A Classifier for Screening Out Leaving Employees

1155143605 Li SHENGWEI, 1155116154 CHEONG Euikyun, 1155158054 LUI Chak Sum,

1155158643 CHAU Ka Yan, 1155142918 HO Tsz Hin, 1155158385 WONG Tuen Hung

1. Introduction

1.1 Background

Based on the Guangdong-Hong Kong-Macau Greater Bay Pay and Benefit Survey (2023), resignation rates in Hong Kong companies range from 8.9% to 29.4%. Talent attrition is a common issue in Hong Kong organizations, leading employers to seek strategies to reduce resignations and save resources. High turnover rates result in financial losses, wasted time, and increased costs for companies that invest heavily in their employees.

1.2 Motivation

To address the turnover rate, effective talent retention approaches (Ott et al., 2018) have advocated four methods since 2018. These methods include fostering a strong organizational culture and values, providing relevant training opportunities, improving the working environment, and establishing clear career advancement paths. While many companies have implemented these methods, the results have been unsatisfactory, suggesting a missing element in the talent retention process. This crucial step involves identifying potential individuals who may be considering leaving the company and implementing additional measures to retain them, such as salary adjustments. Proactively identifying employees inclined to leave is essential for addressing their concerns and preventing their departure.

1.3 Dataset Overview

Variables	Content	Variables	Content	Variables	Content
Education	Bachelors, Master, Phd	Payment Tier	1,2,3	Ever Benched	1,0
Joining Year	2012 to 2018	Age	22 to 41	Experience	0 to 7 years
City	Pune, Bangalore...	Gender	Male & Female	Leave or Not	1,0

Table 1: Dataset Overview

The employee dataset obtained from Kaggle (Elmetwally, 2023) consists of 8 independent variables and 1 dependent variable. These variables include education, year of joining, working city, payment tier, age, gender, ever benched, and experience in the current domain. The payment tier variable has three possible outputs: 1 represents a high salary range, while 3 represents a low salary range. Additionally, the ever-benched variable has two outputs: 1 indicates that the employee has not been assigned any tasks for a month.

2. Data Visualization

In this section, we will analyze two aspects of our dataset. First, we will examine the proportion of employees leaving and staying to understand the distribution of employee attrition. Secondly, we will explore the relationship between features and employee attrition to identify any patterns.

2.1 Proportion of Leaving and Staying Employees

In Figure 1, the data reveals that we have 34.4% loan employees leaving and 65.6% staying, indicating that leaving employees are the minority in the dataset. However, the imbalanced nature of the dataset can potentially result in less satisfactory outcomes when classifying leaving employees. Therefore, in Section 5, we will evaluate our models by considering precision and recall.

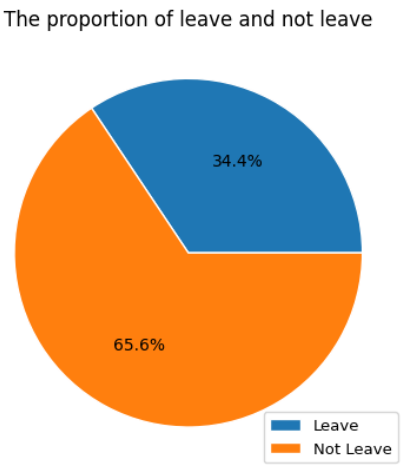

Figure 1: The proportion of leaving and staying employees

2.2 Relationship between Features and Employee Attrition

The bar plots in Figure 2 provide valuable insights on the relationship between features and employee attrition.

Three types of employees are more likely to leave the company. Firstly, new employees without strong bonds to the company may seek the opportunity to leave. Secondly, employees in payment tier 2, who typically have more years of service, may seek better prospects if growth opportunities are lacking. Lastly, employees in Pune may be influenced by serval factors such as industry concentration, infrastructure, and community networks in Pune. Therefore, joining year, payment tier, and city are critical factors in determining employee attrition. In Section 4, we will further assess their significance using feature importance to strengthen our findings.

