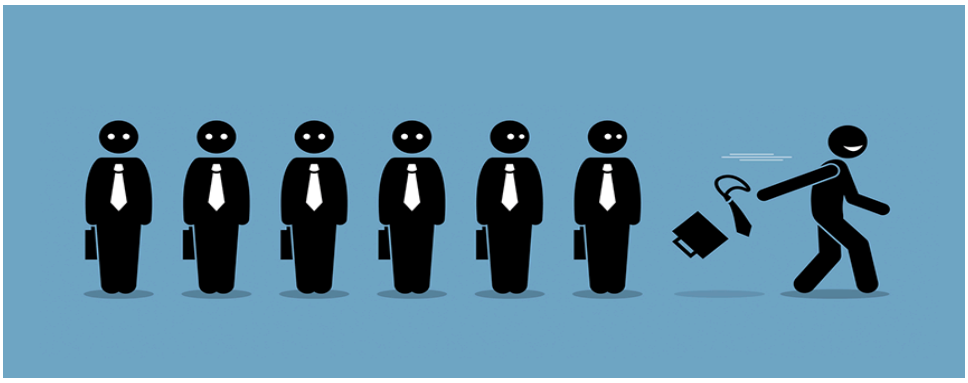
University of California, Irvine
Paul Merage School of Business

Employee Churn Prediction

Predicting what causes employees to resign and strategies on how to retain them



BANA 273: Machine Learning for Analytics
Instructor**:** Professor Mingdi Xin

Prepared by Team 12A:
Anna Haroutounian
Zhiwei Lu
Viraj Vijaywargiya
Yuh-Shin Yen
Danqi Zheng

December 2024

Table Of Contents

EXECUTIVE SUMMARY

Employee retention is a critical concern for organizations, with high turnover rates leading to increased costs, decreased productivity, and loss of institutional knowledge. Recognizing this challenge, our project aims to utilize data analytics and machine learning to predict employee churn and provide actionable insights to enhance retention strategies.

Our study analyzes a rich dataset of employee demographics, performance, work habits, and satisfaction metrics. By applying advanced predictive modeling techniques, including Decision Trees, Random Forest, and XGBoost, we aim to identify key factors driving employee churn and proactively manage at-risk employees. These models were rigorously tested and validated using metrics such as Confusion Matrix, Stratified Accuracy, and Expected Value. Our approach involves identifying the models' benchmark performance, followed by pre-processing the data, addressing class imbalances and leveraging hyperparameter tuning to optimize model performance.

The project's ultimate goal is to empower HR departments with data-driven tools to:

1. Identify employees most likely to resign.
2. Understand the underlying drivers of churn.
3. Design targeted, cost-effective retention strategies.

By integrating insights from machine learning models with business knowledge, this project can help position this organization to reduce turnover, enhance employee engagement, and save costs.

This report documents our process, key findings, and strategies for implementation, offering a detailed guide for utilizing predictive analytics in workforce management.

INTRODUCTION

Employee retention is a critical aspect of organizational success. High employee turnover not only incurs direct costs such as recruitment and training but also leads to a loss of organizational knowledge and productivity. Our business idea centers on understanding what drives employee turnover and proactively addressing them. By predicting at-risk employees and uncovering the key factors influencing churn, HR departments can focus their efforts on targeted retention strategies, such as improving job satisfaction, career growth opportunities, and workload management.

The dataset used for this analysis is from Kaggle and it includes over 100,000 anonymized employee records, capturing various details such as the employees demographics, job role, performance, productivity, satisfaction, and churn status. This comprehensive dataset forms the foundation for our analysis, and allows us to explore churn patterns and develop predictive models to eventually align with organizational goals.

DATA

Data Description

As shared in our introduction, our Kaggle[1] dataset contains 100,000 rows and 20 columns of anonymized employee data from a single organization. The dataset is a snapshot of the employee population that have resigned and not resigned between 10 years (2014 - 2024). The dataset captures a wide range of variables across several categories, including demographics (e.g., age, gender, education level), performance metrics (e.g., salary, performance scores), work habits (e.g., hours worked, sick days, overtime), and satisfaction metrics (e.g., employee satisfaction scores). Our target variable, "**Resigned**," is a binary indicator representing whether an employee has resigned or not. The dataset includes features listed below.

Attribute	Description
Employee_ID	Unique identifier for each employee.
Department	The department in which the employee works (e.g., Sales, HR, IT).
Gender	Gender of the employee (Male, Female, Other).
Age	Employee's age (between 22 and 60).
Job_Title	The role held by the employee (e.g., Manager, Analyst, Developer).
Hire_Date	The date the employee was hired.
Years_At_Company	The number of years the employee has been working for the company.
Education_Level	Highest educational qualification (High School, Bachelor, Master, PhD).
Performance_Score	Employee's performance rating (1 to 5 scale).

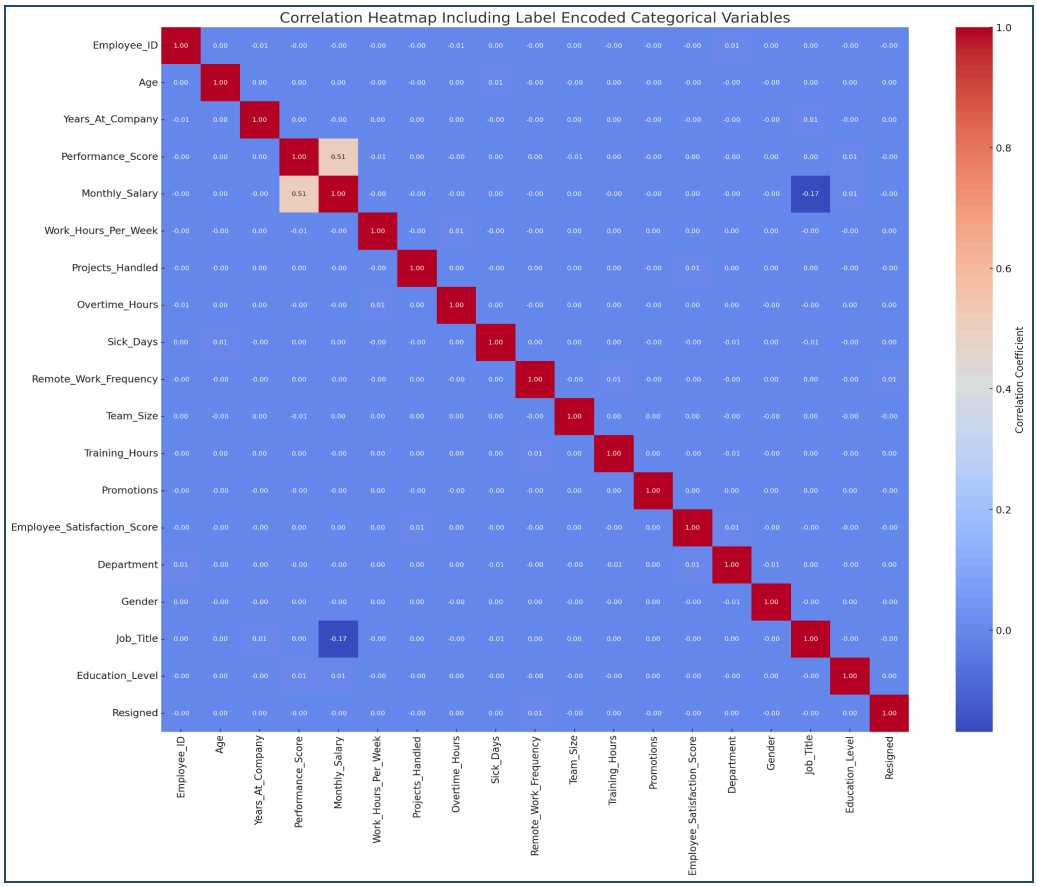
[1] Maxwell, Mex. (2021). *Employee performance and productivity data* [Dataset]. Kaggle. Retrieved from https://www.kaggle.com/datasets/mexwell/employee-performance-and-productivity-data/datahttps://www.kaggle.com/datasets/mexwell/employee-performance-and-productivity-data/data

