Symbiosis Institute of Technology, Pune

Mini Project Report

[illegible]

Name: Parth Singh
PRN: 21070126062
Branch: AIML A3
Email: singh.parth.btech2021@sitpune.edu.in

Problem Statement:

Employee Attrition Prediction

Objectives:

1. To find possible reasons for employee attrition, in order to prevent valuable employees from leaving.

2. Evaluating possible trends and reasons for employee attrition, in order to prevent valuable employees from leaving.

3. Help companies to be prepared for future employee loss, and to categorize them under High Performance Rating and Low Performance Rating.

Dataset Description:

Dataset: IBM HR Analytics Employee Attrition and Performance

The dataset was made available under Open Data Commons Open Database License (ODbL) through Kaggle on January 28, 2019.The dataset is a fictional dataset created by IBM data scientists on 1470 employees' data as rows.

It consists of 35 columns explaining the different parameters and features of an employee.

The dataset is mixture of various categorical and numeric features and the objective is predict a binary output of employee attrition, in the form of 'yes' or 'no'.

The dataset has no null values, and consists of 26 numerical features and 9 categorical variables.

Exploratory Data Analysis:

On the basis of studying all the columns and its respective categorical variables, numeric variables and its outliers, some sub-groups can be derived for our analysis:

1. Impact of Finances

2. Impact of Work Convenience

3. Impact of Educational Field

4. Impact of Work Role

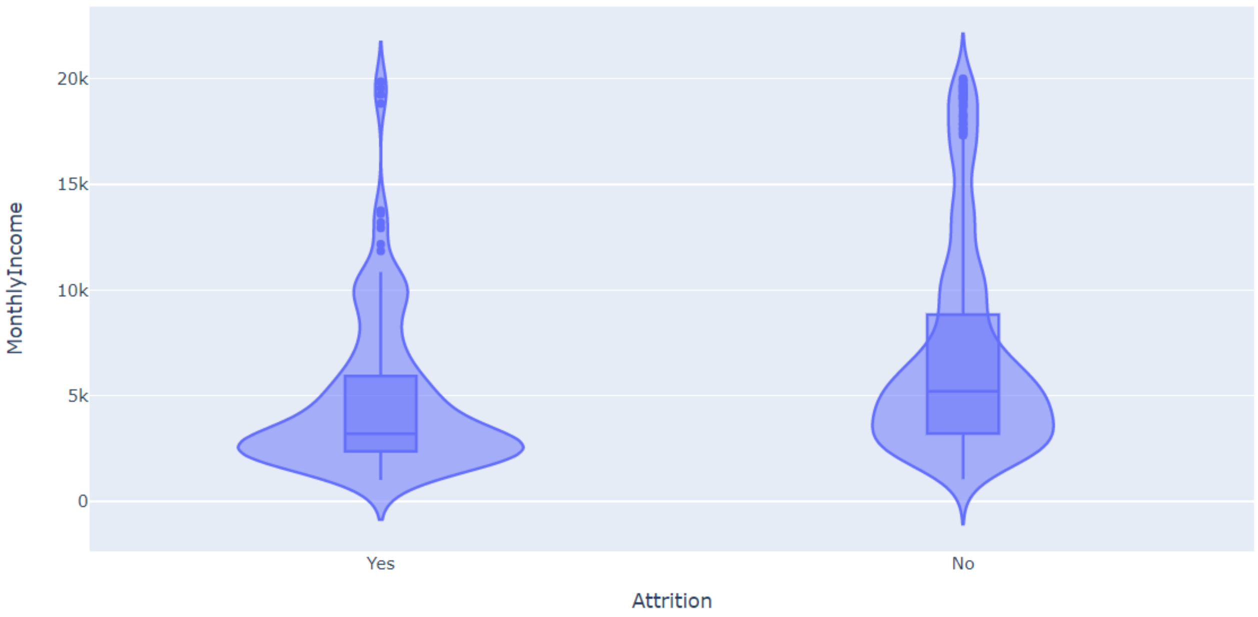
5. Impact of Personal Life

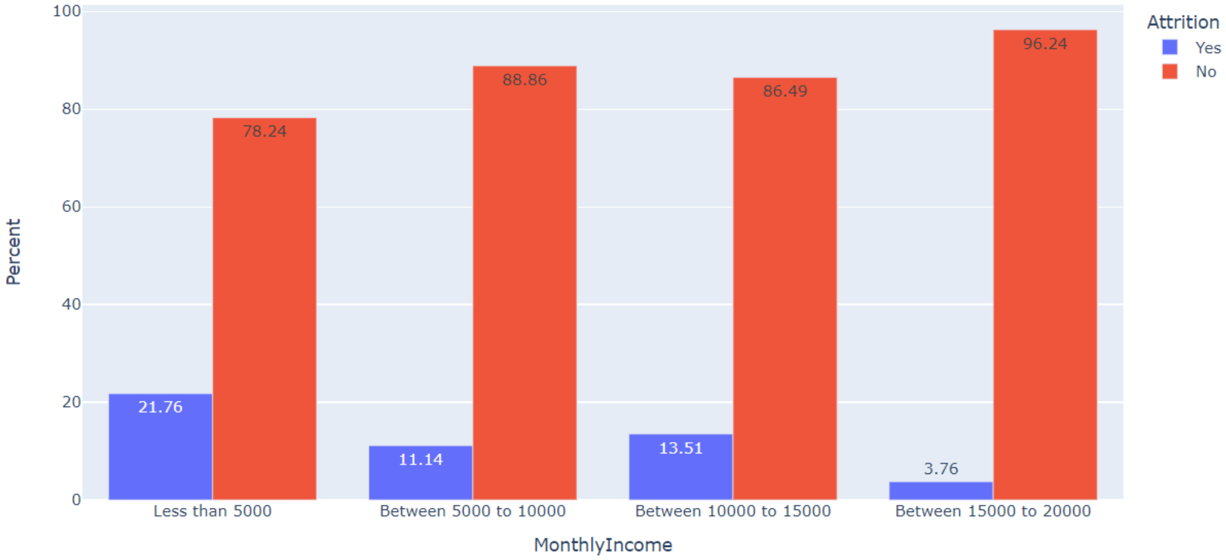
6. Impact of Work Experience

Impact of Finances

Columns Considered: 1. Monthly Income
2. Percent Salary Hike
3. Stock Option Level

1. Monthly Income

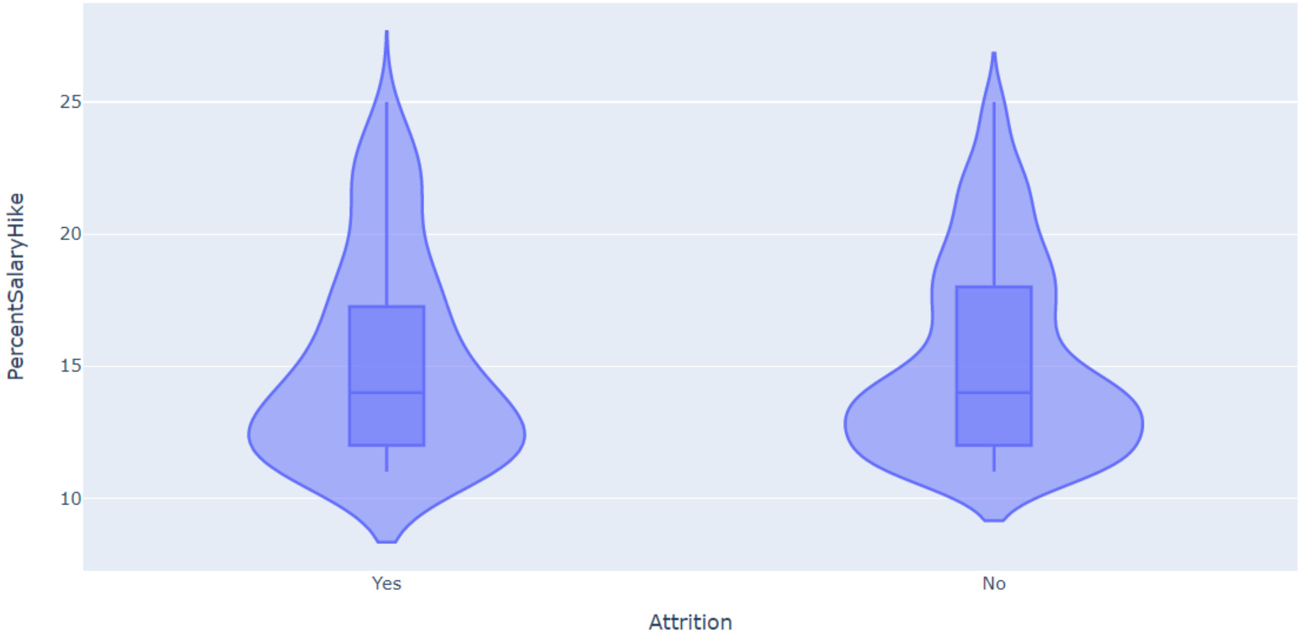


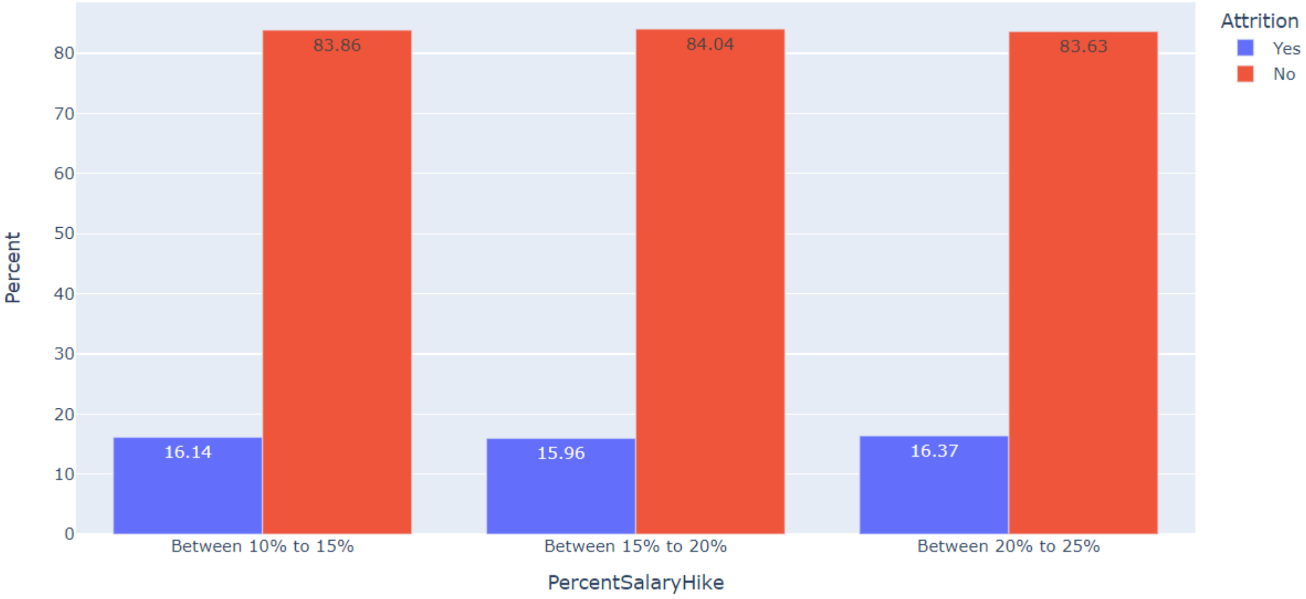


Outcome: The common observation comes that employees resigned had a fairly lesser median salary than those currently in the company.