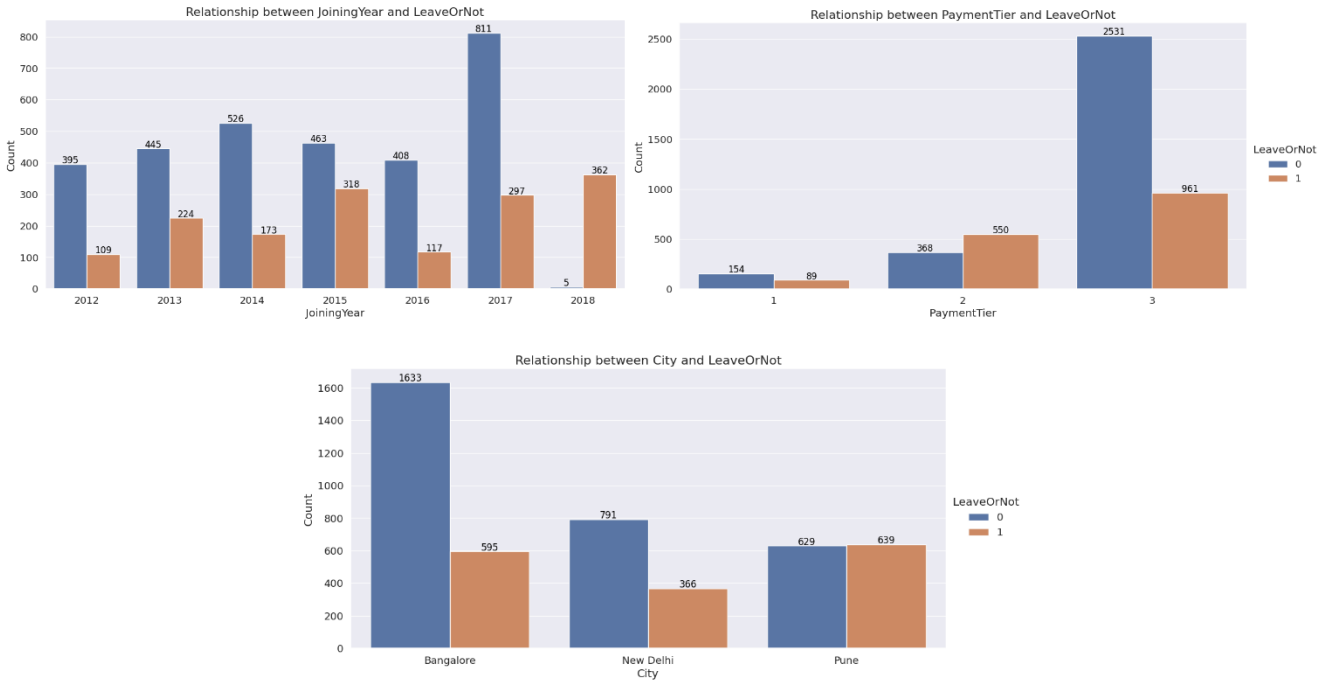

Figure 2: Relationship between joining year, payment tier, city, and employee attrition

3. Data Preparation

3.1 Data Cleansing

Our dataset underwent rigorous quality assurance checks to ensure its suitability for analysis. These checks included identifying missing values, incorrect data formats, duplicate entries, and potential outliers. No missing values were found, and all entries aligned with the expected data types and formats. We also confirmed the absence of duplication and outliers by crosschecking with descriptive statistics. In conclusion, our dataset is of high quality, supporting the robustness of our analysis and the validity of the results.

3.2 Data Preprocessing

To facilitate subsequent modeling, we performed preprocessing steps such as One-Hot Encoding and Data Splitting. Since our dataset contained categorical variables, we converted them into numerical format using One-Hot encoding. This technique transforms each categorical variable with n categories into n binary features, where each represents a single category. We retained all binary features, including the first column, as tree-based models benefit from having more information and they can handle correlated features effectively. Appendix F shows the result with n-1 binary features.