Monthly_Salary	The employee's monthly salary in USD, correlated with job title and performance score.
Work_Hours_Per_Week	Number of hours worked per week.
Projects_Handled	Total number of projects handled by the employee.
Overtime_Hours	Total overtime hours worked in the last year.
Sick_Days	Number of sick days taken by the employee.
Remote_Work_Frequency	Percentage of time worked remotely (0%, 25%, 50%, 75%, 100%).
Team_Size	Number of people in the employee's team.
Training_Hours	Number of hours spent in training.
Promotions	Number of promotions received during their tenure.
Employee_Satisfaction_Score	Employee satisfaction rating (1.0 to 5.0 scale).
Resigned	Boolean value indicating if the employee has resigned.

This dataset was designed for HR analytics, to explore various factors including employee churn prediction, performance evaluation, and workforce optimization.

Visualization

The correlation heatmap below highlights there is almost no correlation between all the features in the dataset (categorical features transformed using Label Encoder), indicating that the decision to resign is not strongly related to other variables. Most of the feature pairs show low or near-zero correlations, indicating weak linear relationships. There is no strong correlation between other features and the "Resigned" status, which might indicate that the decision to resign is not linearly related to the other variables. As a result, Regression Models may not be the best choice for analysis since our target variable, “Resigned,” is binary and there are non-linear relationships between many features. Instead, we will be looking at other classifiers.



ANALYSIS

To begin our process of predicting employee churn, we utilized three machine learning models: **Decision Tree, Random Forest, and XGBoost**. These models were chosen due to their ability to capture non-linear relationships and interactions between features, as revealed during data exploration. We split the dataset into 78% for training and 22% for testing to ensure robust model evaluation.

Our evaluation strategy focused on metrics that balance accuracy with the financial implications of churn. Specifically, we assessed the models using (i) confusion matrices (to capture True Positives, False Negatives, etc.) and stratified accuracy (to address class imbalance). Additionally, we conducted a (ii) cost-benefit analysis using Expected Value, incorporating real-world costs of retention interventions and employee turnover.

By iteratively pre-processing the data (e.g., feature engineering, SMOTE resampling) and fine-tuning hyperparameters, we aimed to optimize each model's performance while aligning with the business objectives of reducing employee churn.

Benchmark

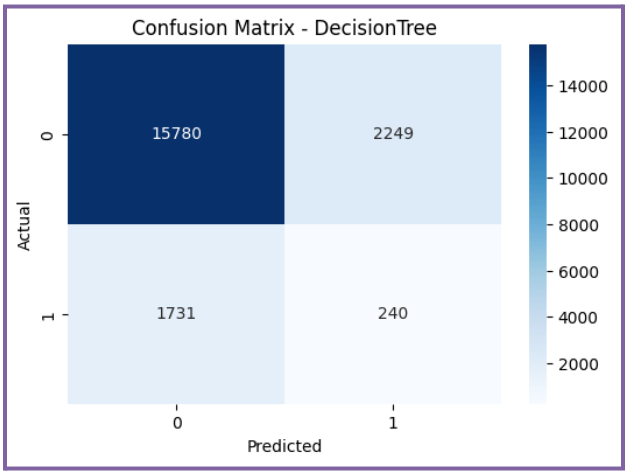
Proportions of the Class Variable

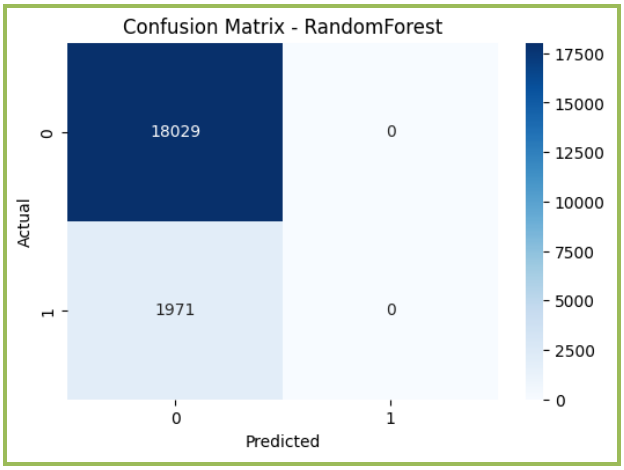
After initial analysis, a class imbalance was discovered with our target variable "Resigned." **Class 0 (Not Resigned)** accounts for **89.99%** of the instances, while **Class 1 (Resigned)** comprises only **10.01%**. This imbalance highlights the inadequacy of accuracy as a standalone metric. When running raw accuracy for the pre-processed data, the accuracies are near perfect. Raw accuracy shows a near perfect score of 80% for DecisionTree, 90% for RandomForest and XGBoost. However, Class imbalance skews the models toward maximizing overall accuracy at the cost of minority class detection. Therefore, metrics such as **Stratified Accuracy** and **Confusion Matrix** are better suited for evaluating the performance of models on this dataset.

