KULLIYYAH OF INFORMATION & COMMUNICATION TECHNOLOGY

CSCI 4340 MACHINE LEARNING

SEMESTER 1, 2022/2023

SECTION 1

Project Title

Predict Employee Turnover

Group G

AHMED JOBAER	**1918243**

LECTURER

DR. AMELIA RITAHANI BINTI ISMAIL

Table of content

1.0 Abstract:

In this study, we aimed to develop a model for employee turnover using the DT and XgBoost algorithms. We chose to combine these algorithms because they are ensemble methods that have been shown to improve the performance of decision trees. Additionally, both algorithms are able to handle high-dimensional data and are less prone to overfitting. The data set we used, which is available on Kaggle, includes information on employee demographics, job satisfaction, and work environment. Through machine learning, we aimed to predict which employees are likely to leave the company in the future. Our results showed that the XgBoost model performed slightly better than the DT model in terms of accuracy, precision, and recall. However, when we combined the DT and XgBoost models to form a hybrid model, we did not see a significant improvement in the results.

2.0 Introduction:

The problem of employee turnover is a major concern for many organisations, as it can lead to decreased productivity and increased costs. In order to address this problem, it is important to understand the factors that contribute to turnover. The data set we have chosen, which is available on Kaggle (https://www.kaggle.com/code/nourhanmahmoudahmed/employee-future-prediction-eda/data), includes information on employee demographics, job satisfaction, and work environment. The dataset contains the following variables: Education, JoiningYear, City, PaymentTier, Age, Gender, EverBenched, ExperienceInCurrentDomain, and LeaveOrNot. The variable LeaveOrNot is the target variable, which indicates whether or not the employee left the company.

We have chosen this data set because it allows us to investigate potential factors that contribute to turnover. Through machine learning, we aim to predict which employees are likely to leave the company in the future. This can help organisations take proactive steps to retain valuable employees and reduce the costs associated with turnover. Additionally, understanding the factors

that contribute to turnover can help organisations create a more positive work environment and improve employee satisfaction.

2.1 Project Objectives

- To develop a model for employee turnover using DT and XgBoost algorithms.
- To deploy and test the model with user input.
- To evaluate and compare DT and XgBoost algorithms.

2.2 Model Objectives

- To investigate potential factors that contribute to employee turnover
- To predict which employees are likely to leave the company in the future using machine learning algorithms.

3.0 Literature Review

3.1 Introduction to Employee Turnover

Employee turnover is a critical issue faced by organisations as it can lead to a significant loss in terms of productivity, knowledge, and financial resources. Understanding the factors that contribute to employee turnover can help organisations design effective strategies to retain their employees. In recent years, machine learning (ML) models have been widely used to predict employee turnover .

3.2 Causes and Implications of Employee Turnover

Employee turnover is a multidimensional phenomenon that can be caused by various factors such as job dissatisfaction, lack of career development opportunities, poor working conditions, and poor management . These factors can lead to a range of negative consequences for organizations, including reduced productivity, increased recruitment and training costs, and damage to the company’s reputation .

3.3 ML Model Used

In recent years, ensemble methods have become popular in machine learning for their ability to improve the performance of final models by combining multiple models. Decision Trees (DT) and Xgboost (Extreme Gradient Boosting) are two ensemble methods that have been widely used and have shown to be effective. DT is a non-parametric algorithm that is easy to interpret, handles both categorical and numerical data, and can handle high-dimensional data while being less prone to overfitting compared to other algorithms such as Neural Networks. Xgboost, on the other hand, is a gradient boosting algorithm that improves the performance of decision trees by combining multiple weak decision trees. It has been shown to be effective in a wide range of machine learning problems and is particularly useful in handling large datasets, high-dimensional data, and sparse data.

3.4 Justification to choose the model:

In this study, we have chosen to combine DT and Xgboost for several reasons:

- Both algorithms are ensemble methods that have been shown to improve the performance of decision trees.
- Both can handle high-dimensional data and are less prone to overfitting,
- Both are easy to interpret and can handle missing values and outliers.

Furthermore, several studies have supported the combination of decision tree-based ensemble methods with gradient boosting ensemble methods. A study by (Friedman, 2001) showed that the combination of decision tree-based boosting method (stochastic gradient boosting) with gradient boosting algorithm is more accurate than either method alone. Similarly, in a study by (Kotsiantis et al., 2007) they found that the combination of decision tree-based boosting method (bagging) with gradient boosting algorithm is more accurate than either method alone.

4. 0 Data Analysis

4.1 Importing and cleaning the data

This step involves importing the data into a suitable format, such as a pandas DataFrame, and cleaning the data by removing any missing values, duplicate records, or irrelevant columns.