Attrition rate has a positive correlation with salary, higher the salary, lesser the attrition rate.

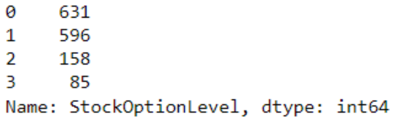
2. Percent Salary Hike



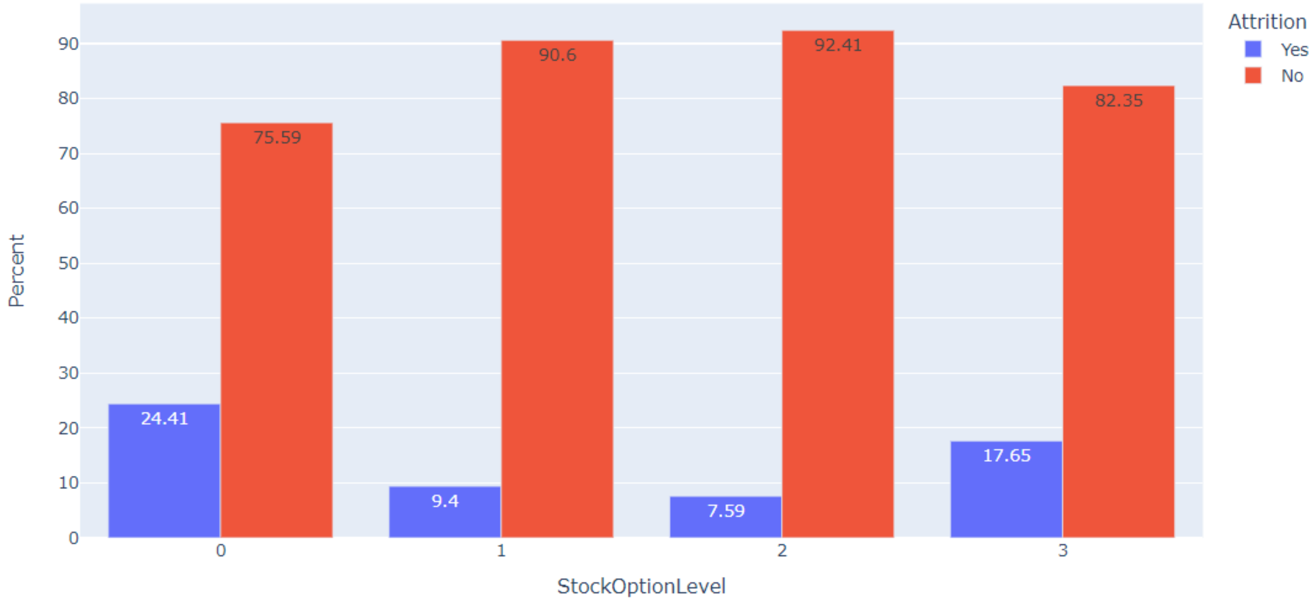


Outcome: Percent Salary hike doesn't have a very major impact in predicting attrition rate. The average attrition rate is nearly same irrespective of a higher percent increase in salary.

3. Percent Salary Hike



```
0    631
1    596
2    158
3     85
Name: StockOptionLevel, dtype: int64
```



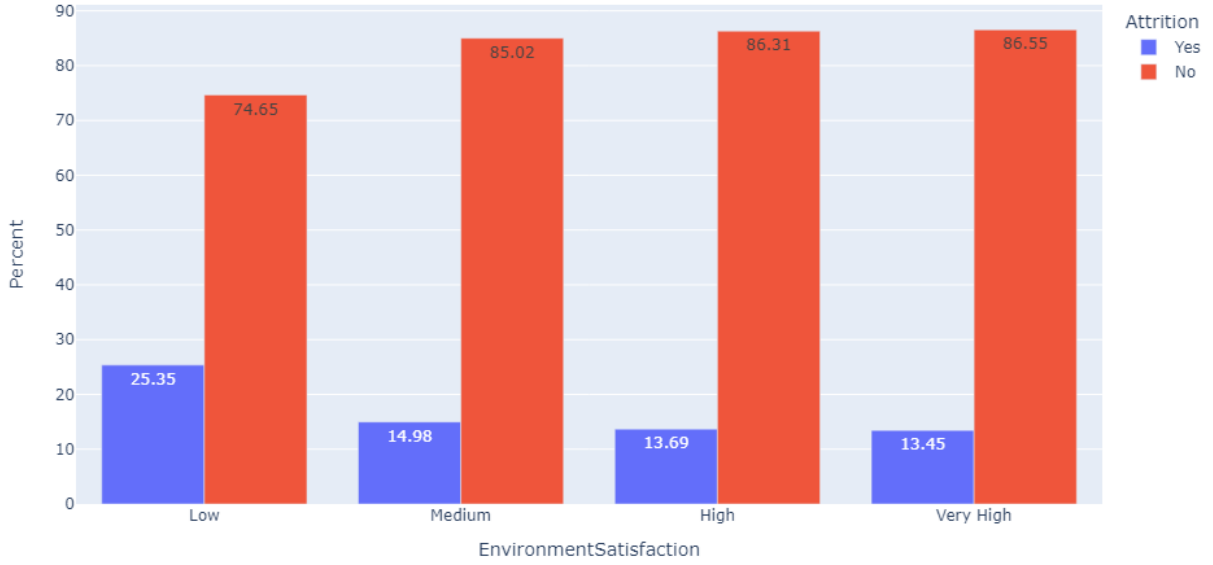
Outcome: Almost 50% of the employees don't have a stock option level and they contribute with the highest attrition rate.

Employees with stock option level 1 and 2 have the lowest attrition rate.

Impact of Finances

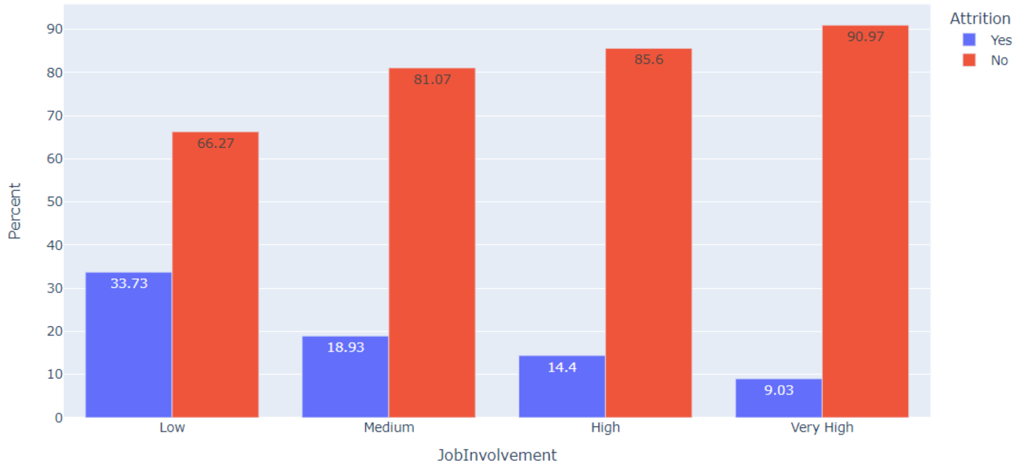
Columns Considered: 1. Environment Satisfaction
2. Job Involvement
3. Job Satisfaction
4. Performance Rating

1. Environment Satisfaction:



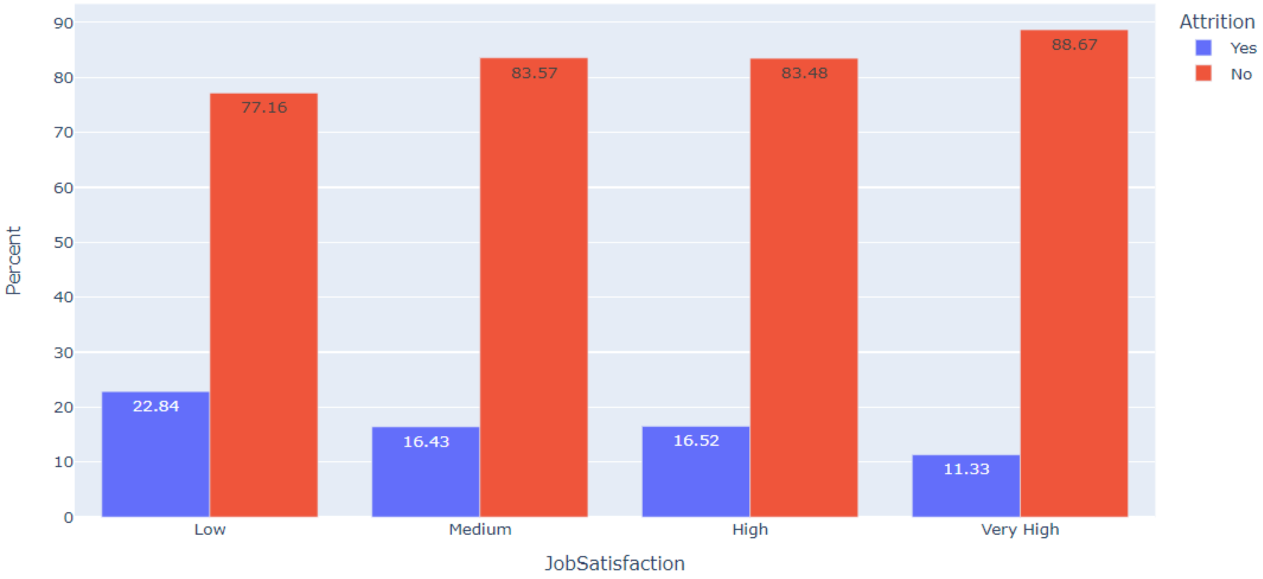
Outcome: Majority employees contribute to an above average satisfaction level in the company and havee the least attrition rates. Employees with low environemnt satisfaction rate have the highest attrition rate. Environment satisfaction has a direct coreelation with attrition levels.