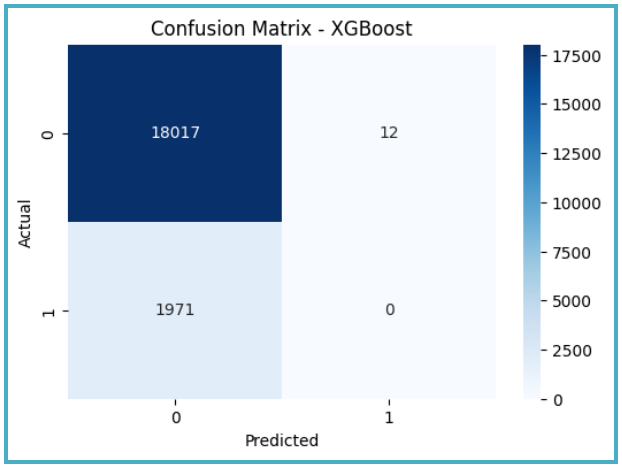
Benchmark Model Evaluation

(i) Stratified Accuracy & Confusion Matrix

Three models were evaluated: **DecisionTree**, **RandomForest**, and **XGBoost**. The following are the observations based on the confusion matrices and stratified accuracy for each model:







The **Decision Tree** model correctly identified 15,780 true negatives and 240 true positives for Class 1. However, the model struggled significantly with the minority class, with a Class 1 accuracy of only 12.18%, misclassifying 1,731 instances as false negatives and 2,249 instances as false positives. This indicates that while the model demonstrates a decent ability to classify the majority class, it faces challenges in accurately predicting the minority class, highlighting the impact of class imbalance on its performance.

- Class 0 Accuracy: 87.53%
- Class 1 Accuracy: 12.18%

The **Random Forest** model perfectly classified the majority class, achieving 100% accuracy for Class 0, with 18,029 true negatives and no false positives. However, it entirely failed to identify any instances of the minority class (Class 1), with a Class 1 accuracy of 0% and 1,971 false negatives. This performance suggests that the model overfits to the majority class and fails to capture the features of the minority class, making it unsuitable for addressing class imbalance in this scenario.

- Class 0 Accuracy: 100%
- Class 1 Accuracy: 0%

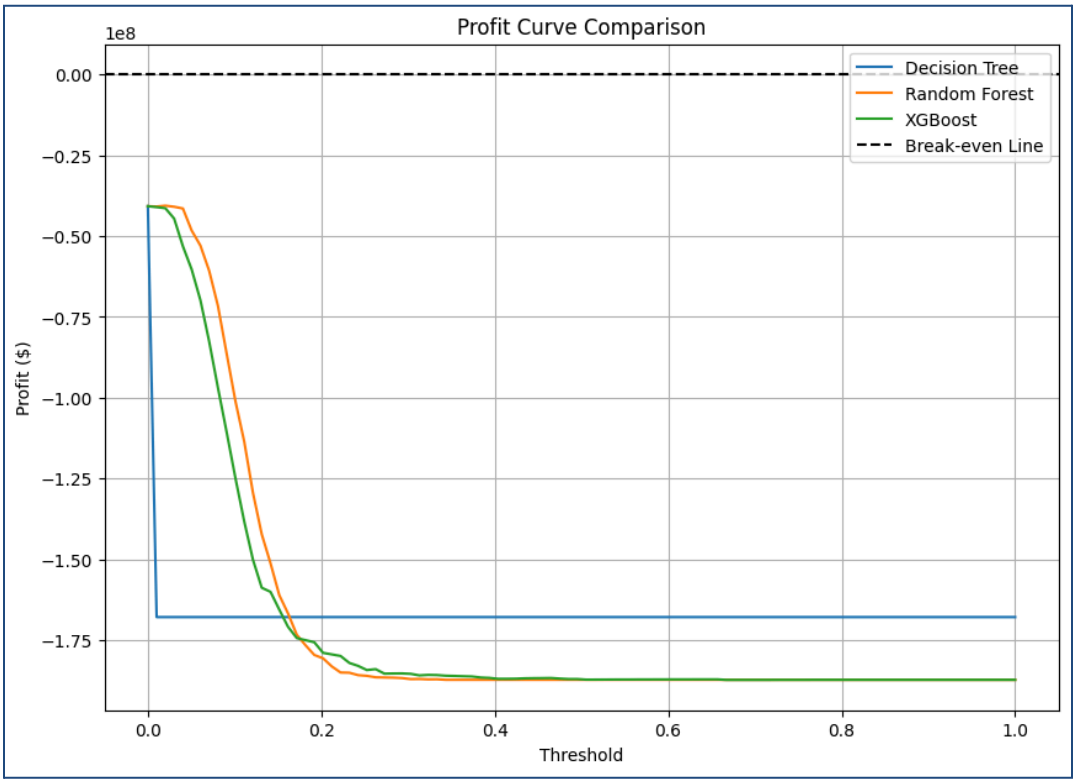
XGBoost demonstrated near-perfect classification for the majority class (Class 0), achieving an accuracy of 99.93% with 18,017 true negatives and only 12 false positives. However, like the Random Forest model, it completely failed to classify any instances of the minority class (Class 1), with a Class 1 accuracy of 0% and 1,971 false negatives. While XGBoost marginally improved on reducing false positives compared to Random Forest, it remains heavily biased toward the majority class, rendering it ineffective for identifying resignations.

- Class 0 Accuracy: 99.93%
- Class 1 Accuracy: 0%

(ii) Cost-Benefit Analysis (Expected Value)

The Cost Benefit Analysis quantifies the financial implications of employee turnover and retention. The financial implications of employee turnover and retention are significant in this case. Here, the **Cost of Turnover** is estimated at **$95,000** per employee, while the **Cost of Retention** is **$11,400** per employee. The potential **Benefit of Retaining At-Risk Employees** is calculated as **$83,600** per employee (difference between turnover and retention costs)[2]. For this, we calculated the Expected Value for each model and then visualized the performances using a Profit Curve.

The profit estimations for the models were calculated based on the number of True Positives (correctly identified resignations), False Positives (employees incorrectly predicted to resign), and False Negatives (resignations missed by the model). The **DecisionTree** model had an expected value of -$167,842,200, while **RandomForest** and **XGBoost** had even worse expected values of -$187,245,000 and -$187,180,400, respectively. These negative values indicate that the models fail to provide financial benefits due to their inability to capture the minority class.



[2] Workstream. (October 19, 2022). How to understand employee turnover costs. Retrieved December 6, 2024, from https://www.workstream.us/blog/how-to-understand-employee-turnover-costs

Our **Profit Curve Comparison** further supports this finding. None of the models produced a positive financial outcome, as their poor performance for Class 1 predictions resulted in significant costs associated with missed resignations. Despite these outcomes, the Decision Tree performs slightly better overall, suggesting that optimizing retention strategies could still yield better returns.