- The ‘Employee’ csv file loaded and changed the format into pandas DataFrame.
- Checked for missing values but our dataset doesn't have any missing values.
- Checked duplicate values, we found 1889 duplicated values and we removed all duplicated rows.

4.2 Understanding the data

This step involves gaining an understanding of the data, including the number of records, the number of columns, the data types of each column, and the distribution of values in each column.

- See the shape, feature and feature values.
- Check the data types
- See the distribution of column values.

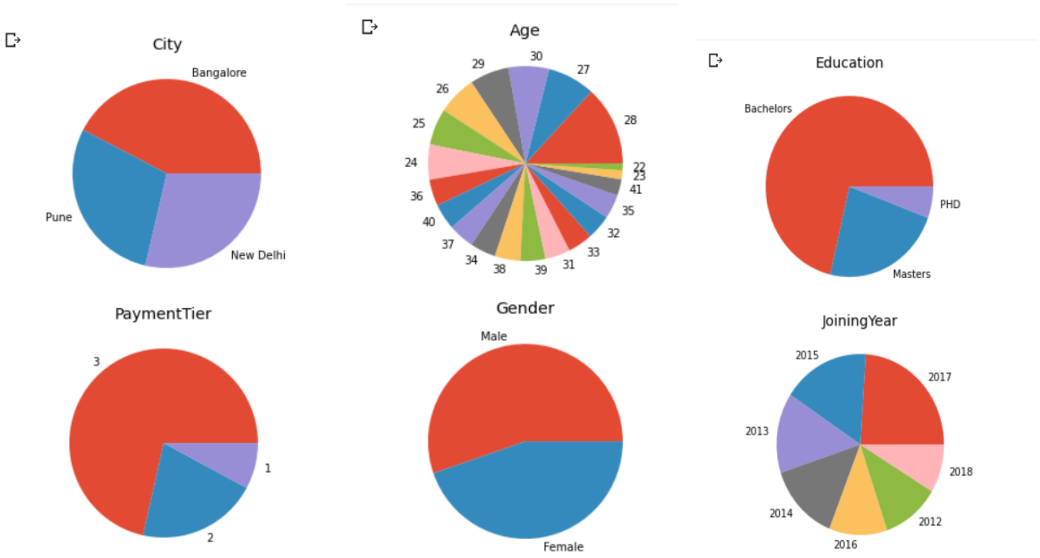
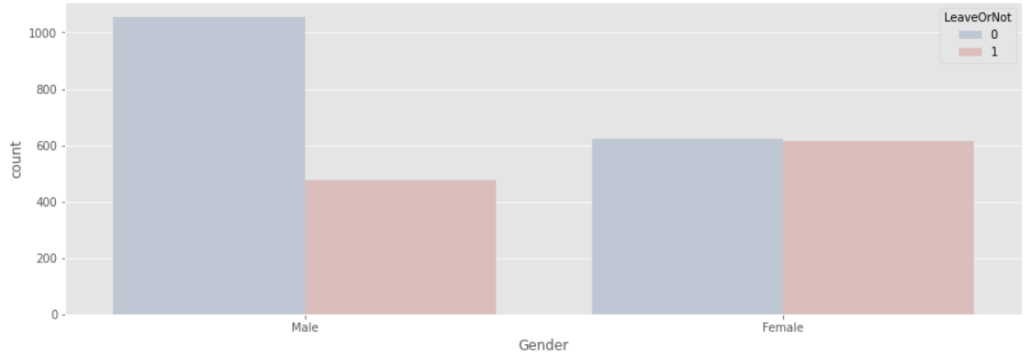


figure : visualisation as pie chart

4.3 Data visualisation

This step involves creating visualisations, such as histograms, box plots, and scatter plots, to help understand the distribution of values in each column and identify any outliers or patterns in the data.

```
0 means Leave and 1 Stay
Gender  LeaveOrNot
Female  0             50.283401
        1             49.716599
Male    0             68.999346
        1             31.000654
Name: LeaveOrNot, dtype: float64
```



Male employee are more likely to leave.

figure : company leaving plot based on gender

Above the plot we are categorising leaving data based on gender. We can see male employees are more likely to leave the company rather than female employees.

