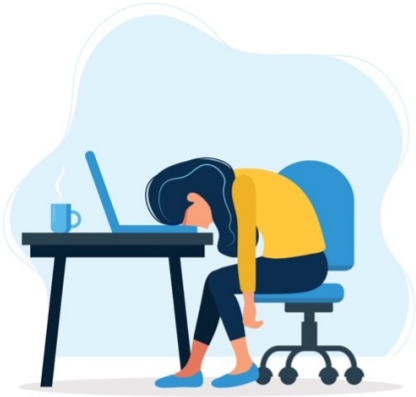
HR Analytics:

Prediction of Employee Attrition

BUDT704-0506: Data Processing and Analysis in Python Project

Fall 2022



Group: 14
Team: Sri Sai Alekya Ghanta, Sahil Kumar Rao, Sarvesh Rajwade, Varun Shrivastava, Pakshal Shah, Jarrar Haider

Under the guidance of

Dr. Peng Huang

UNIVERSITY OF MARYLAND
ROBERT H. SMITH SCHOOL OF BUSINESS

TABLE OF CONTENTS

1. Background

More than 4.25 million people in America quit their jobs in January 2022, up from 3.3 million in 2021 as reported by U.S. Bureau of Labor Statistics (BLS). Estimates show that the expense of replacing an existing employee is around one and a half to two times the annual salary of the existing employee.

According to a 2018 Gallup survey, the main contributors to employee burnout include:

- Unfair treatment at work
- Unmanageable workload
- Lack of role clarity
- Lack of management support and communication
- Unreasonable time pressure.

Attrition is the silent killer that can destroy even the most successful and stable organizations quickly in a startlingly short period of time. Recognizing the causes and predicting employee attrition is the crucial first step in understanding the issue and effectively resolving it before serious, long-lasting harm is done to the company.

2. Introduction

This project is based on the HR Analytics data set procured from Kaggle **.**Data was obtained in the CSV format and thereafter imported to Python for further analysis. It contains information about employees who are currently working or have previously worked in the company.

3. Mission Objectives

The goal of this project was to help Founders, CEO, Leadership Team and the HR Department to gain key insights regarding critical factors leading to employee attrition and predict employee attrition beforehand.

4. Data Overview

The dataset has 14,999 observations and 10 features.

Predictors (Columns)	Description
satisfaction_level	Satisfaction level of the employees (Ranging from 0 to 1)
last_evaluation	Last performance evaluation of the employe (Ranging from 0 to 1)
number_project	Number of projects handled by the employee
average_monthly_hours	Average number of hours per month worked by the employee
time_spend_company	Number of years an employee has spent in the company
Work_accident	Whether an employee has been involved in an accident at work (marked as 1) or not (marked as 0)
left	Whether an employee has left the company (marked as 1) or not (marked as 0)
promotion_last_5years	Whether an employee has been promoted in the last 5 years (marked as 1) or not (marked as 0)
Department	Department to which the employee belongs (Sales, Accounting, HR, Technical, Support, Management, IT, Product Management, Marketing, R&D)
salary	Salary level of the employee (Low, Medium, High)