Data Pre-Processing

To address the imbalance and improve classification performance, several pre-processing steps were used:

1. **Dropped Ineffective Features**: We first dropped Features such as Employee_ID and Hire_Date from the dataset as they did not contribute meaningful predictive power to our models. This step simplified the model while maintaining performance.

2. **Transformed Categorical Variables**: Then, we used Label Encoding on categorical features like Department, Gender, Job_Title, and Education_Level to convert them into numeric representations for the models. This step simplified the model while maintaining performance.

3. **Resampling with SMOTE**: Very importantly, we used Synthetic Minority Oversampling Technique (SMOTE) to balance the dataset, almost equalizing the proportions of Class 0 and Class 1. After this step, the revised class distribution is: **False (50.1%)** and **True (49.9%)**.

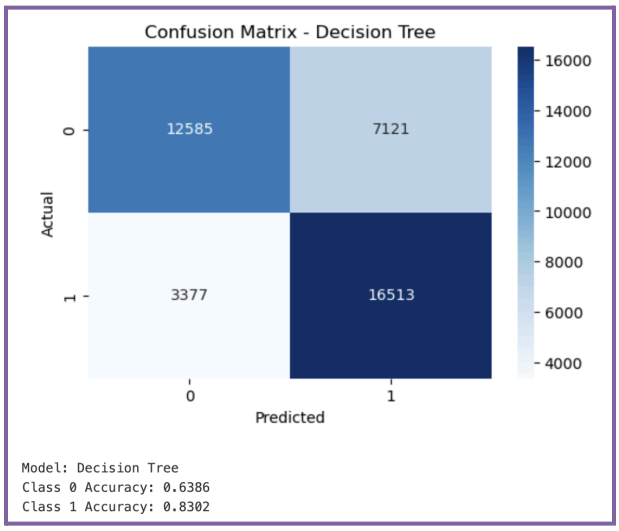
4. **Hyper-parameter Tuning**: In this case, grid search was used to find the best set of parameters and optimize the Decision Tree, Random Forest, and XGBoost models. We found the best hyper-parameters for our model are:

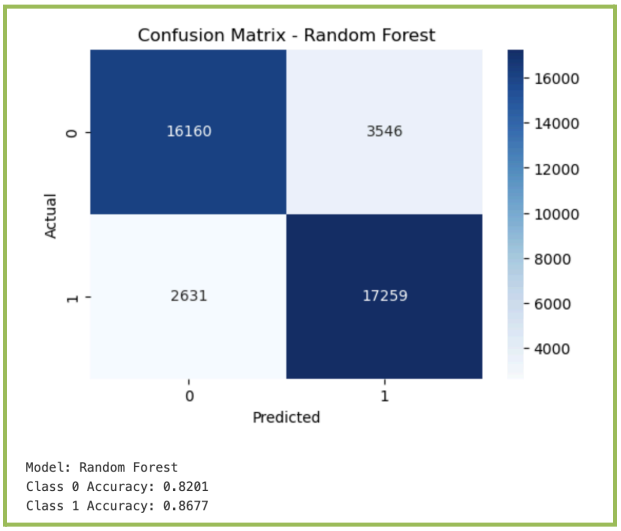
```
Best Hyperparameters:
Decision Tree: {'criterion': 'gini', 'max_depth': 20, 'min_samples_leaf': 10, 'min_samples_split': 2}
Random Forest: {'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 300}
XGBoost: {'learning_rate': 0.2, 'max_depth': 10, 'n_estimators': 300, 'subsample': 0.8}
```

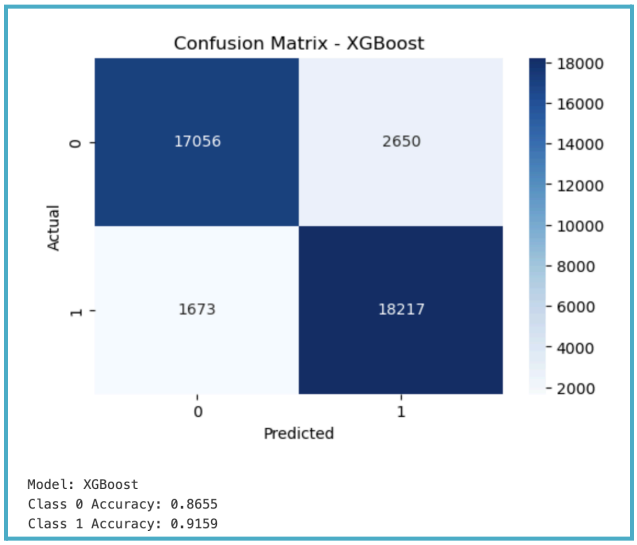
Optimized Models

Model Evaluation after Pre-Processing

(i) Stratified Accuracy and Confusion Matrix







The **Decision Tree** model showed improvement in identifying the minority class (Class 1), with 16,513 true positives correctly identified. However, it misclassified a significant portion of the majority class, with 7,121 false positives and a reduced accuracy for Class 0 at 63.86%. While its focus on improving minority class prediction is notable, the trade-off resulted in lower overall performance for the majority class.

- Class 0 Accuracy: 63.86%
- Class 1 Accuracy: 83.02%

Random Forest demonstrated a balanced performance, identifying 17,259 true positives for Class 1 and maintaining higher accuracy for the majority class. With 3,546 false positives, the Class 0 accuracy stood at 82.01%, and the Class 1 accuracy reached 86.77%. This balance makes Random Forest a reliable model for scenarios where equal attention to both classes is required.

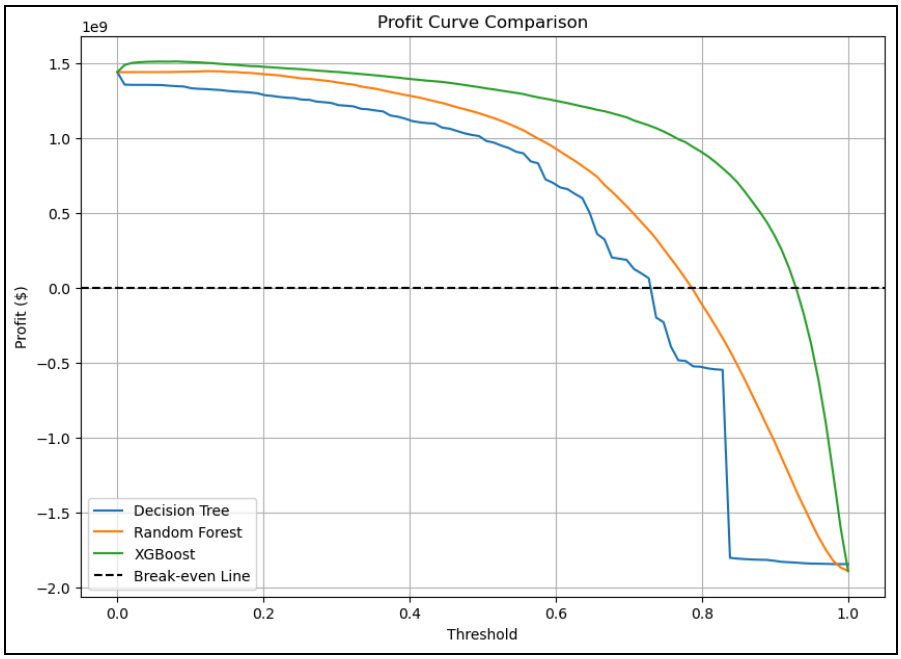
- Class 0 Accuracy: 82.01%
- Class 1 Accuracy: 86.77%