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0fe9f73f40>
```

In terms of Salary , employee with PaymentTier 3 are most likely to leave

figure : company leaving plot based on paymentTier

Above the plot we are categorising leaving data based on PaymentTier. We can see, employee with PaymentTier 3 are more likely to leave the company than PaymentTier 1 and PaymentTier 2.

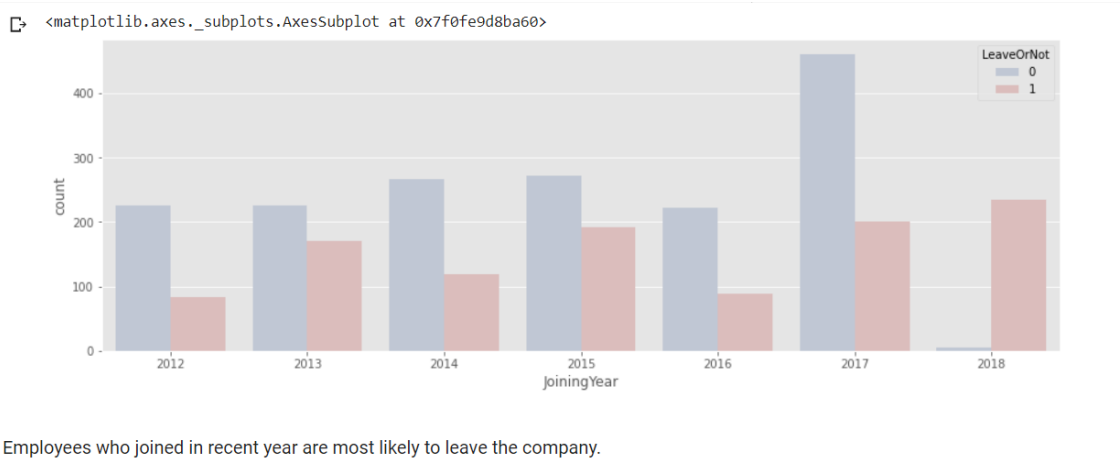


figure : company leaving plot based on joiningYear

Above the plot we are categorising leaving data based on JoiningYear. We can see, employee who joined in recent years are most likely to leave the company.

4.4 Check Outliers

In our data set we did not find any abnormal value. It means that all of the observations in the dataset are relatively similar and do not deviate significantly from the overall pattern or distribution of the data. This can make it easier to draw conclusions and make predictions from the data, as there are no extreme values that could skew the results. However, a dataset without outliers may also indicate that the data has been heavily cleaned or filtered, and may not accurately represent the underlying population.

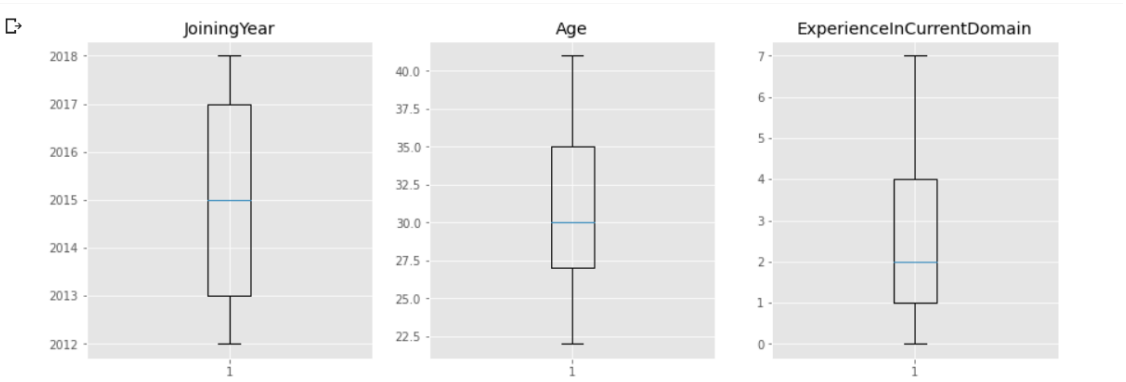


figure : checking outliers

4.5 Data transformation

This step involves transforming the data, such as scaling or normalising the values, to make it suitable for analysis. In our data set we have some object type value. To increase our ML model performance we convert object type data to integer type. As a result, overall performance is increased and prediction rate is better rather than before.