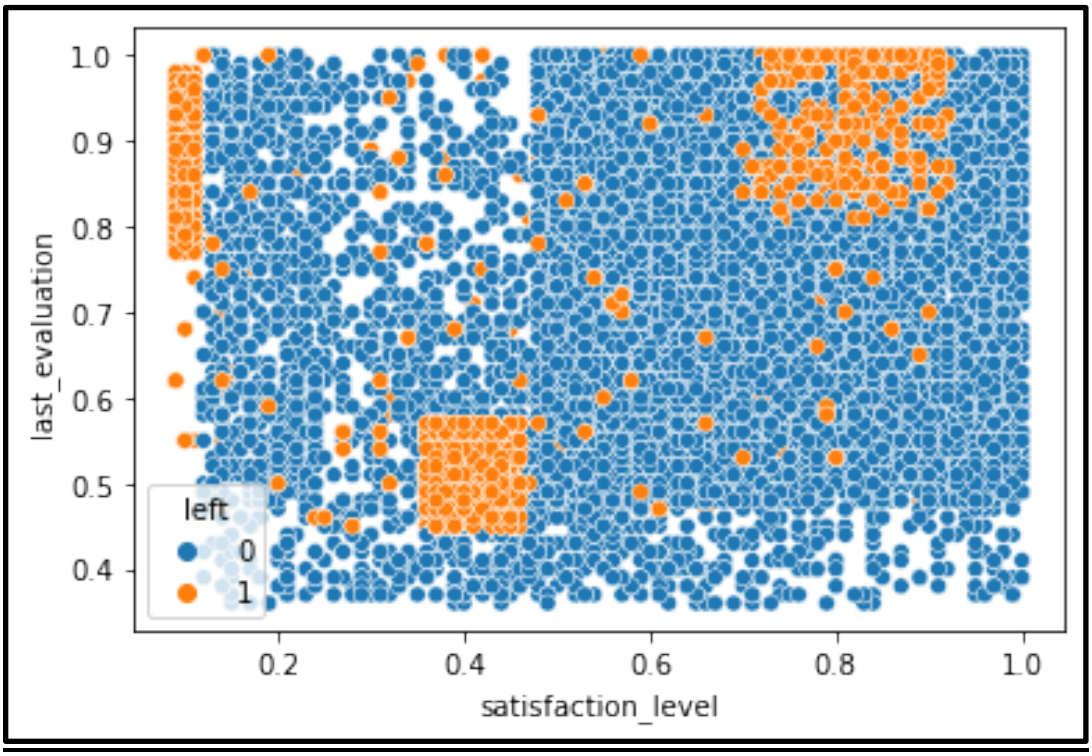
5. Approach and Methodology

5.1 Exploratory Data Analysis

Employee Statistics:

- Number of employees who left: 1991
- Number of employees who were not promoted: 11788
- Number of employees in each salary bracket: Low-5750, Medium:5261,
- High: 990
- Number of employees who did not meet with an accident at work: 10141
- Number of employees who stayed: 10000
- Number of employees who were promoted: 203
- Number of employees who met with an accident at work: 1850

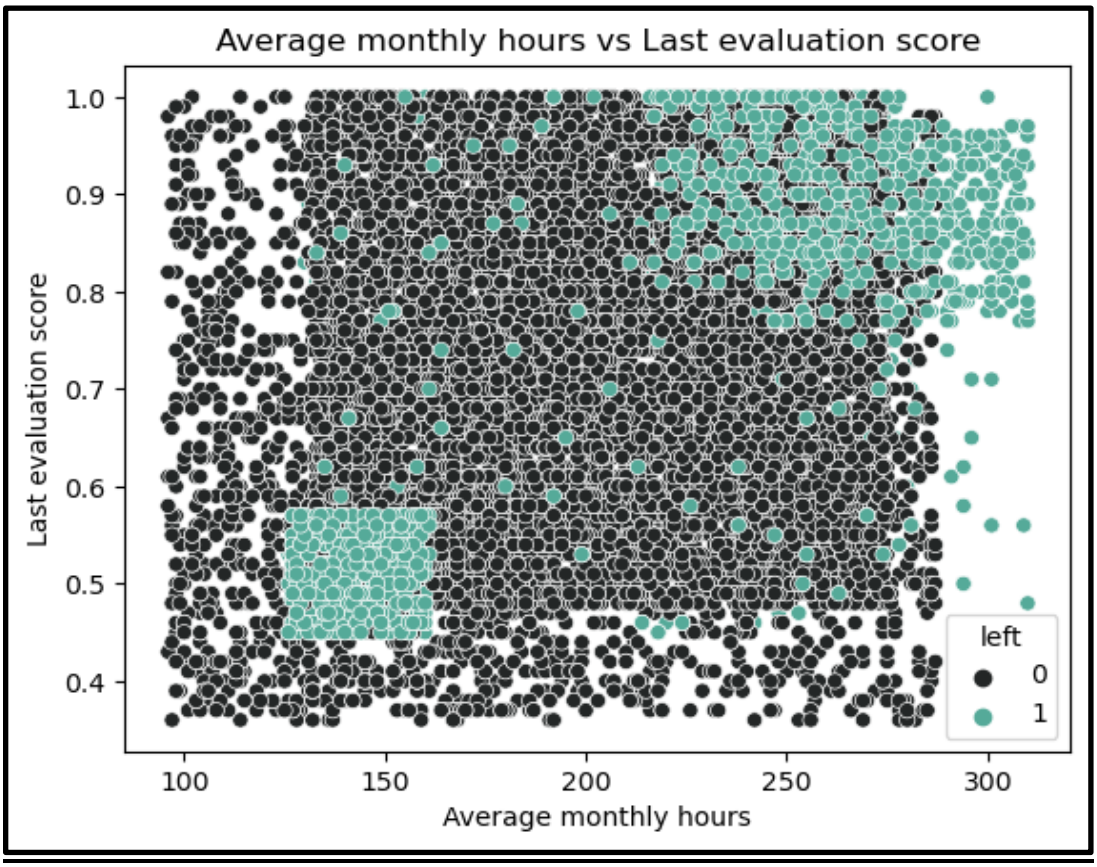
Last Evaluation vs Satisfaction Level:



From the scatter plot, 3 clusters can be observed - 1: employees with high evaluation score and low satisfaction score have left the company, 2: employees with low

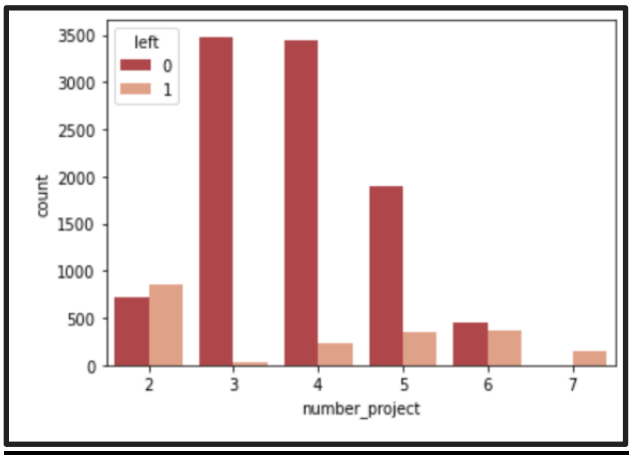
evaluation score and low satisfaction, 3:Employees with high evaluation score and high satisfaction level. Surprisingly, there are employees who left despite having high satisfaction levels and a high evaluation score.

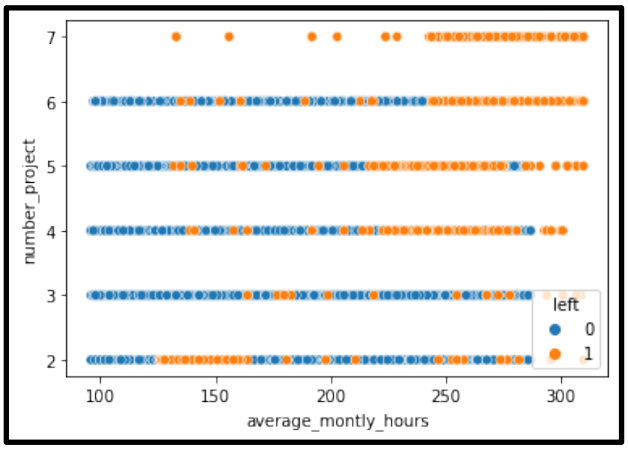
Average Monthly Hours vs Satisfaction Level :



From the scatter plot, 2 clusters can be observed - 1: employees who have worked for less number of hours per month with low evaluation score have left the company, 2: employees who have worked for more number of hours per month with high evaluation score have left the company.This shows that, the company has been losing valuable employees. It would be a huge loss to the company, if the employees who contributed the most number of hours leave.

Attrition statistics based on workload :

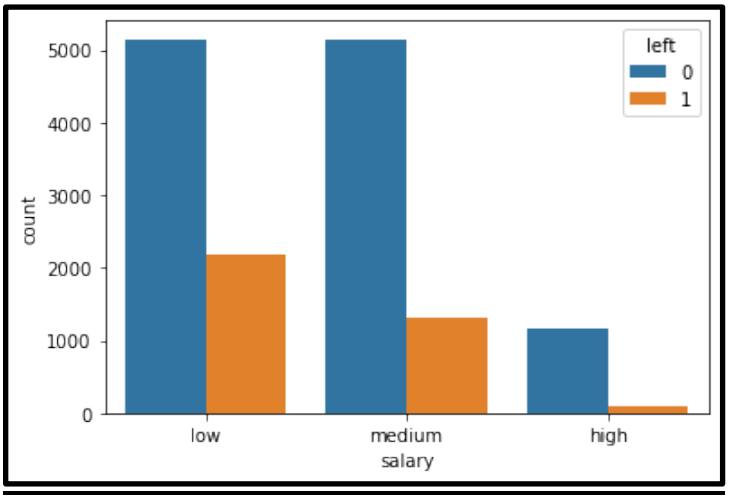




We can see that most of the employees working on 5 or more projects left the company. In fact, all employees who worked on 7 projects left the company. This could be due to the heavy workload and a disregard for work-life balance by the company.
On the other end of the spectrum, we see that a large proportion of employees working on 2 projects also left. It could be that the company fired them as they were not contributing enough.