2. Job Involvement:



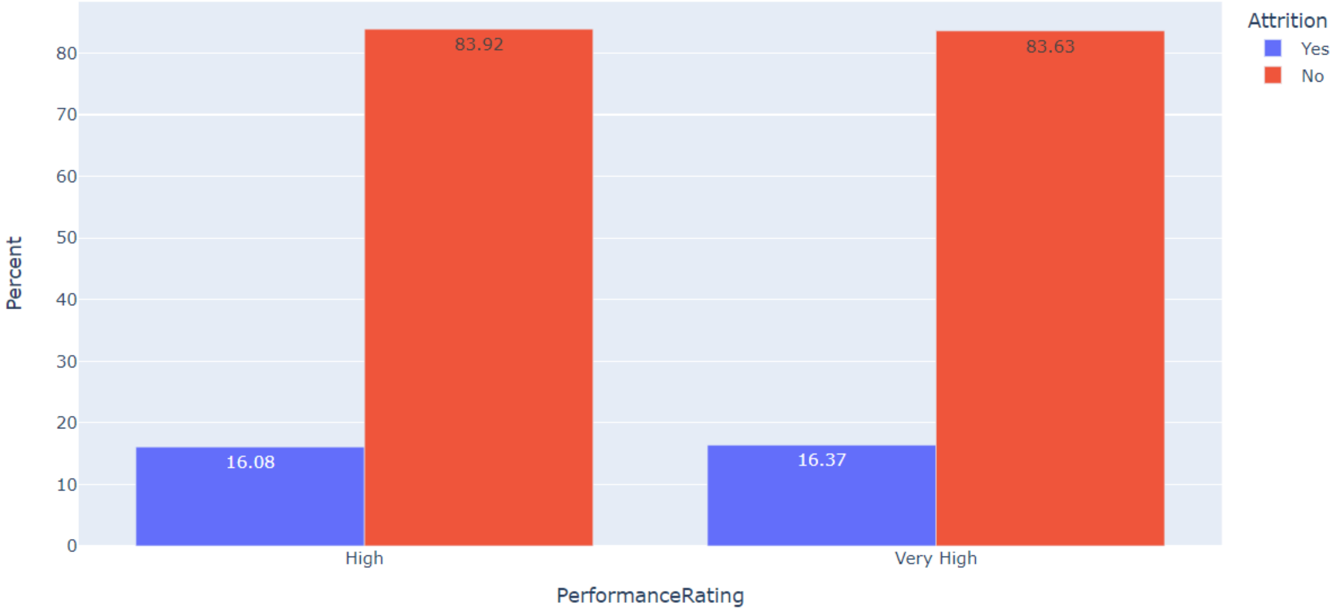
Outcome: Employees with low job involvement have a very high attrition rate.A greater job involvement promises a lesser of attrition rates.

3. Job Satisfaction:



Outcome: Environment Satisfaction, Job Involvement and Job Satisfaction are the show direct influence on the attrition rate.

4. Performance Rating:

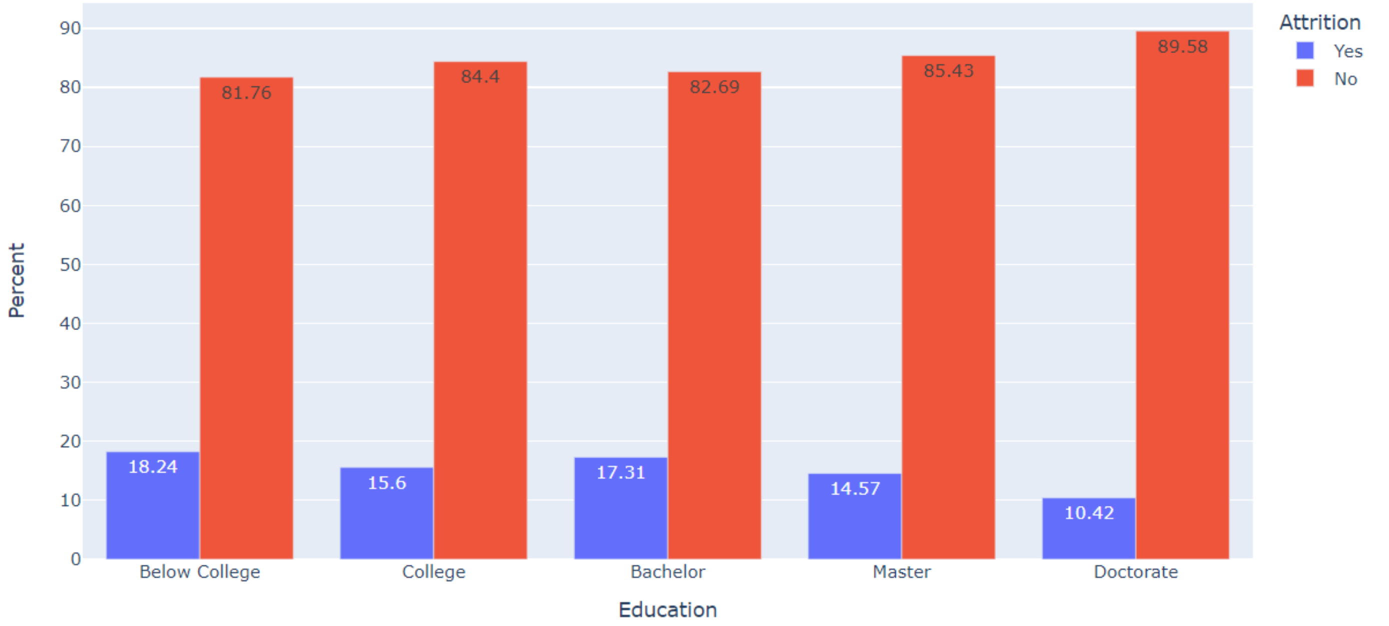


Outcome: Performance rating is seen to have no differentiable impact on attrition levels. Employees with a higher performance rate are to have nearly same attrition rates.

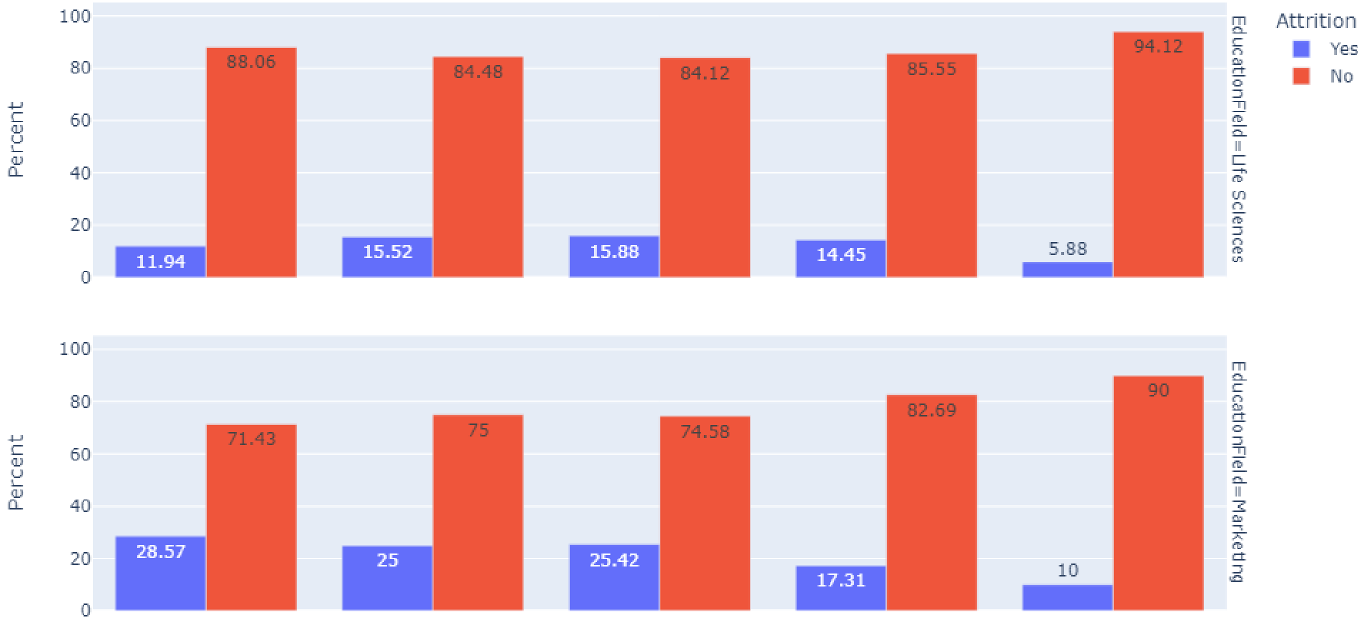
Impact of Educational Field

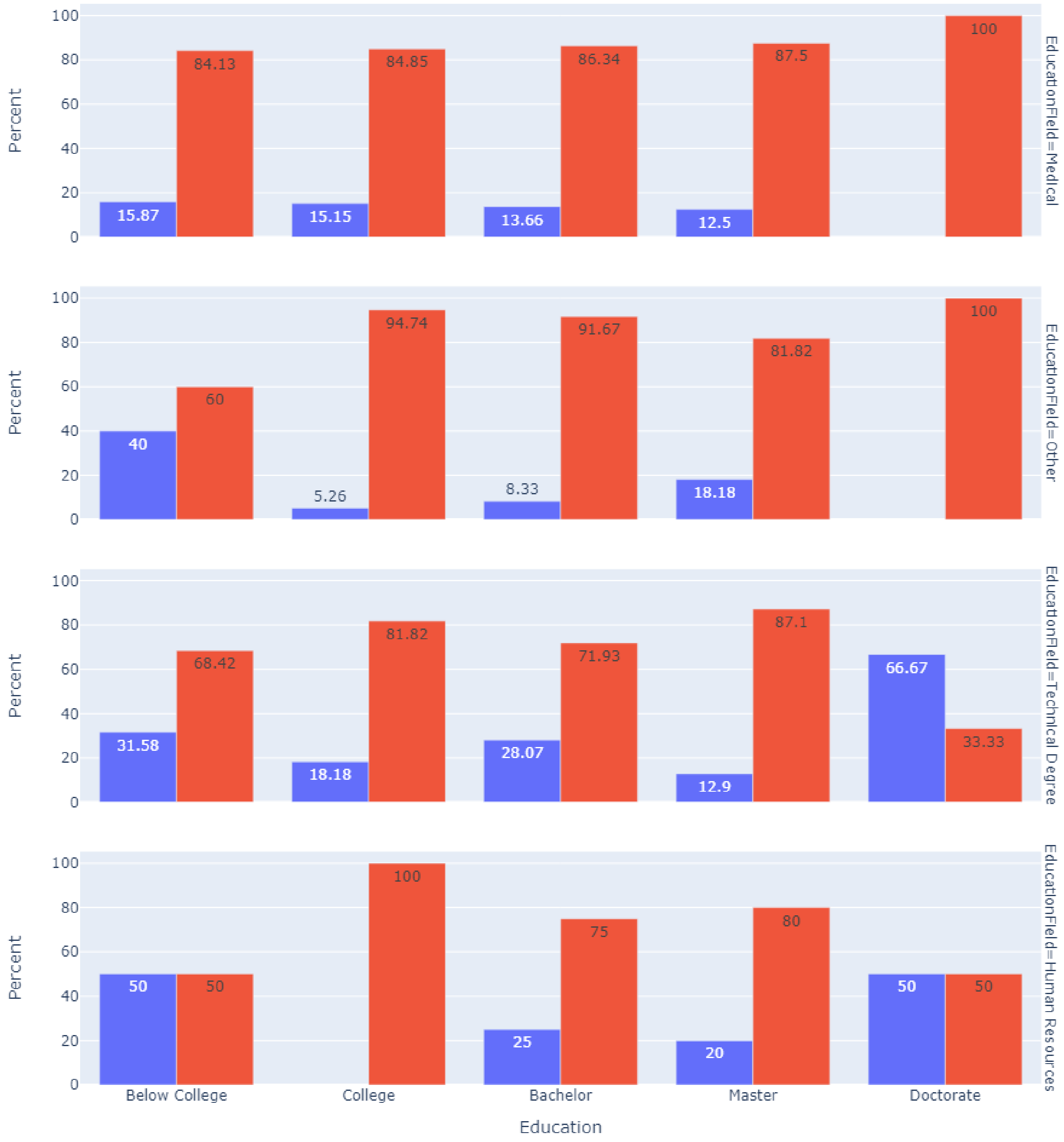
Columns Considered: 1. Education
2. Education Field
3. Job Satisfaction
4. Performance Rating

1. Education



2. Education Field





Outcome: 1. Maximum company employees have a bachelors as highest form of education. Doctorate employees have the least attrition levels below college employees have the highest attrition levels.