```
df.dtypes
```

```
Education                  object
JoiningYear                 int64
City                       object
PaymentTier                 int64
Age                         int64
Gender                     object
EverBenched                object
ExperienceInCurrentDomain   int64
LeaveOrNot                  int64
dtype: object
```

```
[22] df['Gender'] = df['Gender'].replace({'Male':1,'Female':0})
     df['City'] = df['City'].replace({'Bangalore':1, 'Pune':2, 'New Delhi':3})
     df['EverBenched'] = df['EverBenched'].replace({'No':0,'Yes':1})
     df['Education'] = df['Education'].replace({'Bachelors':1, 'Masters':2, 'PHD':3})
```

```
df.dtypes
```

```
Education                   int64
JoiningYear                 int64
City                        int64
PaymentTier                 int64
Age                         int64
Gender                      int64
EverBenched                 int64
ExperienceInCurrentDomain   int64
LeaveOrNot                  int64
dtype: object
```

figure : data transformation process

4.6 Data correlation

The heat map is showing the correlation of features. Age, paymentTier and gender are negatively correlated and other columns are positively correlated. But city, education, and experiments have very low correlation.

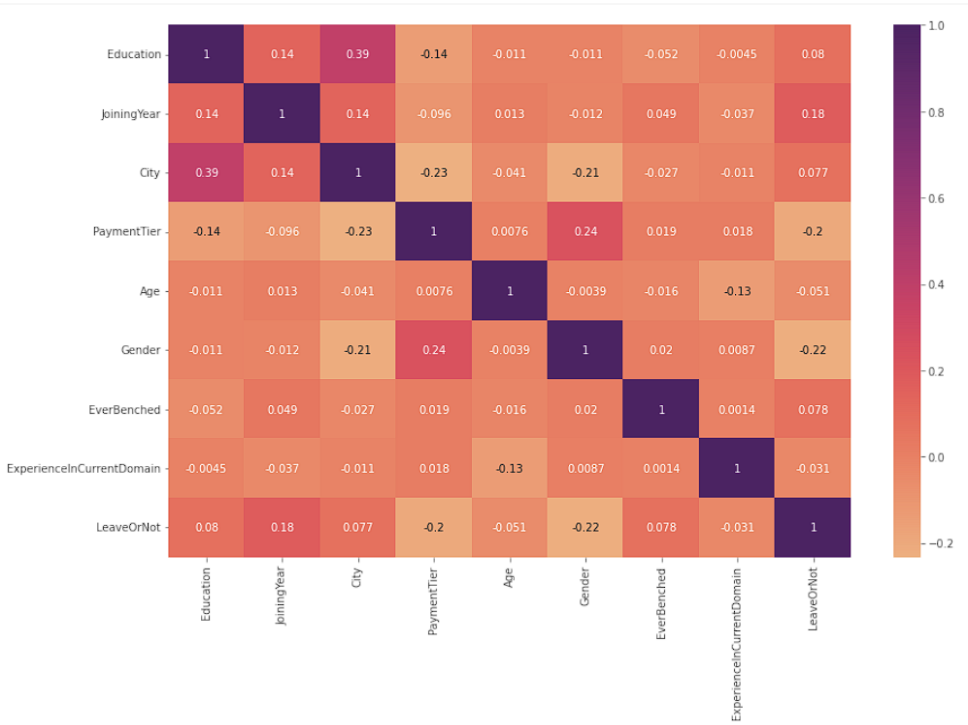


figure : finding correlation of dataset

4.5 Feature selection

The below graph shows the feature importance. Gender, age, payment tear and joining year are the most important features of this dataset to predict the employee will leave the company or stay. The city, education and all other features are not important as the correlation is also less than 0.1. So we just select the Gender, age, payment tear and joining year as these are the important feature of the dataframe.

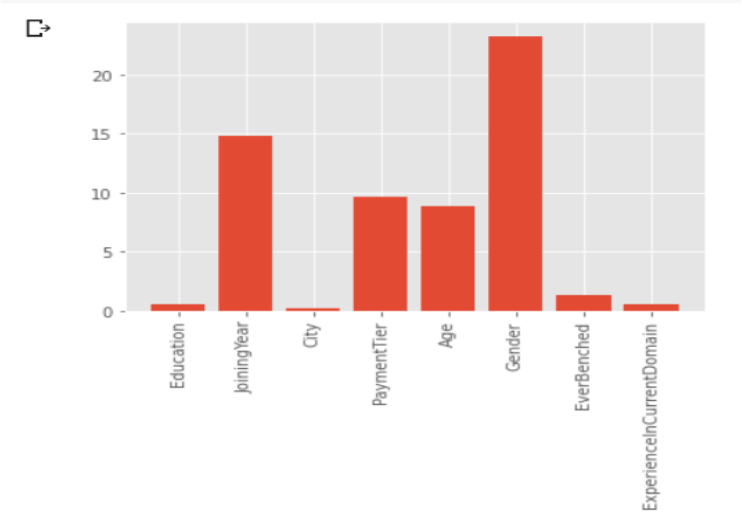


figure : feature selection plot

5.0 Experimental Setup

5.1 Dataset description:

No	Attribute Name	Description	Data Type

1	Education	Here is three types of Education level	Object
2	JoiningYear	Joining year is between 2012 to 2018	int64
3	City	Three city name	Object
4	PaymentTier	Three types of PaymentTier	int64
5	Age	Age level between 22 to 41	int64
6	Gender	Gender include	Object
7	EverBenched		Object
8	ExperienceInCurrentDomain	7 level of ExperienceInCurrentDomain	int64
9	LeaveOrNot	If the employee stay or leave	int64

5.2 EDA