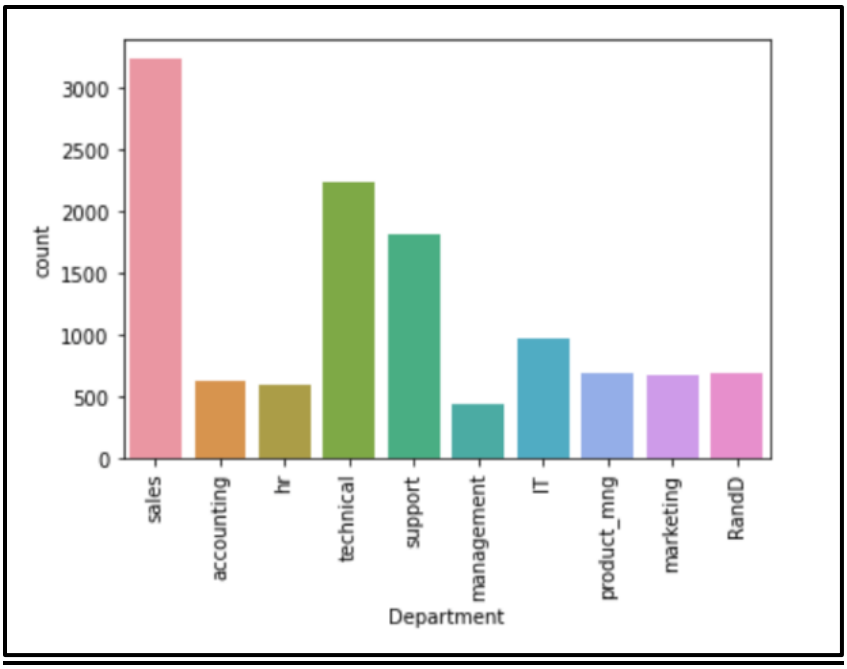
Additionally, we can observe that almost everyone leaves the company when the number of projects is 4, 5, 6 and 7 and the average monthly hours is over 200, indicating that the employees may be leaving due to an excessive amount of work. In contrast, when the number of projects is 6, 5, and 4, but the average monthly hours is not as high, the employees have stayed. This could indicate that when the workload is managed properly, employees do not leave the company.

Attrition statistics based on Salary:



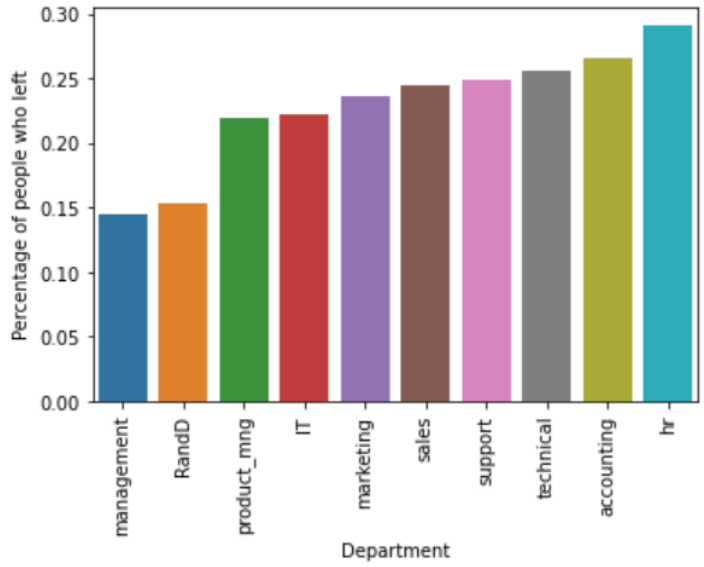
As expected, we see that a large proportion of employees that leave the company have either low or medium salary. There could be serious salary disparities between employees based on their position or the project or the department that they work in. This could need fixing by the company to improve satisfaction levels among the employees.

Employee count by Department:



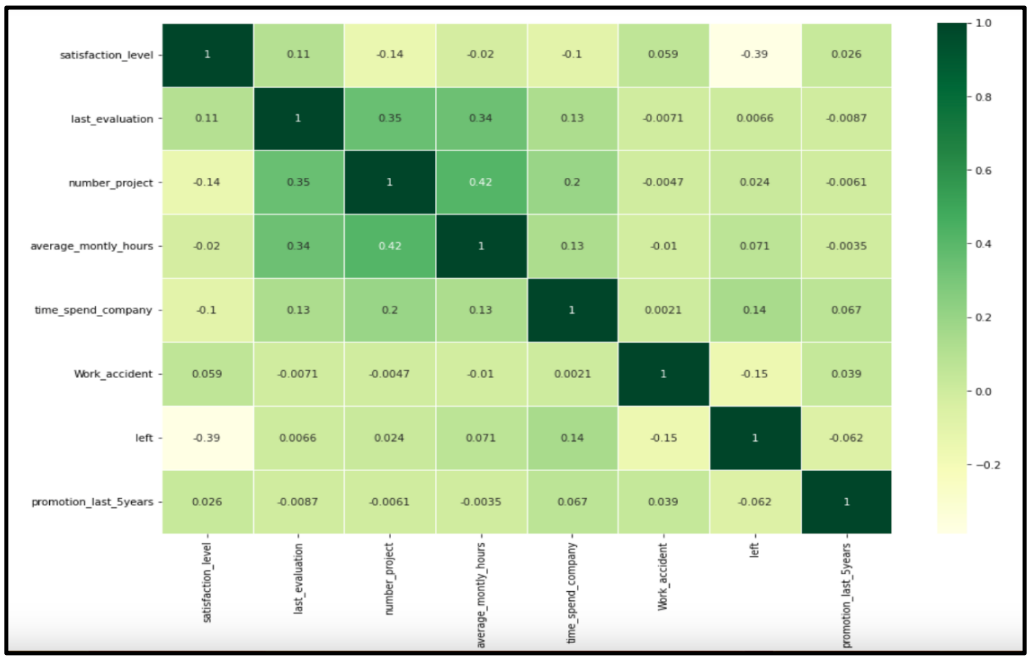
We explore the employee count department wise to understand which department has the most number of employees. This helps us to understand which department has the highest attrition rate.

Employees left proportion by Department:



We can see that a very small proportion of employees leave from the management and R & D departments. It could be that these departments have a good performance measure and the company is satisfied with their results

Correlation HeatMap:



There is no evidence of a very strong or very weak relationship of any feature with the Employee left feature. However, Satisfaction level seems to be the most strongly negatively correlated with an employee leaving company and Last evaluation seems to be the least correlated with an employee leaving company.

5.2 Data Preprocessing:

Handling duplicate values - There were 3,008 duplicate observations in the dataset which have been removed using .drop_duplicates(). The new dataset contains 11991 observations.

Before:

```
In [3]: HR_Data.shape
Out[3]: (14999, 10)
```

After:

```
In [7]: HR_Data.drop_duplicates(inplace=True)
        HR_Data.shape
Out[7]: (11991, 10)
```

Handling Missing values - There were no missing values in the dataset.

Handling Categorical Variables - The features salary and department in the dataset were categorical.These have been changed to numerical variables by creating dummy variables using the preprocessing.LabelEncoder() from sklearn library.

```
In [14]: from sklearn import preprocessing

         changing_categorical = preprocessing.LabelEncoder()

         HR_Data_2['salary']= changing_categorical.fit_transform(HR_Data['salary'])
         HR_Data_2['Department']= changing_categorical.fit_transform(HR_Data['Department'])
```

Removing Unnecessary Features - The feature Work_accident has been removed as it is only specific to certain departments like R&D. The new dataset contains 9 features and 11991 observations which will be taken to build the predictive model.

In [13]:
```
HR_Data_2 = HR_Data.copy()
HR_Data_2 = HR_Data.drop('Work_accident', axis=1)
HR_Data_2
```

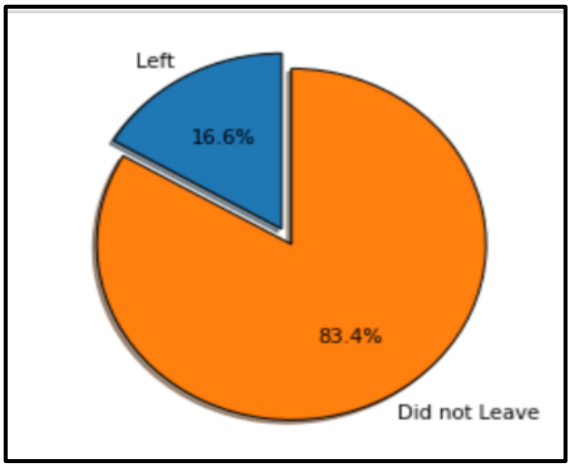
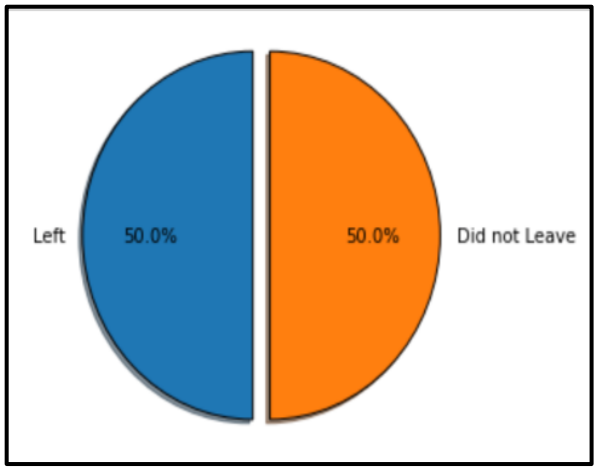
Out[13]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	left	promotion_last_5years	Department	salary
0	0.38	0.53	2	157	3	1	0	sales	low
1	0.80	0.86	5	262	6	1	0	sales	medium
2	0.11	0.88	7	272	4	1	0	sales	medium
3	0.72	0.87	5	223	5	1	0	sales	low
4	0.37	0.52	2	159	3	1	0	sales	low
...	...	...	...	...	...	...	...	...	...
11995	0.90	0.55	3	259	10	0	1	management	high
11996	0.74	0.95	5	266	10	0	1	management	high
11997	0.85	0.54	3	185	10	0	1	management	high
11998	0.33	0.65	3	172	10	0	1	marketing	high
11999	0.50	0.73	4	180	3	0	0	IT	low