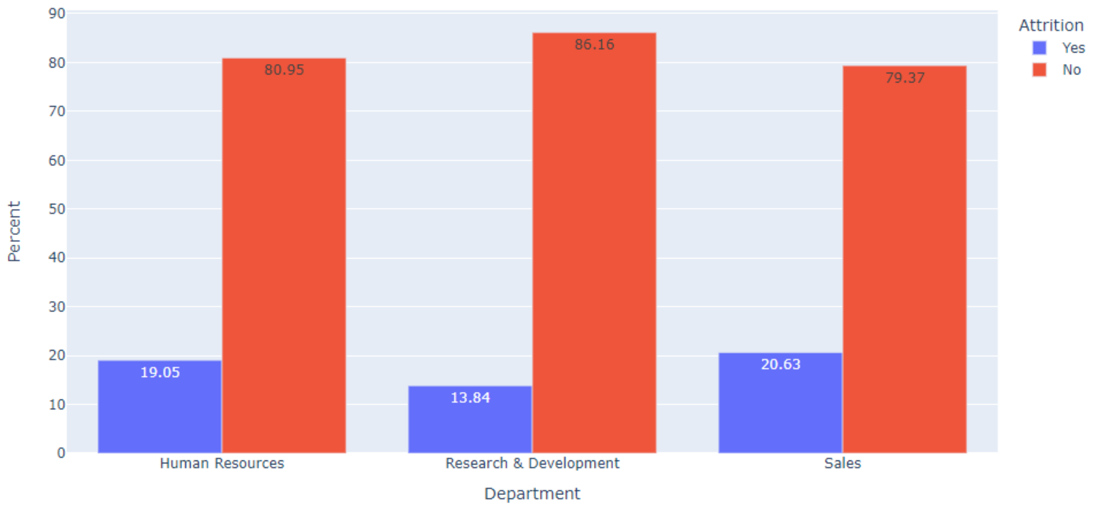
2. Employees having an education field in life sciences and medical field are maximum in number and have the least average attrition rate amongst all the employees.

3. Employees with Marketing, Technical Degree, Human Resources and other education fields consist of a lesser workforce in the company, and also have a higher average attrition rate.

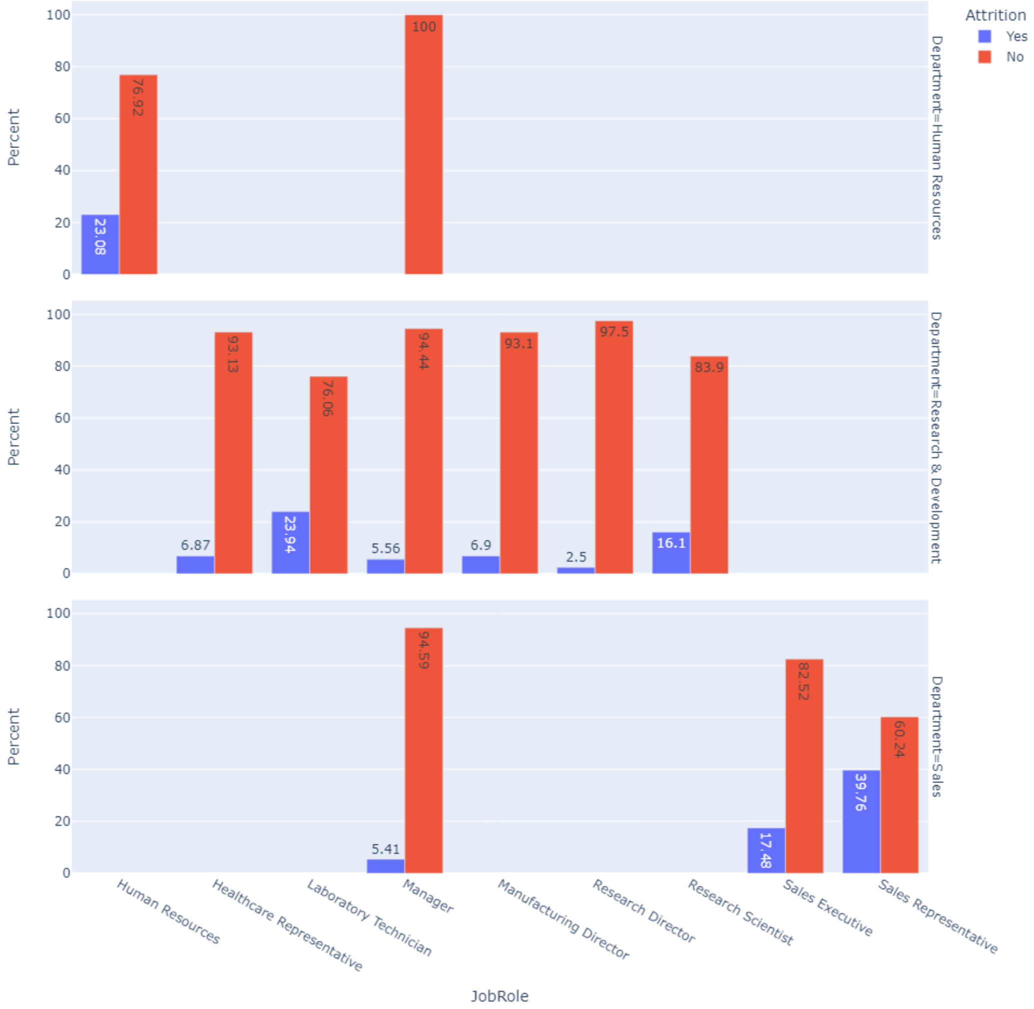
Impact of Work Role

Columns Considered: 1. Department
2. Job Role
3. Overtime

1. Department



2. Job Role

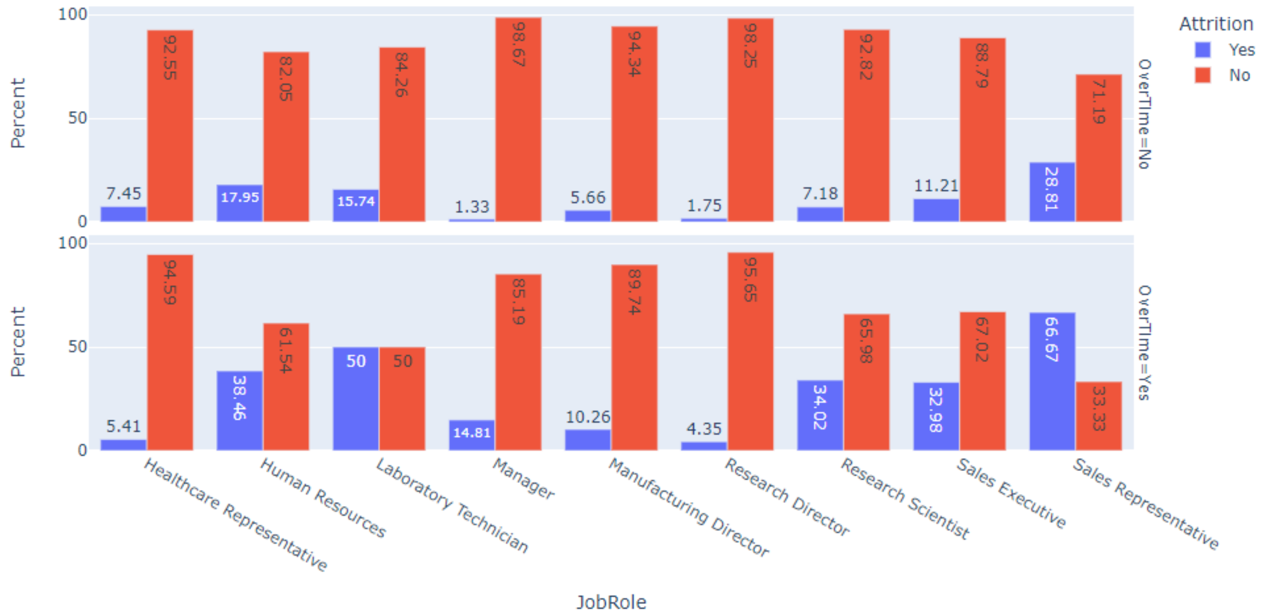


Outcome: 1. Maximum company employees have a bachelors as highest form of education. Doctorate employees have the least attrition levels below college employees have the highest attrition levels.

2. Employees having an education field in life sciences and medical field are maximum in number and have the least average attrition rate amongst all the employees.

3. Employees with Marketing, Technical Degree, Human Resources and other education fields consist of a lesser workforce in the company, and also have a higher average attrition rate.

3. Overtime



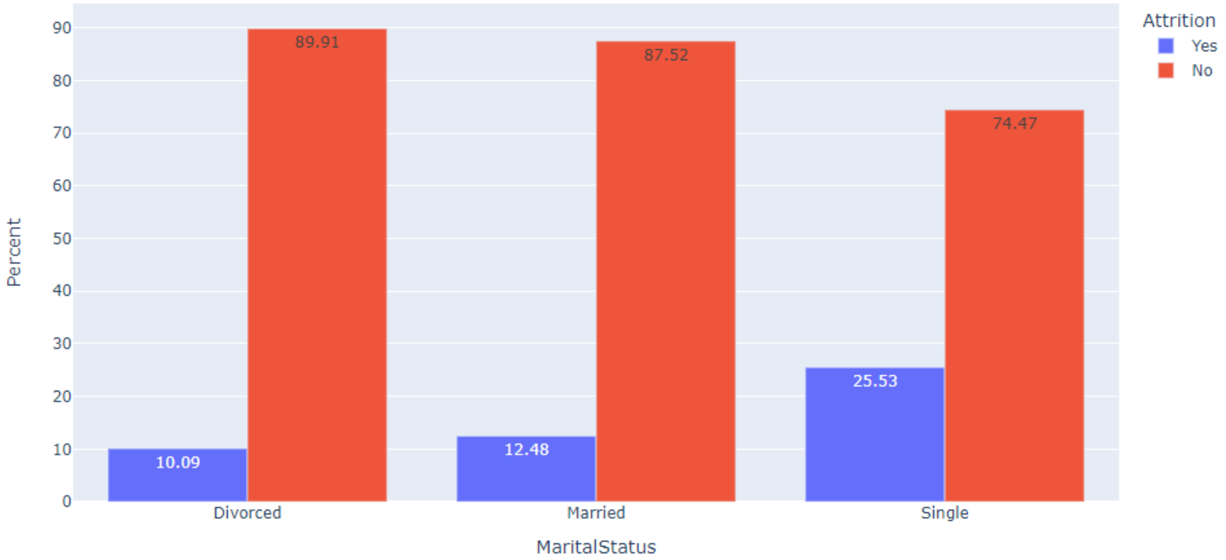
Outcome: It is a clearly visible overtime employees have a higher attrition rate.