To assess the performance of our models, the dataset is divided into 80% of training data for training our models, and the remaining 20% of testing data was reserved for testing the model's performance. This commonly used technique helps to avoid overfitting and provides an unbiased evaluation of the model's ability to generalize to the unseen or new data.

4. Model Building

Tree-based classifiers such as decision tree, light gradient boosting machine (LightGBM), bagging, and random forest, were trained. Grid search with cross-validation was applied to tune the hyperparameters using the training data. To evaluate the performance of our models, we calculated the average accuracy, precision, recall, F1-score, and false positive rate (FPR) returned from tuned models with 10 random seeds using the testing data.

4.1 Decision Tree

A decision tree can be employed by learning simple decision rules from data features to mitigate the turnover rate.

Instead of using grid search, we evaluated the decision tree at various depths. Our findings indicate that max_depth at 9 yields the highest score for both the testing and training datasets. After the hyperparameter tuning, the model's performance resulted in an accuracy of 86.62%, precision of 88.91%, recall of 69.91%, F1-score of 78.27%, and FPR of 4.59%. Although the result seems not bad, it is expected that LightGBM, a more complex model, would yield better results. For feature importance, highlighting JoiningYear_2018, Payment_Tier2, and City_Pune as the most important factors in the decision tree model, which confirmed the findings from Section 2.

4.2 Light Gradient Boosting Machine

LightGBM, an advanced model building on a basic one, iteratively incorporates new trees to rectify previous mistakes. It employs a histogram-based methodology to efficiently categorize data into bins. This approach is particularly effective for large datasets, as it chooses the optimal splits for tree growth, enhancing performance.

By grid search, the optimal settings were determined as: N_estimator at 2000, Learning_rate at 0.001, Max_depth set to -1, Num_leaves at 31, and Min_child_samples at 5. Post hyperparameter tuning, the model exhibited an accuracy of 87.86%, precision of 93.33%, recall of 69.78%, F1-score of 79.86%, and a FPR of 2.62%. While accuracy, precision, and F1-score saw improvements, recall experienced a decline. Regarding feature importance, the 'female' and 'age' attributes significantly influenced the model's performance.

4.3 Bagging

Bagging is a technique that involves training multiple models on different subsets of the training data and combining their predictions. It is particularly useful when dealing with imbalanced data, as it utilizes bootstrap samples. Compared to LightGBM, bagging was chosen for its ability to handle imbalanced data effectively.

After hyperparameter tuning, the model achieved impressive results: an accuracy of 87.89%, precision of 91.71%, recall of 71.34%, F1-score of 80.25%, and a low FPR of 3.39%. These metrics indicate that the model outperforms both the decision tree and LightGBM models in terms of accuracy, recall, and F1-score. However, it is anticipated that Random Forest, an extension of bagging, will yield even better results.

4.4 Random Forest

Random Forest is a bagging algorithm; the difference with regular bagging is that it only draws random subsets of features for training multiple decision trees, and this makes the multiple decision trees more independent of each other, so they often have better predictive performance.

After the hyperparameter tuning, the model's performance resulted in an accuracy of 87.60%, precision of 93.60%, recall of 68.75%, F1-score of 79.27%, and FPR of 2.48%. These metrics show that the random forest model outperforms the other three regarding precision and false positive rate. In terms of feature importance, our analysis

confirmed the findings from Section 2 again, highlighting JoiningYear_2018, City_Pune, and Payment_Tier2 as the most influential factors in the random forest model. This consistency strengthens the significance of these features in predicting the outcome.