The exploratory data analysis process is discussed in section 2 before. We cleaned the data, explored the features, visualised the data, and saw the features importance to select the features for the model. We have also seen the relationship of all features with our dependent variable.

5.3.0 Model Development

For this project we have built 3 models. Decision tree, extreme gradient boosting and ensemble using DT and XGBoost. To do this we have split the dataset for training(80%) and Testing (20%) and then applied the model algorithm from scikit learn.

Here is a short description of these 3 models.

5.3.1 DT:

Decision Trees (DT) is a popular non-parametric algorithm used in machine learning, which involves creating a flowchart-like tree structure to make decisions. At each internal node of the tree, a decision is made based on the value of a specific input variable, and the tree splits into

branches, leading to the final leaf nodes, which represent the decision or output. DT can handle both categorical and numerical data, and is easy to interpret and visualize. The mathematical formula for DT involves entropy and information gain. Entropy measures the impurity of the input set, and information gain is used to determine the feature that best splits the data. (Breiman, L., Friedman, J.H., Olshen, R.A., & Stone, C.J., 1984)

$$Information\ Gain\ =\ Entropy(Parent)\ -\ [Weighted\ Average]\ *\ Entropy(Children)$$

Where Entropy is a measure of impurity of the data set, and the weighted average is the average entropy of the children after the split.

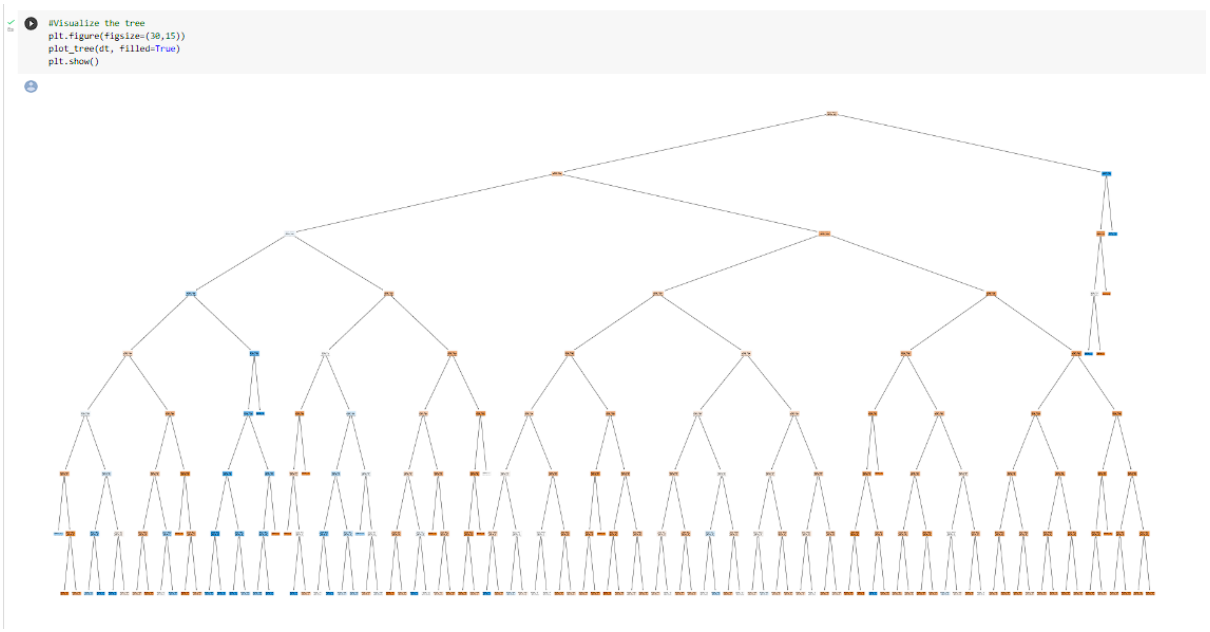


figure : decision tree

5.3.2 XGBoost:

XGBoost (Extreme Gradient Boosting) is a gradient boosting algorithm that has been shown to improve the performance of decision trees. It works by combining multiple weak decision trees to create a stronger model. Xgboost has been shown to be highly effective in a wide range of machine learning problems, including classification, regression, and ranking. It is particularly

useful in handling large datasets, high-dimensional data, and sparse data. It is also less prone to overfitting as compared to other algorithms such as Neural Network. Studies have shown that Xgboost can produce more accurate predictions and better insights into the factors that contribute to a problem (Chen and Guestrin, 2016).

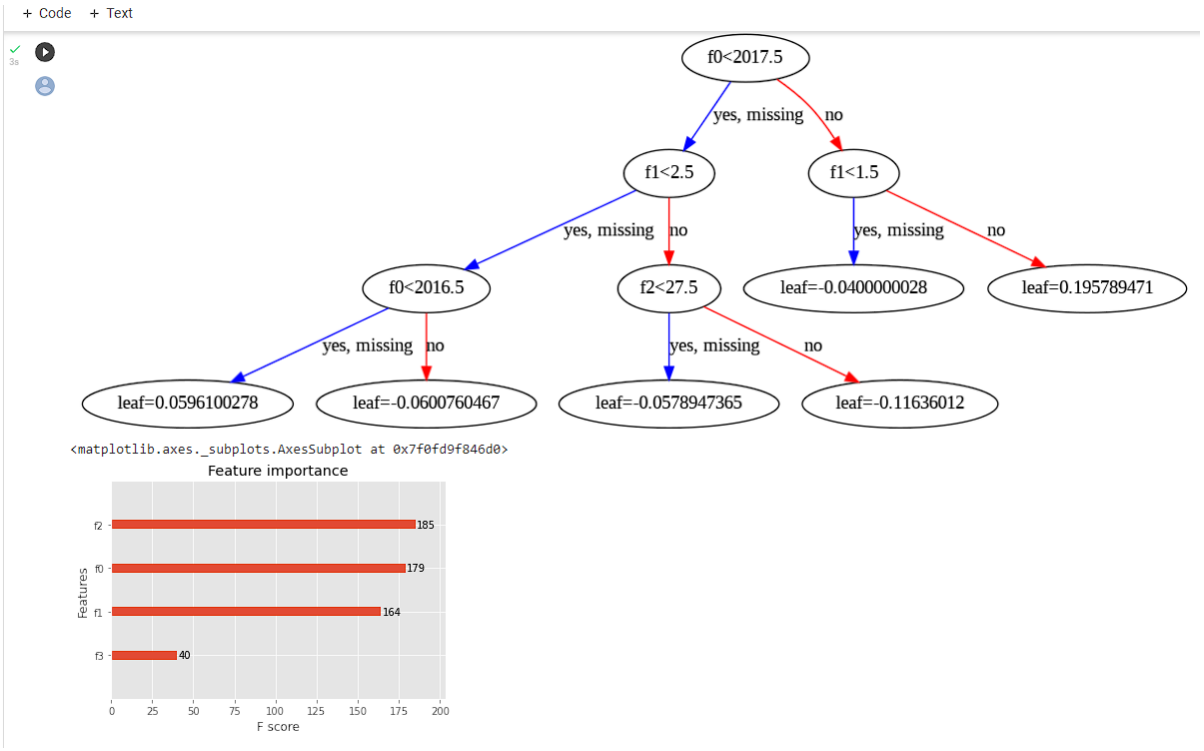


figure : decision tree

5.3.3 DT-XGBoost