Also, which job role has the most overtime employees.

Impact of Personal Life

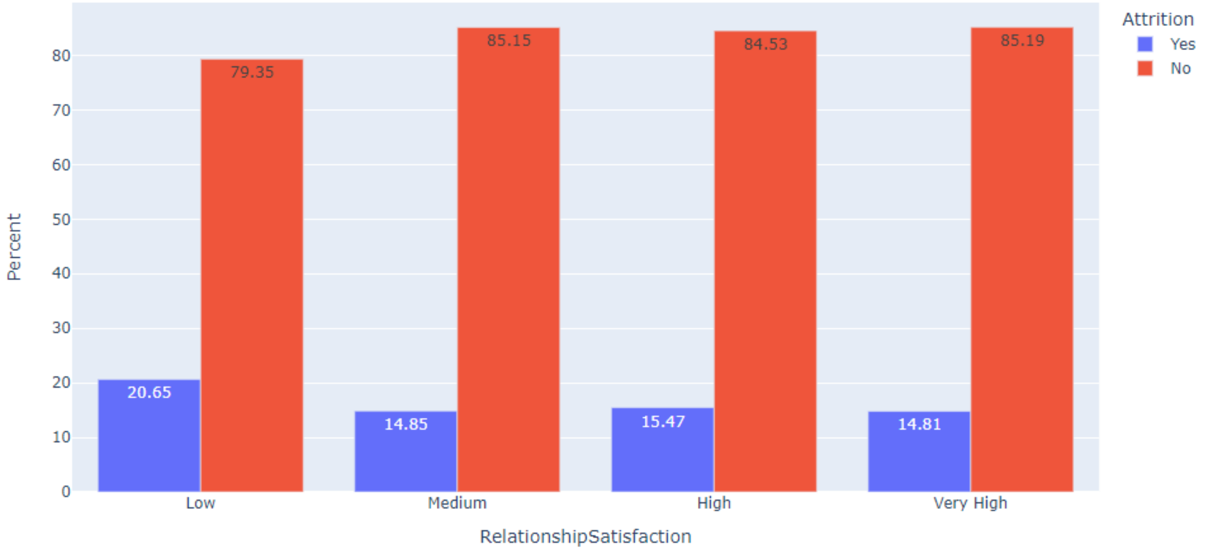
Columns Considered: 1. Marital Status
2. Relationship Saisfaction
3. Work Life Balance

1. Marital Status



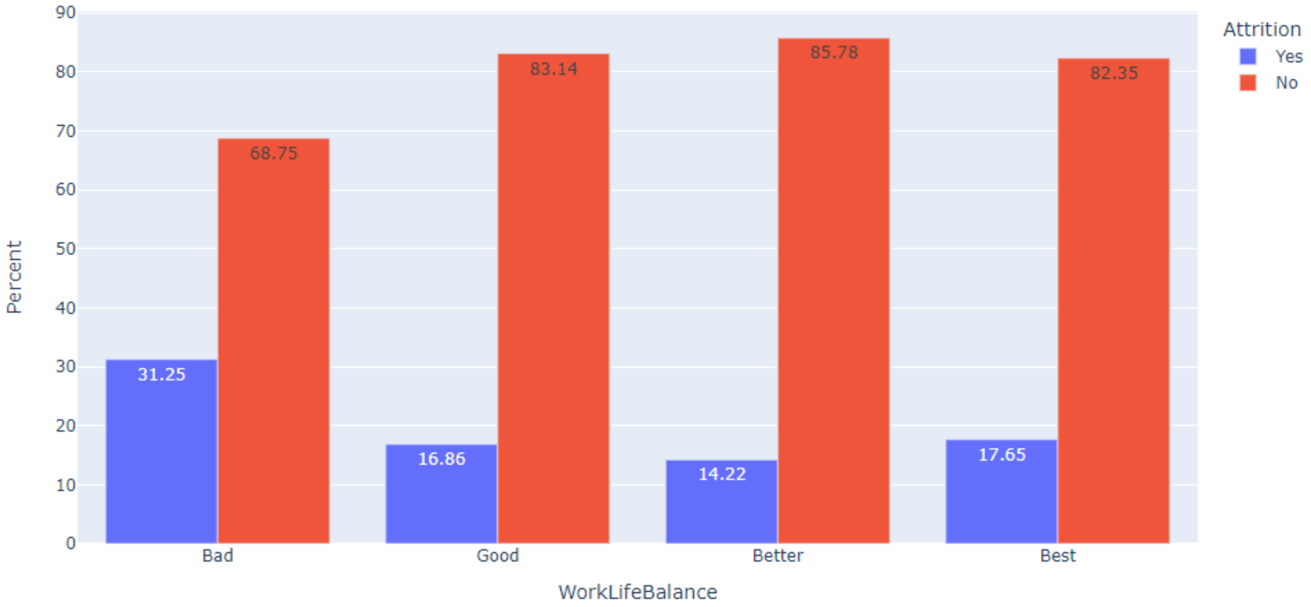
Outcome: Married employees consists more than half of the workforce. Divorced people have the least attrition rate.

2. Relationship Satisfaction



Outcome: Relationship Satisfaction is a direct correlating factor, higher the relationship satisfaction levels, lesser the attrition levels.

3. Impact of Work Life Balance



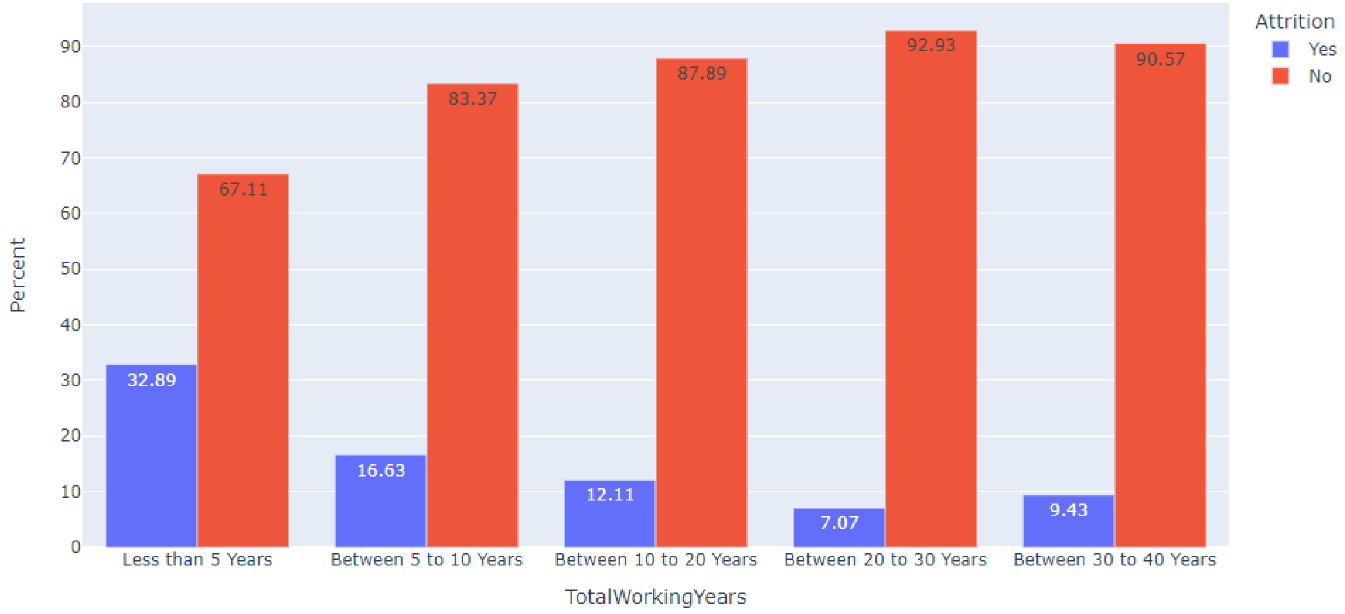
‘

Outcome: People categorizing themselves having a bad work life balance are show a high attrition levels.

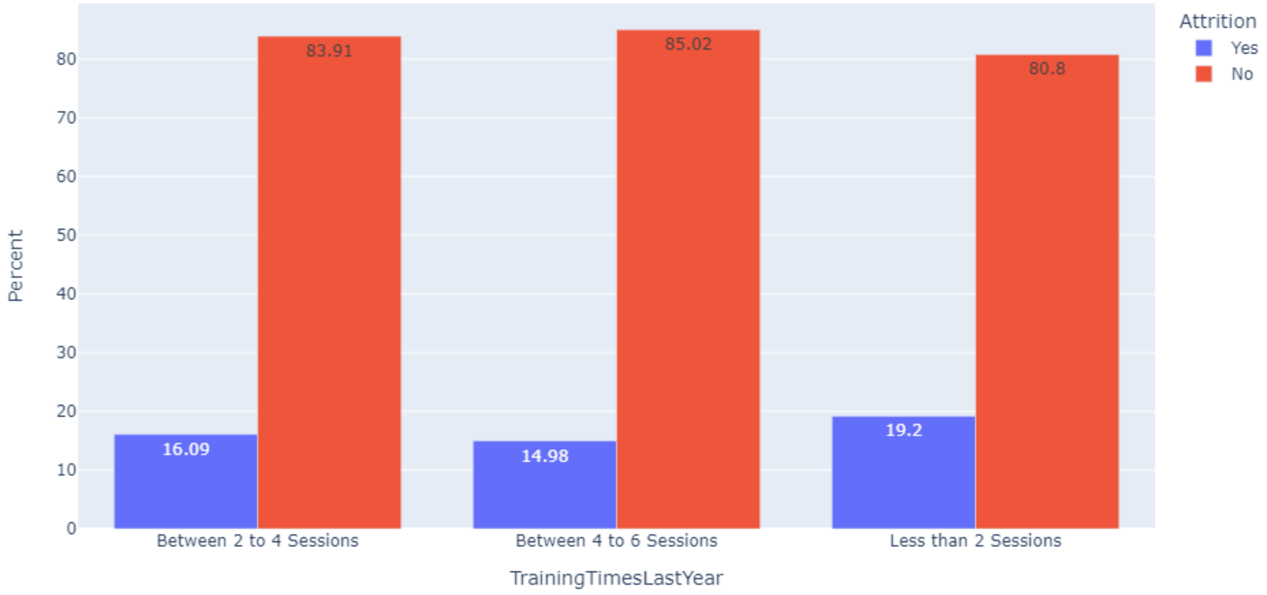
Impact of Personal Life

Columns Considered: 1. Total Working Years
2. Training Times Last Year
3. Years at Company
4. Years in Current Role
5. Years since Last Promotion
6. Years with Current Manager