11991 rows × 9 columns

5.3 Class Imbalance & Upsampling:

The dataset is imbalanced with more observations where Employee did not leave the company when compared to Employee left the company. However, the class of interest to us is Employee left the company. Therefore, in order to improve the performance of the prediction model, the class has been balanced by upsampling. Oversampling is chosen due to the low number of observations in the dataset, to prevent loss of information. The Employees left class has been upsampled using resample() function.

Raw Data (Before Upsampling)	Data (After Upsampling)
	

```
In [16]: from sklearn.utils import resample
         #create two different dataframe of majority and minority class
         df_majority = HR_Data_2[(HR_Data_2['left']==0)]
         df_minority = HR_Data_2[(HR_Data_2['left']==1)]
         # upsample minority class
         df_minority_upsampled = resample(df_minority,
                                          replace=True,    # sample with replacement
                                          n_samples=10000, # to match majority class
                                          random_state=42)  # reproducible results
         # Combine majority class with upsampled minority class
         df_upsampled = pd.concat([df_minority_upsampled, df_majority])
```

5.4 Model Building:

The data is now ready to be used for building models. The aim being to predict attrition of employees, this problem has been considered to be a classification problem. The data has been split into training (70%) and test (30%) to control for underfitting and overfitting. We built 4 Machine Learning Models on the data to predict employee attrition.

1)Logistic Regression -

```
In [60]: # Separating Dependent and Independent Features
         Y = df_upsampled['left']
         X = df_upsampled.drop('left', axis=1)
```

```
In [61]: from sklearn.linear_model import LogisticRegression
         from sklearn.model_selection import train_test_split
         Xlr_train, Xlr_test, Ylr_train, Ylr_test = train_test_split(X,Y, test_size = 0.3, random_state = 42)
         classifierlog = LogisticRegression(max_iter=600)
         classifierlog.fit(Xlr_train, Ylr_train)
         Ylr_pred = classifierlog.predict(Xlr_test)
         from sklearn.metrics import confusion_matrix, accuracy_score
         cmlr = confusion_matrix(Ylr_test, Ylr_pred)
         print(cmlr)
```

2) RandomForest -

```
In [24]: from sklearn.ensemble import RandomForestClassifier
         classifierRand = RandomForestClassifier(max_depth=3, n_estimators=150,
             min_samples_split=2, max_leaf_nodes=8,
             random_state=22)
         Xrf_train, Xrf_test, Yrf_train, Yrf_test = train_test_split(X,Y, test_size = 0.3, random_state = 42)
         classifierRand.fit(Xrf_train, Yrf_train)
         Yrf_pred = classifierRand.predict(Xrf_test)
         from sklearn.metrics import confusion_matrix, accuracy_score
         cmrf = confusion_matrix(Yrf_test, Yrf_pred)
         print(cmrf)
```

3) Decision Tree -

```
In [27]: from sklearn.tree import DecisionTreeClassifier
         # Train our decision tree
         tree = DecisionTreeClassifier(random_state=10,max_depth=3)
         Xdt_train, Xdt_test, Ydt_train, Ydt_test = train_test_split(X,Y, test_size = 0.3, random_state = 42)
         tree.fit(Xdt_train, Ydt_train)
         # Predict the labels for the test data
         Ydt_pred = tree.predict(Xdt_test)
         from sklearn.metrics import confusion_matrix, accuracy_score
         cmdt = confusion_matrix(Ydt_test, Ydt_pred)
```

