Employee Exit Prediction Report

Adarsh and Bois
Roll Numbers: IMT2022123, IMT2022047, IMT2022069

November 15, 2024

Project Repository

You can access the project on GitHub: `https://github.com/mrvandana1/ML-Project`

1 Data Exploration and Visualization

Exploratory Data Analysis (EDA)

Distribution of the Target Variable

To understand the distribution of the target variable `exit_status`, a count plot was generated. This plot revealed the class distribution, showing how many employees stayed versus left the company.

- **Visualization**: Count plot of `exit_status`.
- **Insight**: We observed the number of employees who left the company and those who stayed.

Employee Exit Analysis: Income, Tenure, and Age

We performed a detailed analysis of three key factors — `monthly_income`, `years_at_company`, and `age` — to assess their relationship with employee exit behavior. The following visualizations provide insights into how these features correlate with the likelihood of employees staying or leaving the company.

Correlation Analysis

To further explore the relationships between the features and the target variable, a correlation matrix was computed using only numerical features. This correlation matrix was visualized using a heatmap.

- **Visualization**: Correlation matrix of numeric features.
- **Insight**: The heatmap helped to identify which numerical features are strongly correlated with the target variable (`exit_status`), which is crucial for feature selection and model building.

Summary of Unique Values in Each Column

In addition to the visualizations, we examined the number of unique values in each feature to understand the data better. This helps identify features that may need encoding or transformation.

Key Observations:

- The `exit_status` column has 2 unique values: `Stayed` and `Left`.
- Some categorical features, such as `gender`, `overtime`, `remote_work`, and `job_level`, have only 2 unique values, which indicates they can be directly label encoded.
- Continuous features such as `age`, `monthly_income`, and `years_at_company` have more variability, making them useful predictors in the model.

Conclusion

This exploratory data analysis (EDA) provided valuable insights into the dataset. Visualizations helped us understand the distribution of the target variable, relationships between key features, and potential correlations with employee exit behavior. The correlation matrix revealed important relationships between numeric features and the target variable. This exploration is essential for informed feature selection and preprocessing steps before building predictive models.

Preprocessing Steps

Columns Used/Omitted

- **Used Columns**: We used all columns except for `response_id`. The `response_id` is a unique identifier that does not provide predictive value.

- **Omitted Columns**: The `response_id` column was omitted as it serves no predictive purpose.

Pre-Preprocessing

- **Handling Missing Values**: Missing data in numeric columns (e.g., `age`, `monthly_income`) was imputed using the median, while categorical columns (e.g., `gender`, `job_role`) were imputed using the most frequent value. This ensures that missing data does not introduce bias.
- **Feature Engineering**: A new feature, `promotion_rate`, was created by dividing the number of promotions by the number of years worked at the company, providing a better sense of career progression.
- **Feature Scaling**: Numeric features, including the newly engineered feature `promotion_rate`, were scaled using `StandardScaler` to ensure they contribute equally to the model and avoid bias from varying scales.
- **Encoding Categorical Features**: Categorical variables were encoded using `LabelEncoder` to convert them into numerical form suitable for machine learning models.
- **Outliers**1. Removing the outliers before running the models. 2. Keeping the outliers in the dataset and running the models without any removal.

  This allowed us to compare the impact of outlier removal on the model's performance and determine which preprocessing step yielded better results.

2. Models

Models used and Performance

- **Logistic Regression**:- It struggled to model complex, non-linear interactions, such as those involving combinations of categorical variables (e.g., job role and company size) or continuous variables (e.g., monthly income and distance from home). These relationships likely exist in the data but are beyond the model's linear scope.

- **Random Forest Classifier**: Achieved competitive performance across all metrics (accuracy, precision, recall). It was more robust than Logistic Regression due to its ability to model complex relationships.

- **Decision Tree Classifier**: In this project, we employed a Decision Tree Classifier as part of our machine learning pipeline to predict the `exit_status` of employees (target variable). The decision tree algorithm provided interpretable insights into the important factors contributing to employee exits.

- **XGBoost Classifier**: Outperformed the other models. After performing hyperparameter optimization using `GridSearchCV`, the model's performance was significantly improved. The grid search found the optimal parameters for `max_depth`, `learning_rate`, `n_estimators`, and other hyperparameters, resulting in a high-performing model.

Reasoning Behind Model Selection

- **XGBoost**: Selected for its high efficiency and superior performance on large datasets. Its ability to handle overfitting and produce accurate predictions through boosting was key to its success.

Model Evaluation

- Cross-Validation: Cross-validation was conducted to evaluate the model's generalization capability across unseen data. By splitting the dataset into multiple training and validation folds, we ensured robust performance metrics.

- Performance Metrics : The model's effectiveness was assessed using key evaluation metrics, including:

  - Accuracy - To measure the proportion of correctly classified instances.
  - Precision - To evaluate the model's ability to correctly identify positive predictions.

- Confusion Matrices : Confusion matrices were plotted to provide detailed insights into the model's performance, especially in scenarios involving imbalanced datasets. These visualizations highlighted the true positives, true negatives, false positives, and false negatives, aiding in better interpretation of the results.

- Best Performance : The model achieved a best cross-validated accuracy score of {`0.7544`}, demonstrating its capability to learn from the dataset while avoiding overfitting.