XGBoost outperformed the other models, achieving the best balance and overall accuracy. It identified 18,217 true positives for Class 1 with only 2,650 false positives for Class 0, resulting in the highest accuracy for both classes. XGBoost effectively addressed the class imbalance, making it the most reliable model for predicting resignations.

- Class 0 Accuracy: 86.55%
- Class 1 Accuracy: 91.59%

(ii) Cost-Benefit Analysis (Expected Value)

We ran our Cost-Benefit Analysis again using **Cost of Turnover** ($95,000), the **Cost of Retention** ($11,400) and the **Benefit of Retaining At-Risk Employees** ($83,600). For this, again we calculated the Expected Value for each model and then visualized the performances using a Profit Curve[3].



[3] APPENDIX - Important Code: Calculating Expected Value for each Model

The **Decision Tree** model's expected value is $978,492,400, driven by its ability to identify a reasonable number of at-risk employees but with some inefficiencies due to misclassifications.

Random Forest achieved an expected value of $1,152,483,000, outperforming the Decision Tree model by balancing accuracy for both classes and minimizing misclassification costs.

XGBoost demonstrated the highest expected value of $1,333,796,200, aligning with its superior performance metrics. This model's ability to effectively predict both classes ensures the highest potential financial benefit by accurately targeting at-risk employees for retention.

Key Steps for Model Improvement

In our data pre-processing, **SMOTE** and **Hyper-parameter Tuning** significantly enhanced the performance of all models. SMOTE helped in solving the issue of our dataset being highly imbalanced, which was the main reason for inaccurate accuracies in prior benchmarking. After balancing the dataset using SMOTE, the models began predicting both classes effectively. Hyper-parameter tuning was also a key step in improving performance. By systematically searching through predefined hyper-parameter spaces, we can identify the optimal combination of parameters that maximizes the performance of the model, providing us the best performing model for each classifier.

Choice of Metric and Model

To evaluate and compare model performance, we prioritized metrics that align with the business objective of minimizing churn costs while effectively identifying at-risk employees. Among the metrics analyzed — confusion matrix, stratified accuracy, and Cost-Benefit Analysis (Expected Value) — we selected **Expected Value** as the primary evaluation criterion.

Why Expected Value? Expected Value captures the financial implications of model predictions, considering the real-world costs of False Positives (unnecessary retention efforts) and False Negatives (missed resignations). Unlike traditional metrics like Accuracy scores, Expected Value directly reflects the business impact of employee churn predictions, making it the most actionable and relevant metric for this analysis[4].

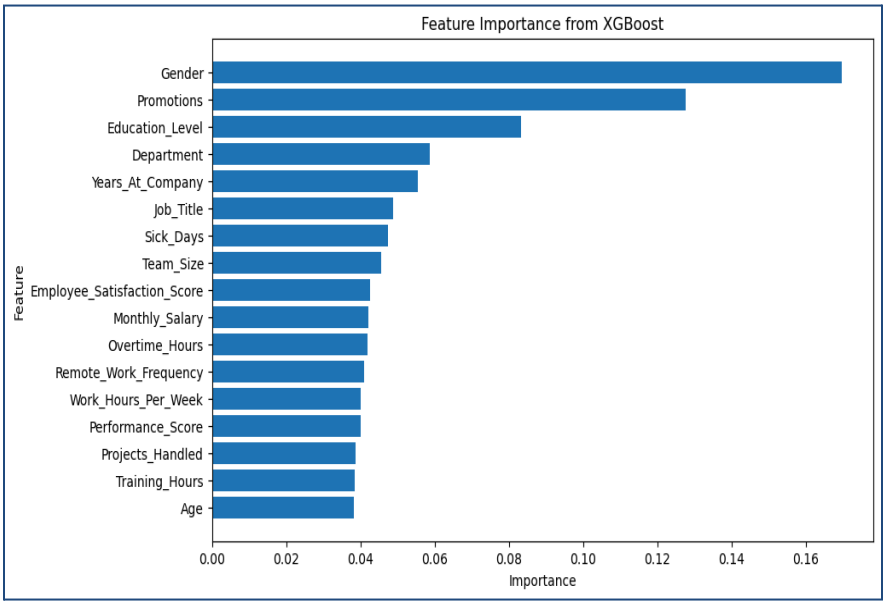
While Confusion Matrix and Stratified Accuracy provided insights into model performance and class balance handling, they do not account for the varying financial significance of prediction outcomes. Expected Value, therefore, was instrumental in identifying the model that delivers the highest net benefit to the organization.

[4] Investopedia. (July 25, 2024). Cost-benefit analysis. Retrieved December 6, 2024, from https://www.investopedia.com/terms/c/cost-benefitanalysis.asp

Based on the analysis, **XGBoost** emerged as the best-performing model for employee churn prediction. It consistently outperformed Decision Tree and Random Forest across all evaluation metrics, achieving the best stratified accuracy, and maximum Expected Value ($1,333,796,200). Its superior performance is due to its boosting framework, which repeatedly optimized predictions by minimizing errors and focusing on hard-to-classify samples. XGBoost's ability to handle class imbalances, specifically through weighted loss functions, ensures accurate prediction of the minority class (Class 1: Resigned) without compromising the accuracy for the majority class (Class 0: Not Resigned). Its performance in different metrics and high accuracy results make it the most reliable model for differentiating between employees likely to resign and those likely to stay. Therefore, XGBoost is the optimal model for employee churn prediction, aligning predictive performance with business priorities to maximize organizational benefits.