4) Naive Bayes -

```
In [32]: from sklearn.naive_bayes import GaussianNB
         nb = GaussianNB()
         XGB_train, XGB_test, YGB_train, YGB_test = train_test_split(X,Y, test_size = 0.3, random_state = 42)
         nb.fit(XGB_train, YGB_train)
         # Predict the labels for the test data
         YGB_pred = nb.predict(XGB_test)
         from sklearn.metrics import confusion_matrix, accuracy_score
         cmGB = confusion_matrix(YGB_test, YGB_pred)
         cmGB
```

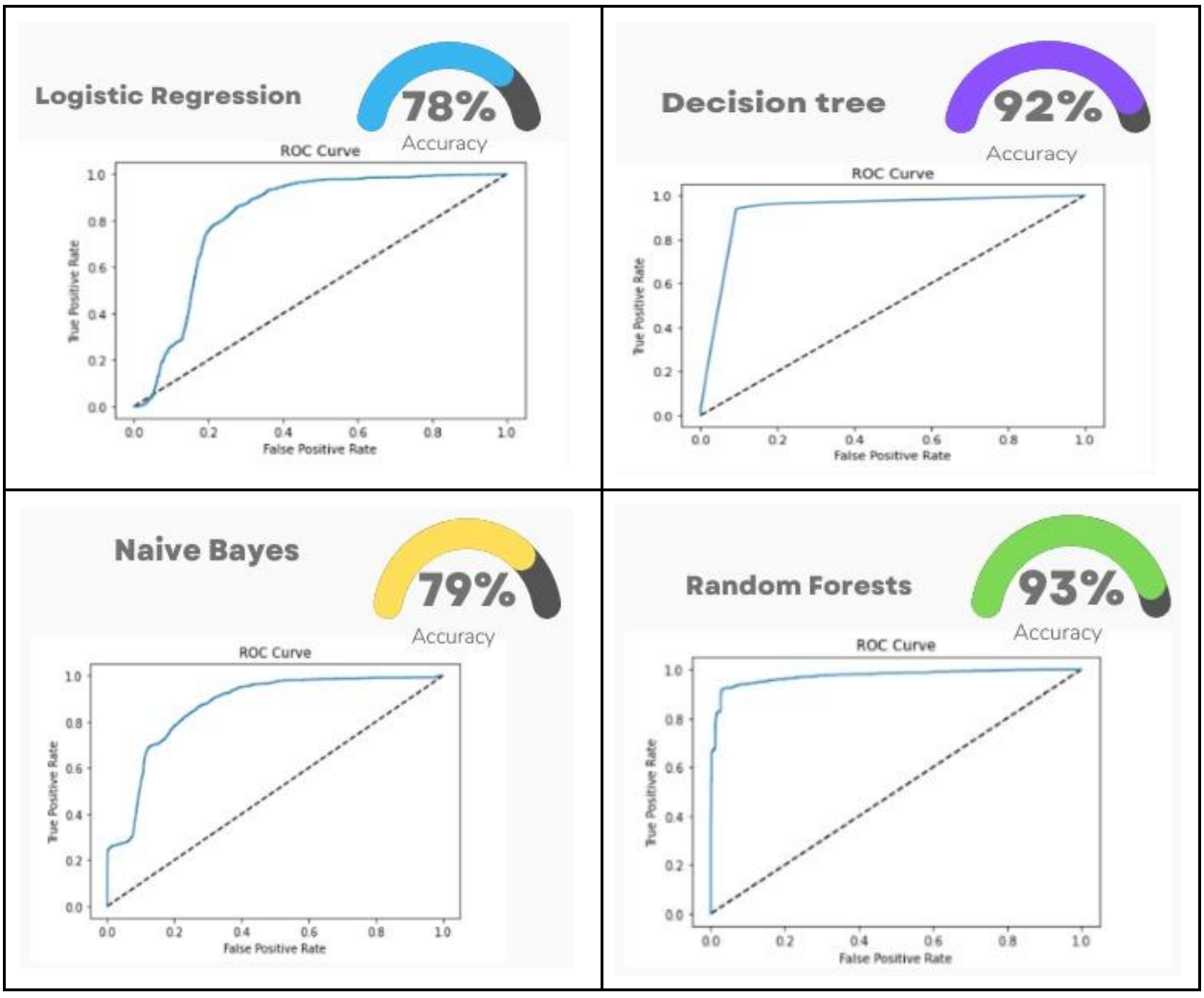
5.5 Model Performance Comparison:

Confusion matrix:

Model Types	Accuracy	False +ve (%)	False -ve(%)
Logistic Regression	78.3%	57.1%	42.9%
Naive Bayes	79.3%	76.7%	23.2
Random forests	93.1%	39.4%	60.6%
Decision Tree	92.3%	59.9%	40.1%

- RandomForest seems to have the highest accuracy followed by Decision Tree, Logistic Regression and Naive Bayes Models.
- NaiveBayes model has the highest False positive rate showing poor performance.
- Logistic Regression model shows lower False positive rate compared to Decision Tree. Therefore, Logistic Regression can also be considered as a good model for prediction.