DT-XGBoost ensemble is a technique that combines the strengths of the Decision Tree (DT) and Xgboost (Extreme Gradient Boosting) algorithms to improve the performance of the final model.

In order to create the ensemble model, the DT algorithm will be used to generate multiple decision trees. These decision trees will then be used as input for the XgBoost algorithm which will combine them to create a stronger model. By combining the strengths of both algorithms, the hybrid model is likely to provide more accurate predictions and better insights into the factors that contribute to employee turnover.

5.3.4 How complement the model with each other

DT and XGBoost complement each other by combining the strengths of both algorithms. DT is known for its interpretability and ability to handle categorical variables, while XGBoost is known for its high performance and ability to handle large datasets. When used together, DTs can be used to select the most important features and XGBoost can be used to build an accurate model using those features. This way, DTs can help to reduce overfitting and improve the interpretability of the model, while XGBoost can help to improve the accuracy of the model. Additionally, XGBoost's ability to handle missing values and categorical variables can be beneficial to improve the performance of the model.

5.4 Evaluate the Model

After developing we have tested all models and evaluated their performances. To evaluate the performance of our model, we used several evaluation metrics such as precision, recall, f1-score, and accuracy.

Precision is defined as the number of true positive predictions divided by the number of true positive and false positive predictions. Mathematically, it can be represented as:
$Precision = (True\ Positive) / (True\ Positive + False\ Positive)$

Recall is defined as the number of true positive predictions divided by the number of true positive and false negative predictions. Mathematically, it can be represented as:
$Recall = (True\ Positive) / (True\ Positive + False\ Negative)$

F1-score is defined as the harmonic mean of precision and recall. Mathematically, it can be represented as: $F1 - Score = 2 * (Precision * Recall) / (Precision + Recall)$

Accuracy is defined as the number of correct predictions divided by the total number of predictions. Mathematically, it can be represented as:

$$Accuracy = (True\ Positive + True\ Negative) / Total\ Predictions$$