EMPLOYEE CHURN ANALYSIS

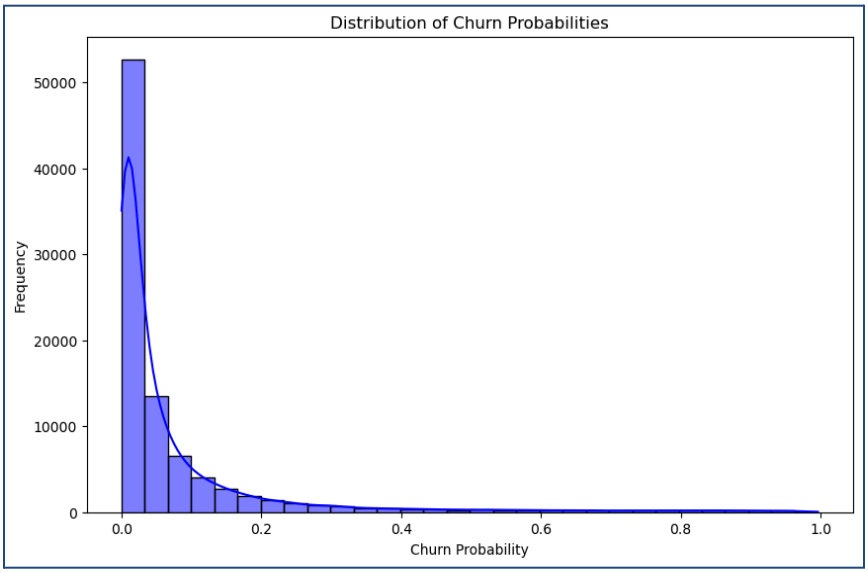
Now, to assess the likelihood of current employees leaving the company, we utilized the best-performing model, XGBoost, to calculate **churn probabilities**. Employees already marked as resigned were removed from the dataset to ensure a focused analysis on the active workforce[5]. The resulting churn probabilities provide actionable insights into the risk profile of the current employee base.



Using XGBoost's feature importance, we identified key factors influencing churn probabilities. **Gender**, **Promotions**, followed by **Department** and **Years At Company** are the most significant

[5] APPENDIX - Important Code: Filtering the data for Current Employees and Predicting Churn Probability

predictors of employee churn. Interestingly, variables such as employee satisfaction score, salary, overtime hours, and performance score were found to be weak or irrelevant predictors, contradicting traditional assumptions about their influence on churn.



The distribution of churn probabilities is highly skewed to the right, as illustrated in the attached histogram. The majority of employees have a churn probability between **0** and **0.2**, indicating a low risk of resignation. A smaller proportion of employees fall into the moderate to high-risk categories, with probabilities forming a long tail. This distribution highlights that while most employees are unlikely to churn, a targeted subset requires immediate attention. The focus of our subsequent analysis will be on understanding the drivers of churn for employees in this high-risk segment.

EDA Insights

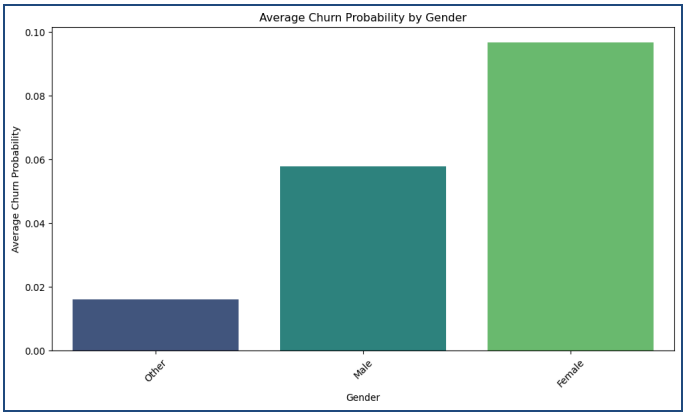
After our employee churn analysis, we've determined that our EDA focus would be on analyzing what causes the employees to be considered as high at-risk for churn. To understand the specifics that influence a higher churn rate, we looked at the **top 75% at-risk employees and their most common attribute values**[6].

Department	Job_Title	Education_Level	Promotions	Remote_Work_Frequency	Years_At_Company
IT	Engineer	Bachelor	0	75	3

From our analysis, we found that the **IT** Department has the most number of high at-risk employees, **Engineer** job, **0** Promotions, **75%** Remote Work Frequency, and **3 Years** at Company.

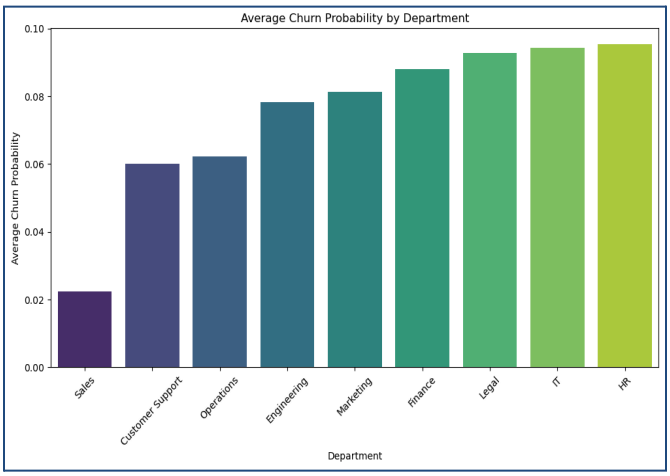
[6] APPENDIX - Important Code: Studying employees with High Churn Probability

Average Churn Probability by Gender



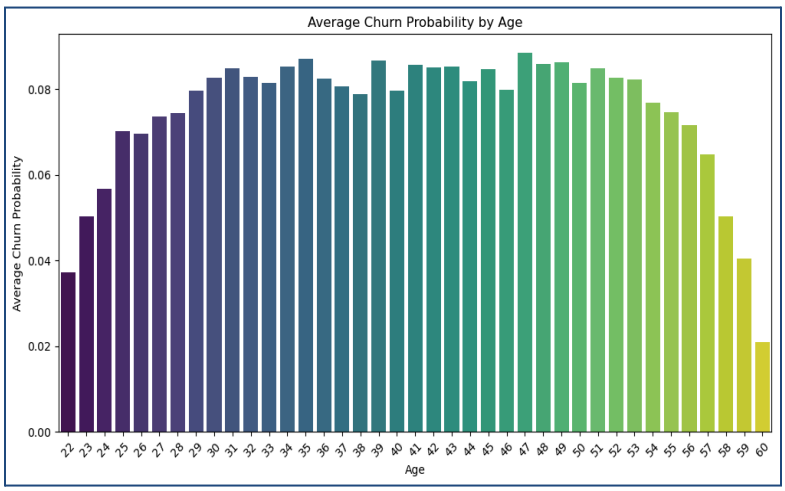
Next, for our first analysis, we looked at the **Average Churn Probability by Gender.** In this chart we learn that Female employees have the highest churn probability compared to Male, who fall in the middle, and Other exhibits the lowest churn probability.