ROC Curves:



- RandomForest seems to have the highest accuracy with the most area under the ROC curve representing optimal compromise between sensitivity and specificity, followed by Decision Tree, Logistic Regression and Naive Bayes Models
- However, the Random Forest model might have an overfitting issue.

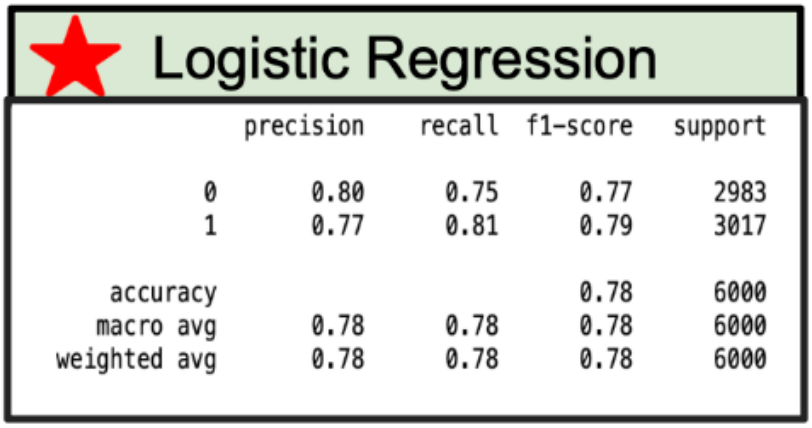
Classification Report :

To compare the performances of each model that we have used, we create a classification report for each one. Some of the important measures taken into account are:

Accuracy: This is just the ratio of the correctly predicted outcomes to the total number of observations. This is not always a good measure as it relies on the dataset being symmetric.

Precision : For our model, this answers the question: Of all the employees that we labeled as left, how many actually left. So, this takes into account the false positives.

Recall : This metric is the answer to the question of all the employees that actually left, how many did we correctly label. This takes into account the false negatives.



	precision	recall	f1-score	support
0	0.80	0.75	0.77	2983
1	0.77	0.81	0.79	3017
accuracy			0.78	6000
macro avg	0.78	0.78	0.78	6000
weighted avg	0.78	0.78	0.78	6000

Decision Tree

	precision	recall	f1-score	support
0	0.94	0.91	0.92	2983
1	0.91	0.94	0.93	3017
accuracy			0.92	6000
macro avg	0.92	0.92	0.92	6000
weighted avg	0.92	0.92	0.92	6000

Random Forest

	precision	recall	f1-score	support
0	0.93	0.95	0.94	2983
1	0.95	0.93	0.94	3017
accuracy			0.94	6000
macro avg	0.94	0.94	0.94	6000
weighted avg	0.94	0.94	0.94	6000

Naive Bayes

	precision	recall	f1-score	support
0	0.88	0.68	0.77	2983
1	0.74	0.90	0.81	3017
accuracy			0.79	6000
macro avg	0.81	0.79	0.79	6000
weighted avg	0.81	0.79	0.79	6000

6. Results

- We did some EDA investigations to gain insights and discover patterns
- Satisfaction level variable seems to be the most strongly negatively correlated with an employee leaving company
- Last evaluation seems to be the variable which is least correlated with an employee leaving company
- Employees who worked on the most projects and put in the most hours per month on average almost always left the company.
- Naive Bayes has a very low recall value. This means it is incorrectly predicting that most employees will leave even when they don't. Therefore, overall Naive Bayes may not be the best choice.
- Random Forest and Decision Tree have very high precision and recall which could be too good to be true. Maybe the model is overfitting and we need to test it on a larger sample size.
- Logistic Regression has modest values compare to others but it is still a good model

7.References

1. Qureshi, F. (2021) *HR analytics job prediction*, *Kaggle*. Available at: https://www.kaggle.com/datasets/mfaisalqureshi/hr-analytics-and-job-prediction?resource=download&select=HR_comma_sep.csv (Accessed: December 16, 2022). Link: Kaggle Dataset Source
2. People Managing People and Merwe, M.van der (2022) *Employee retention statistics and insights 2022*, *People Managing People*. Available at: https://peoplemanagingpeople.com/articles/employee-retention-statistics/ (Accessed: December 16, 2022). Link: People Managing People Article