5. Model Evaluation

The average performance metrics from Section 4 are summarized in the table below:

(Average)	Accuracy	Precision	Recall	F1-score	FPR
Decision Tree	0.866	0.889	0.699	0.783	0.046
LightGBM	0.878	0.933	0.698	0.799	0.026
Bagging	**0.879**	0.917	**0.713**	**0.803**	0.034
Random Forest	0.876	**0.936**	0.688	0.793	**0.025**

Table 2: 4 model classifiers' performance

Based on the information provided, the bagging model demonstrates superior performance in terms of average accuracy, recall, and F1-score compared to the other three models. On the other hand, the random forest model excels in average precision and FPR. It is worth noting that the difference in average accuracy between the bagging and random forest models is minimal, with only a 0.003 margin. To make a final decision on the best model, it is suggested to analyze the ROC curve and PR curve for both the bagging and random forest models.

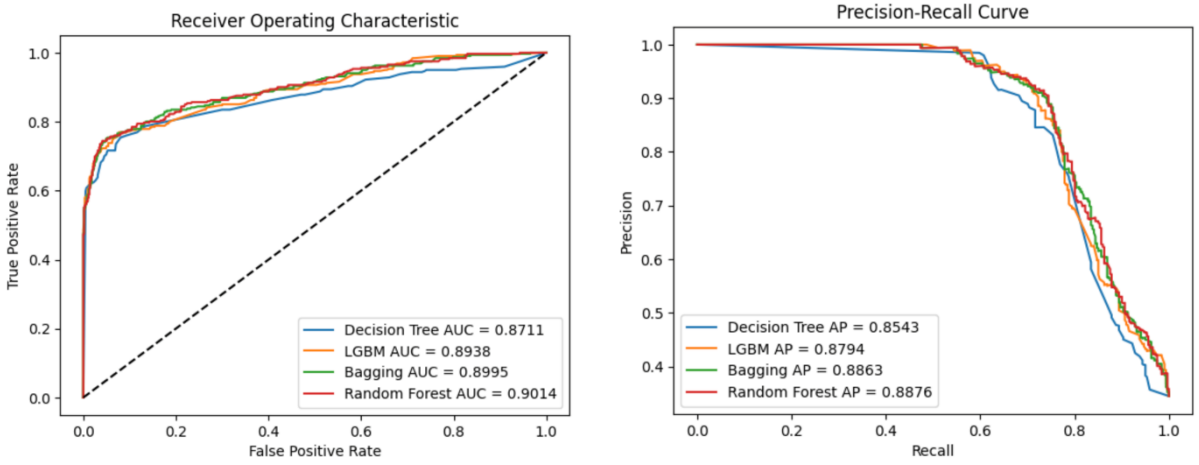

Figure 3: ROC curve and PR curve for the 4 classifiers

Based on the ROC and PR graphs, the random forest model exhibits the highest area under the curve (AUC) values of 0.9014 and 0.8876 in the ROC and PR curves, respectively. The higher AUC value of the random forest model suggests that it has a better ability to discriminate between positive and negative instances, making it more reliable for identifying employees who are likely to leave the company. Therefore, based on these results, selecting the random forest model as the desired model for predicting potential employee turnover seems appropriate.

6. Conclusion

In conclusion, the Random Forest model emerged as the most effective in predicting employee attrition, achieving a precision of 93.6%. Although the bagging model displayed impressive performance, like that of the random forest, the random forest was selected due to its superior AUC value. This higher AUC value is particularly beneficial in identifying employees who are more likely to leave the company.

The analysis revealed that the key features influencing this model include the year of joining (specifically 2018), employees based in Pune, and those classified under the second payment tier. These findings align with those derived from the decision tree and random forest model, underlining the significance of these attributes. These attributes suggest that employees with extended tenure are more inclined to remain with the organization, while those more recently employed are prone to departure. A noticeable turnover rate is evident among employees in the second payment tier, potentially due to more experienced employees seeking superior opportunities. Particularly in Pune, aspects such as industry concentration, infrastructure, and community networks are crucial in determining job stability, thereby highlighting the impact of regional factors on employee retention. Therefore, it is recommended to pay closer attention to these three groups of employees.