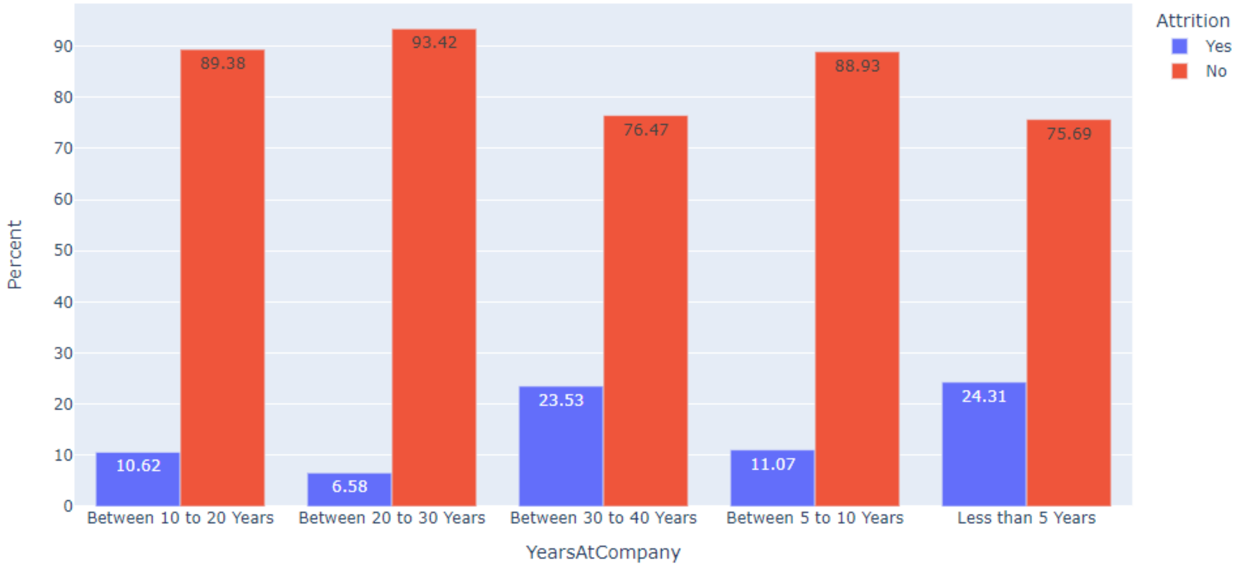
1. Total Working Years



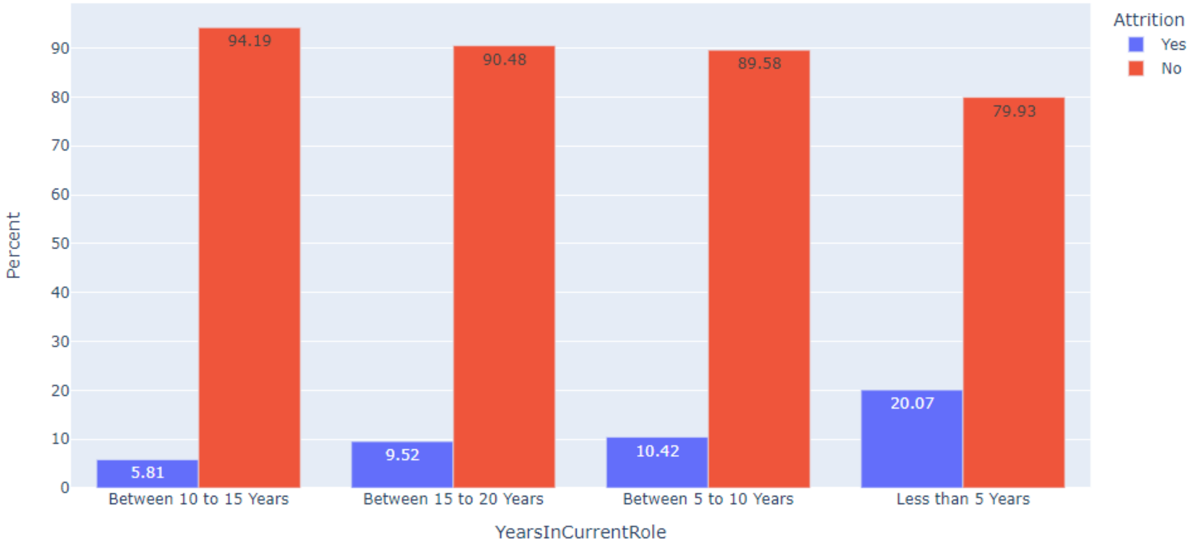
2. Training Times Last Year



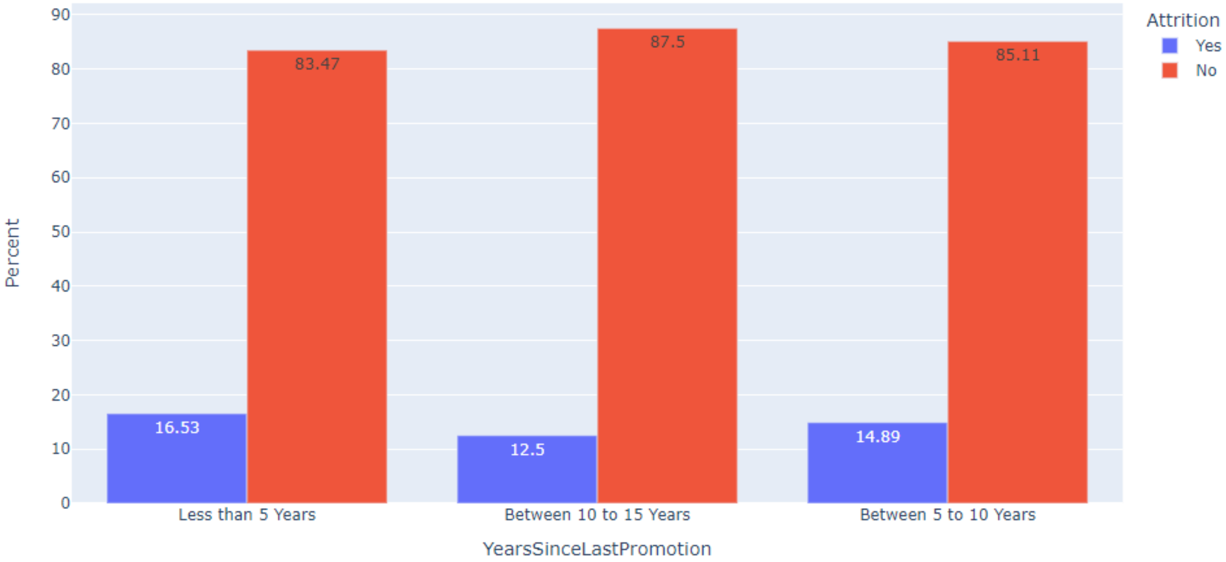
3. Years At Company



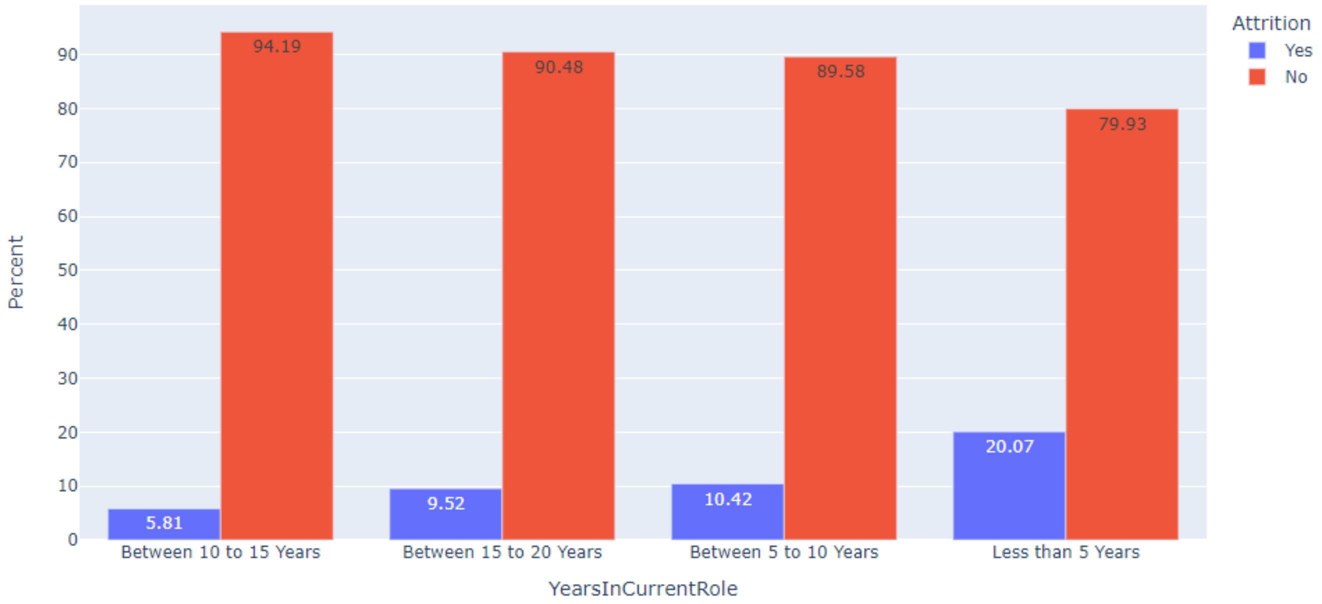
4. Years in Current Role



5. Years since Last Promotion



6. Years with Current Manager



Outcome: Employees in the having higher work experience, and in staying for more than 10 years in this company have least attrition rate.

Working under the same manager and a particular job role is also extreme factor, where if it extends more than 10 years, there is sudden increase in attrition levels.

Employees which have recently received promotion are also likely to leave the company.

Data Preprocessing:

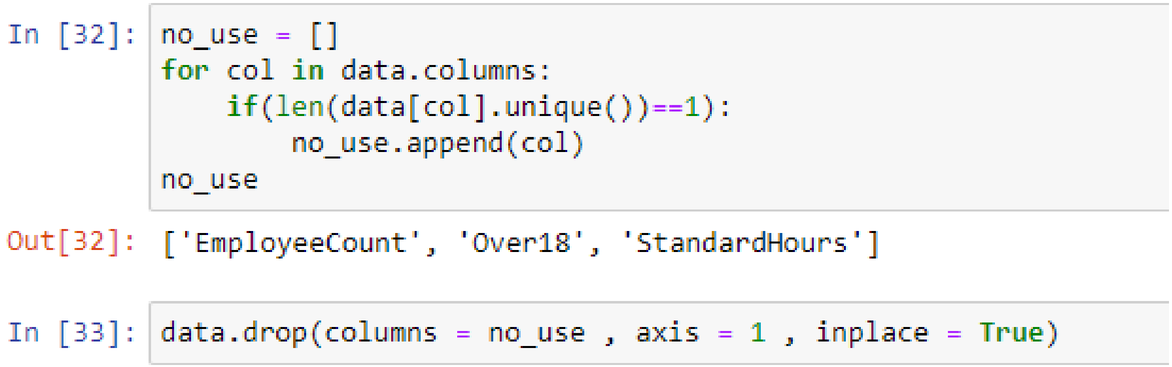
Now that we have understodd the different parameters of the dataset, we need to alter the dataset in accordance to include only important features for the prediction model.

Removing Irrelevant Features

Columns Considered: EmployeeCount, Over18, StandardHours, HourlyRate, DailyRate

Columns of employeecount, over18 and standardhours consist of no unique values, i.e., these parameters are same for all the employees, thus they wouldn't add much value to our prediction mode. Thus, removing them will be more beneifical.

The dataset consists parameters of hourly rate and daily rate, deleting them because there exists a more widely accepted parameter of Monthly Income.



