_Employee_Attrition_Data_Encoding.ipynb

06_Employee_Attrition_Feature_Selection.ipynb

07_Employee_Attrition_Model_Selection_and_Fine_Tuning.ipynb

Note:
Throughout this project, I have created several utility functions for convenience and reuse. These functions are located in the utils directory and are designed to simplify the code by promoting clean, modular, and maintainable practices. By centralizing reusable code in this directory, it becomes easier to apply these functions to future projects, improving efficiency and reducing redundant code.