One challenge encountered in the project was the restricted use of variables, which could potentially impact the precision of the machine learning outcomes. To mitigate this, it is recommended to refine the variable set. This could involve further segmentation of payment tiers for a nuanced understanding of payment dynamics, as well as incorporating additional environmental factors such as average housing prices and transportation facilities. Implementing these adjustments is anticipated to enhance the thoroughness and applicability of the analysis.

References

Chan, T. (2023, October 30). *Pay and benefits trends across Guangdong-Hong Kong-macao greater bay area*. https://www.humanresourcesonline.net/pay-and-benefits-trends-across-guangdong-hong-kong-macao-greater-bay-area

Elmetwally, T. (2023, September 6). *Employee dataset*. Kaggle. https://www.kaggle.com/datasets/tawfikelmetwally/employee-dataset/data

Huichen Z.(2023). *STAT 4001 statistics projects* https://goo.su/dI06cX

Ott, D. L., Tolentino, J. L., & Michailova, S. (2018). *Effective talent retention approaches. Human resource management international digest*, 26(7), 16-19.

Appendix

A. Further Results on Data Visualization

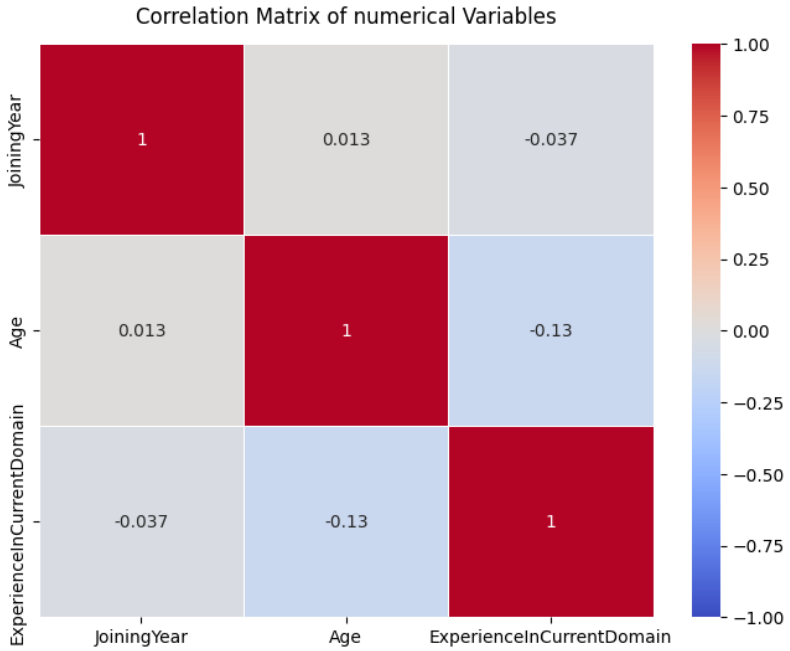

Figure A1: Correlation heatmap of numerical variables

B. Decision Tree

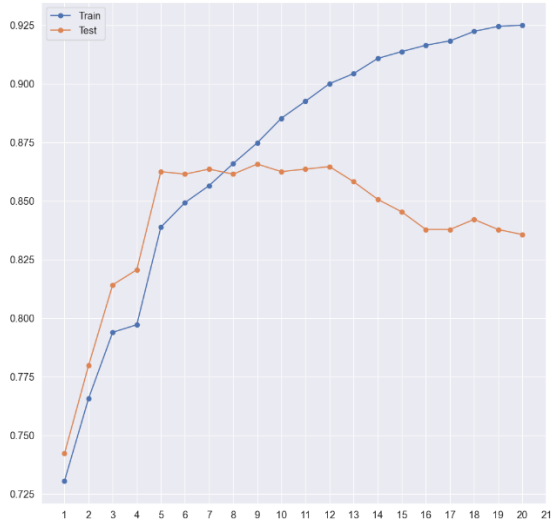

Figure B1: Hyperparameter tuning for Decision Tree

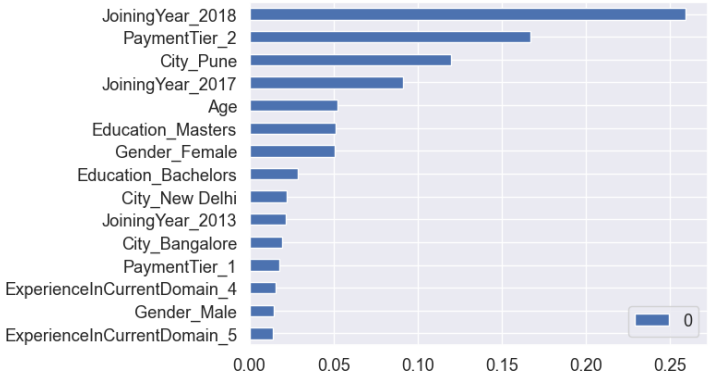

Figure B2: Feature importance for Decision Tree

C. Light Gradient Boosting Machine

N_estimators	100	500	1000	**2000**
Learning_rate	**0.001**	0.01	0.1	0.3
Max_depth	**-1**	5	10	15
Num_leaves	**31**	127	255	511
Min_child_samples	**5**	20	50	100

Table C1: Hyperparameter set for LightGBM hyperparameter tuning

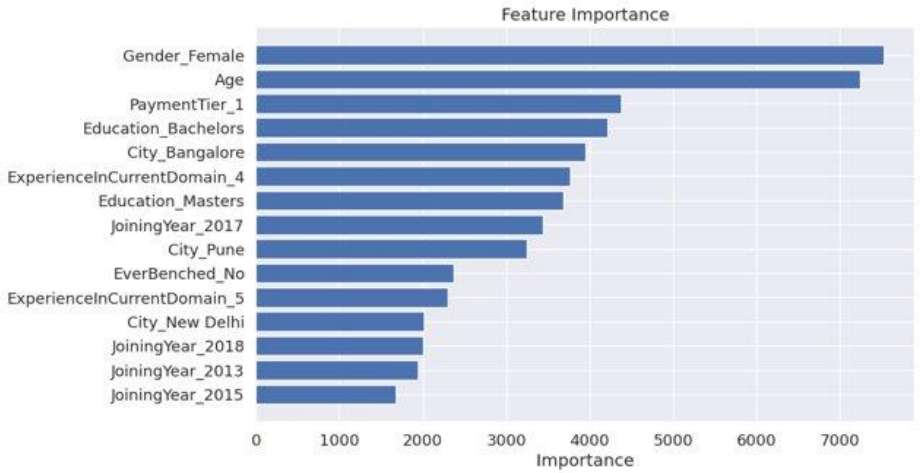

Figure C1: Feature importance for LightGBM

D. Bagging

estimator	DecisionTreeClassifier(max_depth=5)	**DecisionTreeClassifier(max_depth=7)**	DecisionTreeClassifier(max_depth=9)
n_estimators	**1000**	1500	2000
max_samples	**0.5**	0.7	0.9
max_features	0.5	0.7	**0.9**

Table D1: Hyperparameter set for Bagging hyperparameter tuning

E. Random Forest

max_depth	5	**10**	15
min_samples_leaf	1	**3**	5
min_samples_split	**3**	4	5
n_estimators	1000	1500	**2000**