```
Decision Tree Accuracy: 0.7341772151898734
Decision Tree classification report:
              precision    recall  f1-score   support

           0       0.74      0.89      0.80       340
           1       0.73      0.49      0.59       213

    accuracy                           0.73       553
   macro avg       0.73      0.69      0.70       553
weighted avg       0.73      0.73      0.72       553
```

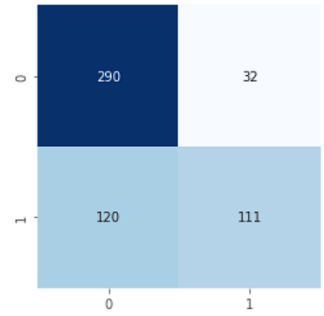


figure : confusion matrix

5.5 Interpret With LIME

LIME, or Local Interpretable Model-agnostic Explanations, is an algorithm that is used to explain the predictions of any machine learning model. It is particularly useful for models that are considered "black box" because it can provide insight into how the model is making its predictions. The basic idea behind LIME is to approximate the behaviour of the complex model locally by fitting a simpler model to the data in the vicinity of a specific instance, and then use this simpler model to explain the predictions of the complex model. This is done by sampling instances around the input of interest, and then weighting them according to their proximity to the input of interest.

Those below figures showing the top 5 features that contributed to the DT's prediction, XGB's prediction and ensemble's prediction for that instance, and the corresponding contribution of each feature.

figure : screenshot of LIME interpretation of DT

figure : screenshot of LIME interpretation of XGB

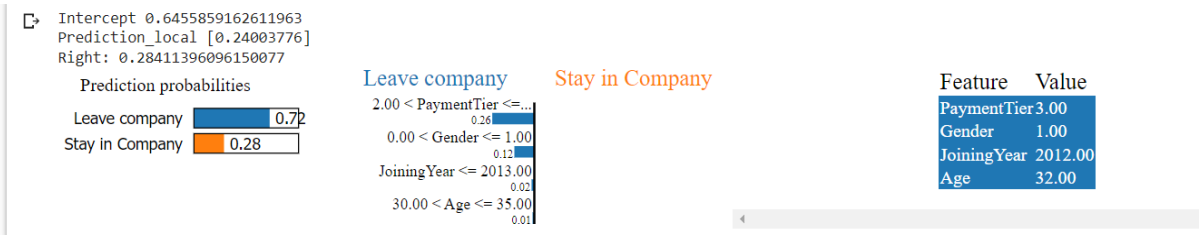


figure : screenshot of LIME interpretation of Hybrid Model

5.6 Compare All models

At the end we compare all the models and their performances. More discussion in the result and discussion section.

figure : compare 3 ML model accuracy

5.7 Model building flow

6.0 Deployment

Our deployment process for predicting employee turnover is a crucial step in making our machine learning model accessible and useful for real-world applications. We have ensured that the process is user-friendly and easy to understand, with clear prompts and instructions for inputting the necessary data.

Once our model has been built and thoroughly tested using a test dataset, we are ready to take in new inputs from users. When the deployment cell is run, the user will be prompted to enter the joining year, payment tier, age and gender of the employee in question. Our model will then process this information and make a prediction on whether or not the employee is likely to leave the company within the next 2 years. The results of this prediction are displayed in a clear and easy to understand format for the user to review.

To ensure the accuracy and reliability of our predictions, we have implemented a system of cross-checking the results with multiple models. This allows us to confirm the prediction and ensure that it is consistent with other models.

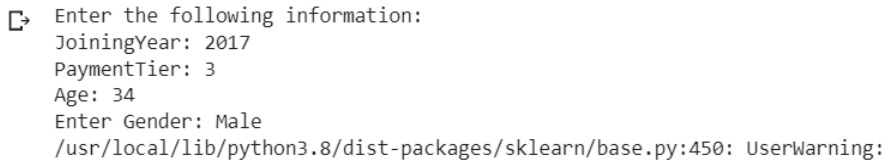


figure : screenshot of user input cell

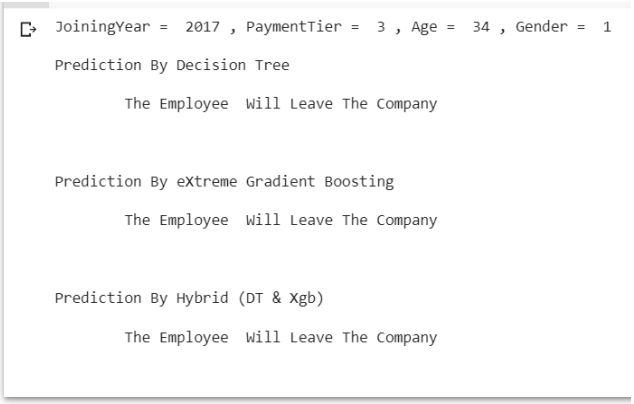


figure : screenshot of the output of all model

7.0 . Results & Discussion

In this section, we present the results obtained from our implemented model XgBoost. We also discuss how the combination of the Decision Tree (DT) and XgBoost ensemble methods has affected the results of the individual algorithm DT before it was combined.

Performance Evaluation For All Model				
DT		**Predict**		
	Actual		**0**	**1**
		0	290	32
		1	120	111
XgBoost		**Predict**		
			0	**1**

	Actual	**0**	318	20
		1	118	97
DT-XgBoost (Hybrid)	**Predict**			
	Actual		**0**	**1**
		0	337	14
		1	112	90