Average Churn Probability by Department



These disparities also tied into our next analysis of **Average Churn Probability by Department.** In this analysis, we found that HR, IT and Legal have relatively high churn rates, while Sales had the lowest. Other departments including Customer Support, Operations, and Engineering fall in the mid-range. We learn that these departments have a majority of female employees, with higher levels of workload and less growth opportunities.

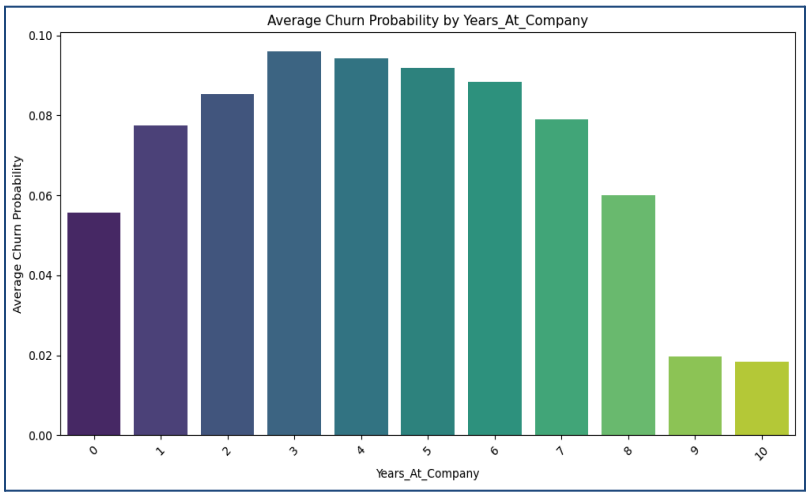
Average Churn Probability by Age



Next, we looked at the **Average Churn Probability By Age**. From this chart, we learn that churn rises steadily and peaks in the mid-40s of the employees. It then stabilizes and then declines after the age of 55. Younger employees (below 29) have lower churn probabilities. This is likely due to

younger employees wanting to build their stills and work experience at an organization. Similarly, older employees above 55 are also less likely to leave due to being closer to retirement. Also, there is a possibility that retention efforts have been more effective for these age groups.

Average Churn Probability by Years at Company



Next, we looked at the **Average Churn Probability by Years at the Company**. In this visual, we learn that churn is highest for employees with a tenure between three to six years at the company, peaking in their third year. However, the longer an employee stays at the organization, the less likely they will churn, and by year 7, there is a steady decline in the probability of churn. In summary, this analysis suggests to us that retention improves as employees gain tenure at the organization.

Strategies and Recommendations

Following the insights we gained from our EDA, we created several strategies and recommendations on how the organization can reduce churn and increase their retention. Our first goal is to target departments with the highest churn - **HR, IT** and **Legal** - this includes redistributing projects, reducing workload and automating repetitive tasks if possible. By addressing these issues, they can target what are the first pain points in these departments. Next, to help with their retention, they must **address what is causing dissatisfaction** within each of these departments individually to understand if the issues are isolated or consistent across. Lastly, the organization can help increase employee engagement by **encouraging collaboration between departments**, allowing for both skill-building and relationship development.

Next the organization can focus on their employees career development by **providing a clear path for growth** and **establishing a criteria for regular promotions**.[7] For their high-risk

[7] Walden University. (n.d.). *Six strategies to reduce employee turnover.* Retrieved December 6, 2024, from https://www.waldenu.edu/online-masters-programs/ms-in-human-resource-management/resource/six-strategies-to-reduce-employee-turnover

employees, **leadership or rotational opportunities can be provided**, and high performing employees can receive **regular performance reviews** to ensure they feel more valued and recognized**.**

Finally, to help employees feel more included at work, the company can **send out gender-specific engagement surveys** to help uncover and understand their concerns. They can also **create ERGs (Employee Resource Groups)** to foster community and support, and to encourage a work-life balance, **reviewing and optimizing their remote work policies** to align with employees needs can help gain loyalty and encouragement.

In summary from our insights, to reduce employee churn, the organization should focus on **department-specific interventions**, **provide career development opportunities**, and **creating a workplace where everyone feels included and supported**.

CONCLUSION

This report takes a detailed look at employee churn, using data-driven methods to predict which employees are at risk of resigning and uncover actionable strategies to improve retention. By combining advanced machine learning models with cost-benefit analysis, we've built a practical framework to tackle the challenges of employee turnover.

XGBoost stood out as the best-performing model for predicting churn, offering the highest accuracy and financial benefits by identifying complex patterns in the data. The analysis pointed to key factors influencing churn, such as gender, promotion history, department, and tenure, while showing that factors like salary and satisfaction scores might be less important than expected.

These findings help organizations focus their retention efforts on employees at the highest risk of leaving, allowing for targeted strategies that are both cost-effective and impactful. With data-driven approaches, businesses can not only reduce turnover but also improve employee satisfaction, support career growth, and build a more stable workforce.

Overall, our analysis underscores the value of combining predictive analytics with HR practices, offering a clear plan for using machine learning to strengthen workforce retention and drive long-term success.

APPENDIX - Important Code

Data Pre-Processing

```
[ ] ineffective_features = ["Employee_ID","Hire_Date"]
    df = df.drop(columns = ineffective_features, axis = 1 )

    df.head
```

Show hidden output

```
[ ] label_encoders = {}
    for column in ["Department","Gender", "Job_Title", "Education_Level"]:
        le = LabelEncoder()
        df[column] = le.fit_transform(df[column])
        label_encoders[column] = le
```

Resampling using SMOTE, and preparing data for Modeling

```
[ ] y = df["Resigned"]
    x = df.drop("Resigned", axis = 1)
```

```
[ ] smote = SMOTE(random_state=42)
    x_resampled, y_resampled = smote.fit_resample(x, y)
```

```
[ ] X_train, X_test, y_train, y_test = train_test_split(x_resampled, y_resampled, train_size=0.78, random_state=42)
```