Table E1: Hyperparameter set for Random Forest hyperparameter tuning

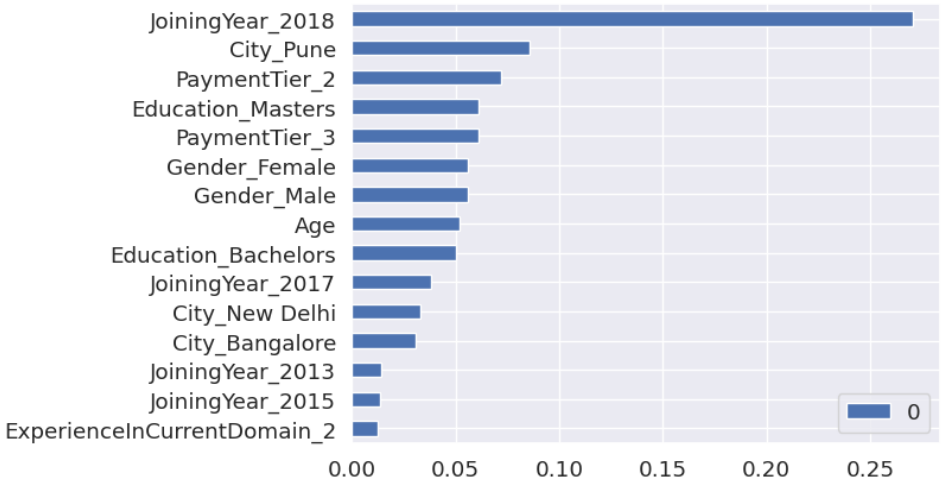

Figure E1: Feature importance for Random Forest

F. Model Evaluation Results with Columns Dropped

We also attempted to apply one-hot encoding with dropping the first column of all features. The Python code and the results can be accessed at this link: https://colab.research.google.com/drive/10Wdg2hLGiPPqeFRn6dJnKcPB-o6SXGjm?usp=sharing. By evaluating the models with the first columns dropped, we observed that the performance of both bagging and random forest models deteriorated compared to the results in Section 5. Additionally, the

average precision in the PR curve decreased for all models. This explained why we did not drop the first column of all features in this project.

(Average)	Accuracy	Precision	Recall	F1-score	FPR
Decision Tree	0.868	0.878	**0.717**	0.789	0.052
LightGBM	0.863	**0.949**	0.636	0.761	**0.018**
Bagging	**0.874**	0.913	0.702	**0.794**	0.035
Random Forest	0.871	0.927	0.679	0.784	0.028

Table F1: 4 model classifiers' performance with the first columns dropping

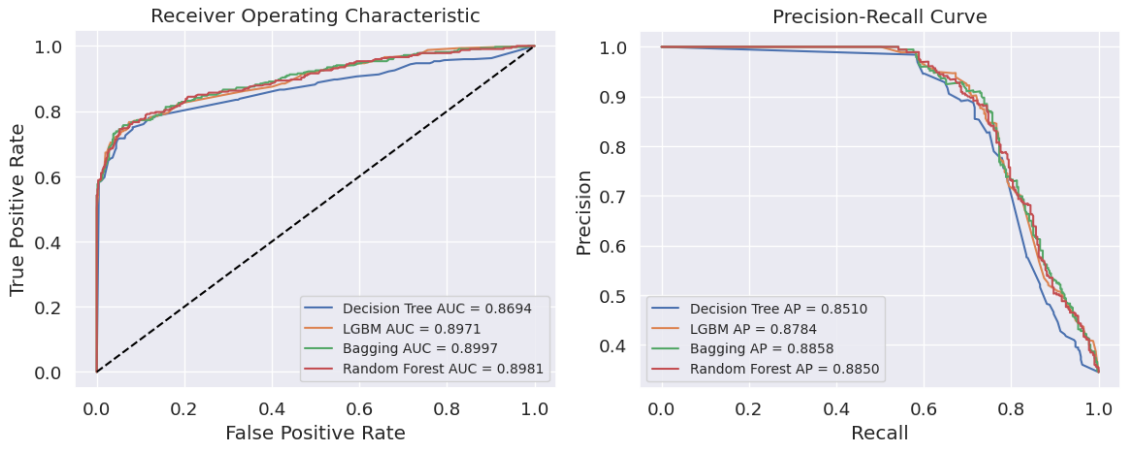

Figure F1: ROC curve and PR curve for the 4 classifiers with the first columns dropping