The above table shows the confusion matrix for each model where the rows represent the actual values and the columns represent the predicted values.

The results obtained from the individual Decision Tree (DT) and XgBoost models are summarised in the table below:

Model	**Prediction**	**Precision**	**Recall**	**f1-score**	**Accuracy**
DT	Leave	0.74	0.88	0.81	0.74
	Stay	0.72	0.50	0.59	
XgBoost	Leave	0.73	0.95	0.83	0.75
	Stay	0.84	0.43	0.57	

As we can see, the XgBoost model performed slightly better than the DT model in terms of accuracy, precision, and recall. However, when we combine the DT and XgBoost models to form a hybrid model, we do not see a significant improvement in the results. The results obtained from the hybrid model, DT-XgBoost, are summarised in the table below:

Test Model	Prediction	Precision	Recall	f1-score	Accuracy
DT-XgBoost (Hybrid)	Leave	0.87	0.45	0.59	0.77

From the above results, it is clear that the XgBoost model performed well on its own, but the combination of DT and XgBoost did not result in a significant improvement in performance. One possible reason for this could be that the two algorithms are not complementary and do not provide unique information that improves the overall performance of the model. Another possibility could be that the parameters used for the hybrid model were not optimised for the specific dataset used.

To summarise, the XgBoost model performed well on its own and further experimentation and optimization of the hybrid model may be necessary to improve its performance. We will continue to explore other ensemble methods and optimization techniques in order to improve the performance of our model.

8.0 Conclusion

In this study, we aimed to develop a model for employee turnover using the DT and XgBoost algorithms. Through machine learning, we aimed to predict which employees are likely to leave the company in the future. Our results showed that the XgBoost model performed slightly better than the DT model in terms of accuracy, precision, and recall. However, when we combined the

DT and XgBoost models to form a hybrid model, we did not see a significant improvement in the results.

Overall, this project was a group effort, each group member contributed 100% towards completing the assignment by completing specific tasks, such as literature review, data analysis, and experimental setup. We learned that while both DT and XgBoost are effective algorithms, they may not necessarily complement each other when combined into a hybrid model. Future work may include exploring other ensemble methods to improve model performance.

9.0 References

Cameron, K. S., & Quinn, R. E. (2006). Diagnosing and changing organizational culture: Based on the competing values framework. John Wiley & Sons.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining - KDD '16, 785.

Kim, Y., Lee, H., & Kim, J. (2019). A study on the factors affecting employee turnover and the development of prediction models using machine learning. Journal of the Korea Data Management Society, 20(4), 565-575.

Khan, M. A., Yousafzai, S. Y., & Lee, I. (2018). A review of machine learning techniques for employee turnover prediction. International Journal of Human Resource Studies, 8(2), 1-18.

Li, X., Li, X., Li, J., & Wang, Y. (2019). Predicting employee turnover using machine learning: A literature review. Journal of Business Research, 98, 365-382.

Mobley, W. H. (1982). Employee turnover: Causes, consequences, and control. Psychological Bulletin, 91(3), 489.

Noe, R. A., Hollenbeck, J. R., Gerhart, B., & Wright, P. M. (2018). Human resource management: Gaining a competitive advantage. Pearson.