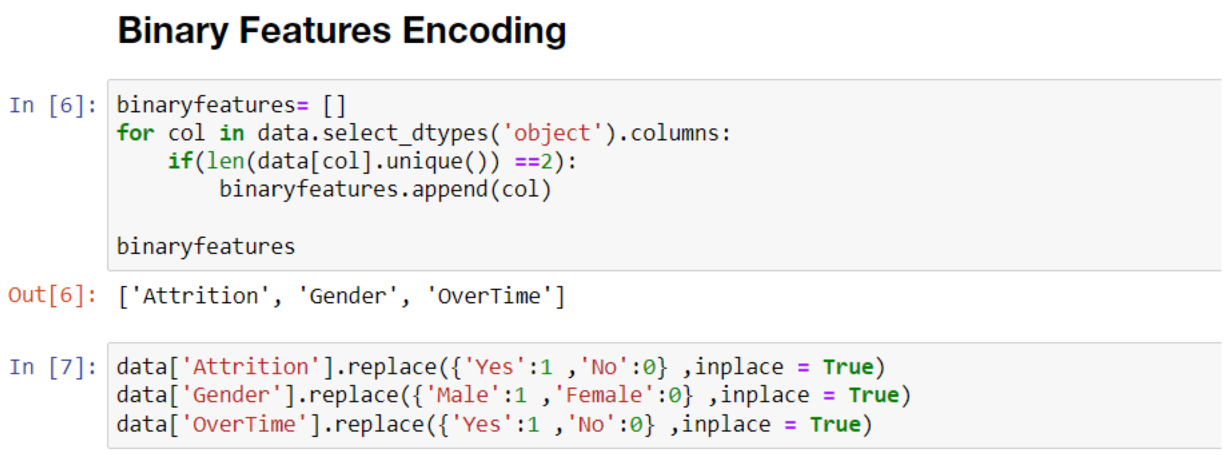
```
In [32]: no_use = []
         for col in data.columns:
             if(len(data[col].unique())==1):
                 no_use.append(col)
         no_use

Out[32]: ['EmployeeCount', 'Over18', 'StandardHours']

In [33]: data.drop(columns = no_use , axis = 1 , inplace = True)
```

Label Encoding

i. Binary Features

Columns Considered: Attrition, Gender, OverTime

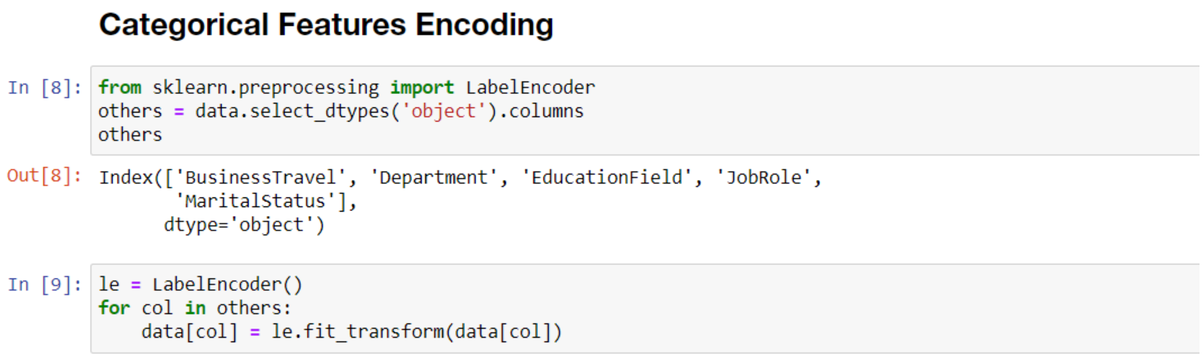
These columns have only two unique values.
i. Attrition: Yes/No
ii. Gender: Male/Female
iii. Overtime: Yes/No



Binary Features Encoding

```
In [6]: binaryfeatures= []
        for col in data.select_dtypes('object').columns:
            if(len(data[col].unique()) ==2):
                binaryfeatures.append(col)

        binaryfeatures

Out[6]: ['Attrition', 'Gender', 'OverTime']

In [7]: data['Attrition'].replace({'Yes':1 ,'No':0} ,inplace = True)
        data['Gender'].replace({'Male':1 ,'Female':0} ,inplace = True)
        data['OverTime'].replace({'Yes':1 ,'No':0} ,inplace = True)
```

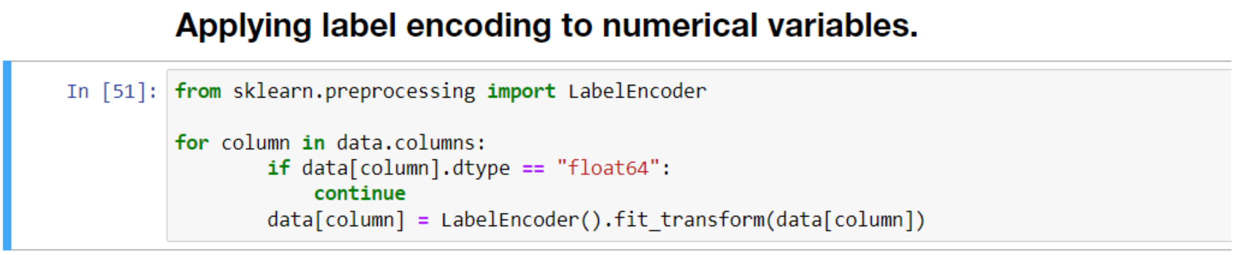
ii. Categorical Variables



Categorical Features Encoding

```
In [8]: from sklearn.preprocessing import LabelEncoder
        others = data.select_dtypes('object').columns
        others

Out[8]: Index(['BusinessTravel', 'Department', 'EducationField', 'JobRole',
               'MaritalStatus'],
              dtype='object')

In [9]: le = LabelEncoder()
        for col in others:
            data[col] = le.fit_transform(data[col])
```

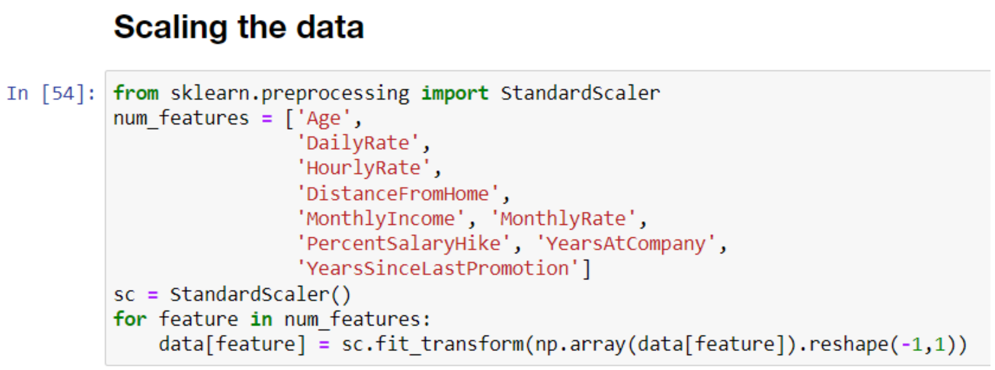
iii. Numerical Features



Applying label encoding to numerical variables.

```
In [51]: from sklearn.preprocessing import LabelEncoder

         for column in data.columns:
                 if data[column].dtype == "float64":
                     continue
                 data[column] = LabelEncoder().fit_transform(data[column])
```

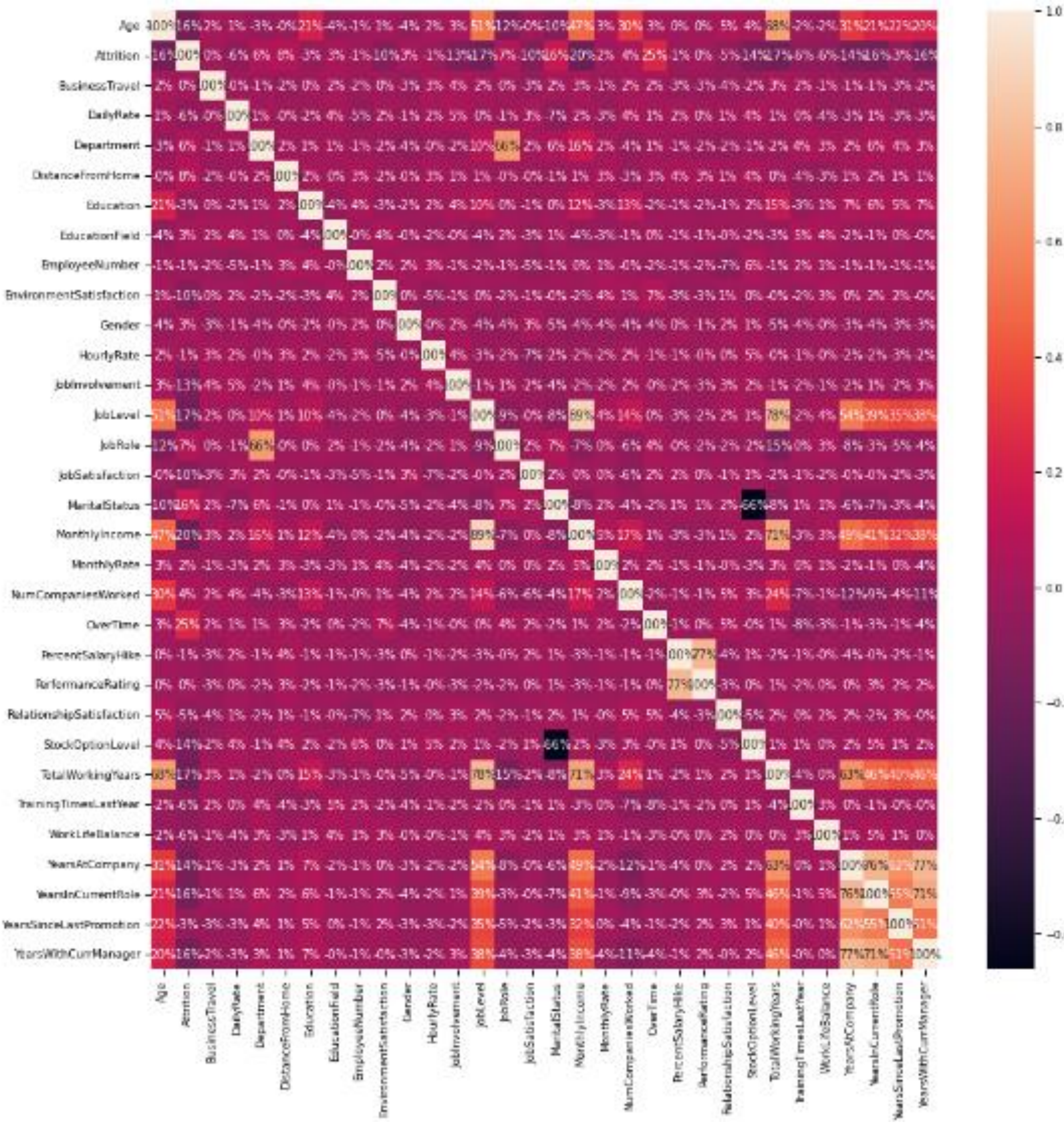
Scaling the Numerical Data



Scaling the data

```
In [54]: from sklearn.preprocessing import StandardScaler
         num_features = ['Age',
                         'DailyRate',
                         'HourlyRate',
                         'DistanceFromHome',
                         'MonthlyIncome', 'MonthlyRate',
                         'PercentSalaryHike', 'YearsAtCompany',
                         'YearsSinceLastPromotion']
         sc = StandardScaler()
         for feature in num_features:
             data[feature] = sc.fit_transform(np.array(data[feature]).reshape(-1,1))
```

Plotting the Correlation Matrix



Dropping the strongly correlated columns.