Hyper-parameter Tuning using GridSearchCV

```
# Hyperparameter Tuning for Decision Tree
dt_param_grid = {
    'criterion': ['gini', 'entropy'],
    'max_depth': [5, 10, 20, None],
    'min_samples_split': [2, 10, 20],
    'min_samples_leaf': [1, 5, 10]
}

dt_grid = GridSearchCV(DecisionTreeClassifier(random_state=42), dt_param_grid, cv=3, scoring='accuracy', n_jobs=-1)
dt_grid.fit(X_train, y_train)
dt_best_model = dt_grid.best_estimator_

# Hyperparameter Tuning for Random Forest
rf_param_grid = {
    'n_estimators': [100, 200, 300],
    'max_depth': [10, 20, None],
    'min_samples_split': [2, 10],
    'min_samples_leaf': [1, 5]
}

rf_grid = GridSearchCV(RandomForestClassifier(random_state=42), rf_param_grid, cv=3, scoring='accuracy', n_jobs=-1)
rf_grid.fit(X_train, y_train)
rf_best_model = rf_grid.best_estimator_

# Hyperparameter Tuning for XGBoost
xgb_param_grid = {
    'n_estimators': [100, 200, 300],
    'max_depth': [3, 5, 10],
    'learning_rate': [0.01, 0.1, 0.2],
    'subsample': [0.8, 1.0]
}

xgb_grid = GridSearchCV(XGBClassifier(use_label_encoder=False, eval_metric='logloss', random_state=42),
                        xgb_param_grid, cv=3, scoring='accuracy', n_jobs=-1)
xgb_grid.fit(X_train, y_train)
xgb_best_model = xgb_grid.best_estimator_
```

Calculating Confusion Matrix and Stratified Accuracy

```
# Function to calculate stratified accuracy scores
def calculate_stratified_accuracy(y_true, y_pred):
    conf_matrix = confusion_matrix(y_true, y_pred)
    class_accuracies = conf_matrix.diagonal() / conf_matrix.sum(axis=1)
    return class_accuracies

# Evaluate models and output confusion matrices and stratified accuracy scores
for model_name, model in models.items():
    # Predictions
    y_pred = model.predict(X_test)

    # Confusion Matrix
    conf_matrix = confusion_matrix(y_test, y_pred)
    plt.figure(figsize=(6, 4))
    sns.heatmap(conf_matrix, annot=True, fmt="d", cmap="Blues")
    plt.title(f"Confusion Matrix - {model_name}")
    plt.xlabel("Predicted")
    plt.ylabel("Actual")
    plt.show()

    # Stratified Accuracy Scores
    stratified_accuracy = calculate_stratified_accuracy(y_test, y_pred)
    print(f"Model: {model_name}")
    print(f"Class 0 Accuracy: {stratified_accuracy[0]:.4f}")
    print(f"Class 1 Accuracy: {stratified_accuracy[1]:.4f}")
    print("-" * 50)
```

Calculating Expected Value for each Model

```
# Define cost-benefit values (example values, adjust as needed)
cost_tp = 83600  # Saved cost of retaining an at-risk employee
cost_fp = -11400  # Cost of retaining a non-at-risk employee unnecessarily
cost_fn = -95000  # Cost of losing an at-risk employee
cost_tn = 0  # No cost for correct non-churn prediction

# Function to calculate expected value
def calculate_expected_value(y_true, y_pred, cost_tp, cost_fp, cost_fn, cost_tn):
    tn, fp, fn, tp = confusion_matrix(y_true, y_pred).ravel()
    expected_value = (
        tp * cost_tp +
        fp * cost_fp +
        fn * cost_fn +
        tn * cost_tn
    )
    return expected_value

# Predictions for each model
dt_predictions = dt_best_model.predict(X_test)
rf_predictions = rf_best_model.predict(X_test)
xgb_predictions = xgb_best_model.predict(X_test)

# Calculate expected value for each model
dt_ev = calculate_expected_value(y_test, dt_predictions, cost_tp, cost_fp, cost_fn, cost_tn)
rf_ev = calculate_expected_value(y_test, rf_predictions, cost_tp, cost_fp, cost_fn, cost_tn)
xgb_ev = calculate_expected_value(y_test, xgb_predictions, cost_tp, cost_fp, cost_fn, cost_tn)
```

Filtering the data for Current Employees and Predicting Churn Probability

```
# Assuming "current employees" means filtering employees who have not resigned in the dataset
current_employees = df[df['Resigned'] == 0].drop(columns=['Resigned'])

# Predict the likelihood of resignation (probabilities) for current employees using the best model
current_employees['Churn_Probability'] = xgb_best_model.predict_proba(current_employees)[:, 1]
```

Studying employees with High Churn Probability

```
# Filter employees with high churn probabilities (greater than or equal to 75%)
high_churn_employees = churn_data[churn_data['Churn_Probability'] >= 0.75]

# Summarize key statistics for high churn employees
high_churn_summary = high_churn_employees.describe()

# Identify most common attributes among high churn employees (e.g., Department, Job Title, etc.)
high_churn_common_attributes = high_churn_employees[['Department', 'Job_Title', 'Education_Level',
                                                     'Promotions', 'Remote_Work_Frequency', 'Years_At_Company', 'Gender']].mode()
```

```
              index          Age  Years_At_Company  Performance_Score  \
count   1193.000000  1193.000000       1193.000000        1193.000000
mean   49079.805532    40.890193          4.034367           2.995809
std    28671.894237    10.158336          2.206912           1.425728
min       24.000000    22.000000          0.000000           1.000000
25%    24006.000000    32.000000          2.000000           2.000000
50%    49209.000000    41.000000          4.000000           3.000000
75%    72803.000000    50.000000          6.000000           4.000000
max    99882.000000    60.000000          8.000000           5.000000

       Monthly_Salary  Work_Hours_Per_Week  Projects_Handled  Overtime_Hours  \
count     1193.000000          1193.000000       1193.000000     1193.000000
mean      6446.102263            44.354568         23.646270       13.998324
std       1357.524511             7.677159         13.082557        7.406010
min       3850.000000            30.000000          0.000000        0.000000
25%       5250.000000            38.000000         12.000000        8.000000
50%       6500.000000            45.000000         23.000000       14.000000
75%       7500.000000            51.000000         35.000000       20.000000
max       9000.000000            60.000000         47.000000       28.000000

        Sick_Days  Remote_Work_Frequency    Team_Size  Training_Hours  \
count  1193.000000            1193.000000  1193.000000     1193.000000
mean      6.583403              50.419111     9.549036       49.422464
std       3.388932              35.174427     4.480324       27.624567
min       0.000000               0.000000     1.000000        0.000000
25%       4.000000              25.000000     6.000000       26.000000
50%       7.000000              50.000000    10.000000       50.000000
75%       9.000000              75.000000    13.000000       74.000000
max      14.000000             100.000000    19.000000       99.000000

       Promotions  Employee_Satisfaction_Score  Churn_Probability
count  1193.000000                  1193.000000        1193.000000
mean      0.585918                     2.978935           0.859524
std       0.609931                     0.868500           0.065985
min       0.000000                     1.090000           0.750399
25%       0.000000                     2.300000           0.803408
50%       1.000000                     2.960000           0.855256
75%       1.000000                     3.640000           0.914214
max       2.000000                     4.850000           0.996288
```