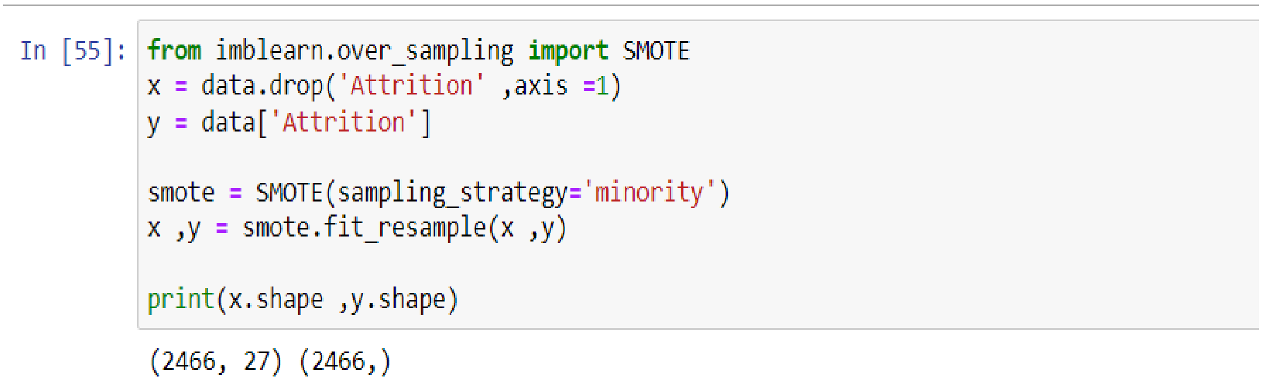
i. 'YearsInCurrentRole' strongly correlated to 'YearsAtCompany'
ii. 'JobLevel' strongly correlated to 'MonthlyIncome'
iii. 'JobLevel' strongly correlated to 'TotalWorkingYears'
iv. 'MonthlyIncome' strongly correlated to 'TotalWorkingYears'
v. 'YearsAtCompany' strongly correlated to 'YearsWithCurrManager'
vi. 'YearsAtCompany' strongly correlated to 'YearsInCurrRole'
vii. 'YearsInCurrRole' strongly correlated to 'YearsWithCurrManager'

Creating the Predictive Model

For creating the predictive model, we would be considering various Sklearn library algorithms to find the best for our dataset.

Balancing the Dataset

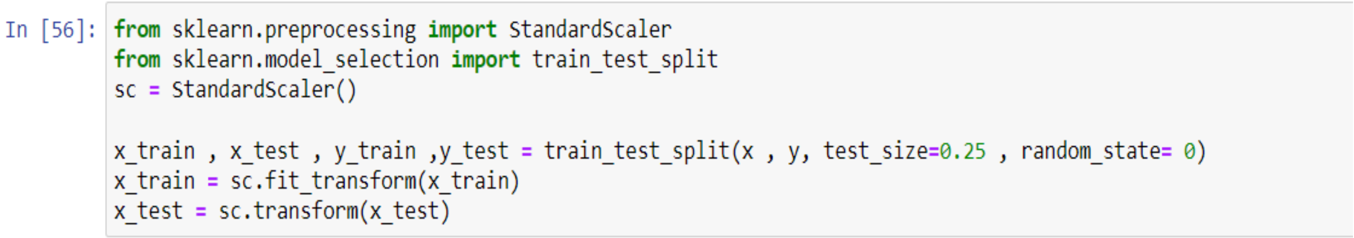
Initially to create a predictive model, the dataset has to balanced. Under the column attrition, the dataset consists 84% employees are under 'yes' and 16% are under 'no', which create a biased model.



```
In [55]: from imblearn.over_sampling import SMOTE
         x = data.drop('Attrition' ,axis =1)
         y = data['Attrition']

         smote = SMOTE(sampling_strategy='minority')
         x ,y = smote.fit_resample(x ,y)

         print(x.shape ,y.shape)

         (2466, 27) (2466,)
```

Creating a Train-Test Split Dataset

The dataset has to be divided into a train test sample, where 75% of the data is used to to create and verify the algorithm, and the rest 25% of the data is used to check the accuracy of the created model.



```
In [56]: from sklearn.preprocessing import StandardScaler
         from sklearn.model_selection import train_test_split
         sc = StandardScaler()

         x_train , x_test , y_train ,y_test = train_test_split(x , y, test_size=0.25 , random_state= 0)
         x_train = sc.fit_transform(x_train)
         x_test = sc.transform(x_test)
```

Deciding the Algorithm

Now as the train and test dataset is ready, it is time to select the algorithm suitable for the dataset. The following algorithms can be considered for the dataset:

1. Logistic Regression

2. KNeigghbors Classifiers

3. Linear Discriminant Analysis

4. Support Vector Machines (SVMs)

5. Decision Tree Classifier

6. AdaBoost Classifier

7. Random Forest Classifier

8. Gaussian Naive Bayes

The algorithm will be selected with high precision, accuracy, f1 score and least negative mean square error.

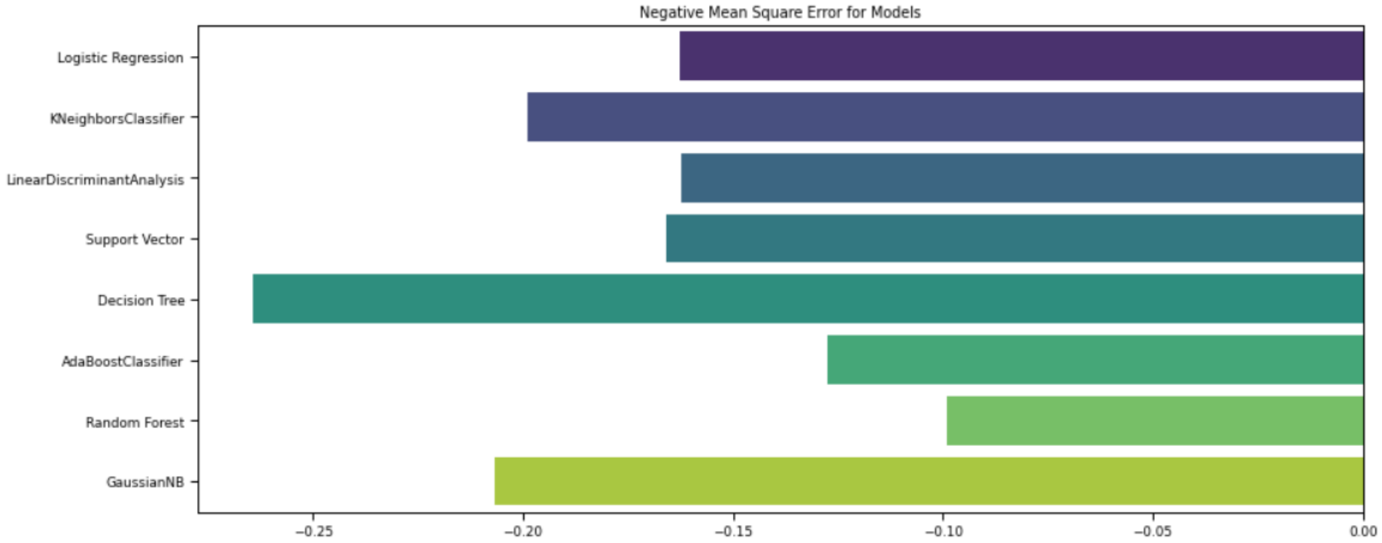
From the classification report of all the algorithms and the graph of negative mean square errors, Random Forest Algorithm comes out to have the highest accuracy, precision, f1 score and the least negative mean square error.

Classification Report:

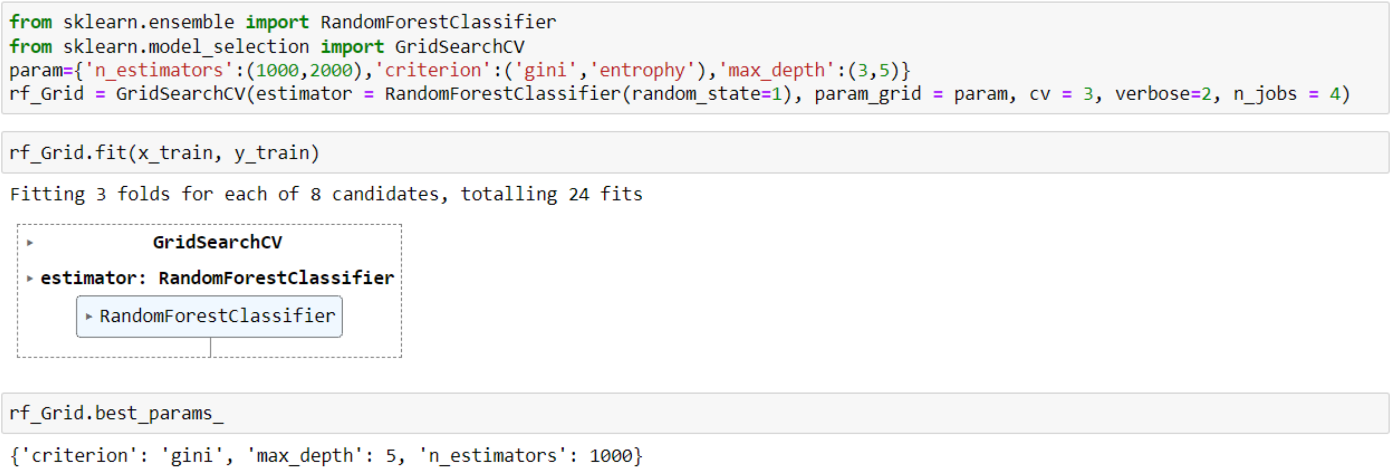
	Precision	**Recall**	**f1-score**	**Support**
0	0.88	0.93	0.90	934
1	0.93	0.87	0.90	915
Accuracy			0.90	1849
Macro Avg	0.90	0.90	0.90	1849
Weighted Avg	0.90	0.90	0.90	1849

All the algorithms have been tested on through K-Folds Cross Validation with sample size of 10 data sets.

Negative Mean Square Error Graph:



Applying GridCV Search on Random Forest Classifier



```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
param={'n_estimators':(1000,2000),'criterion':('gini','entrophy'),'max_depth':(3,5)}
rf_Grid = GridSearchCV(estimator = RandomForestClassifier(random_state=1), param_grid = param, cv = 3, verbose=2, n_jobs = 4)
```

```
rf_Grid.fit(x_train, y_train)
```

```
Fitting 3 folds for each of 8 candidates, totalling 24 fits
```

```
▸        GridSearchCV
▸ estimator: RandomForestClassifier
      ▸ RandomForestClassifier
```

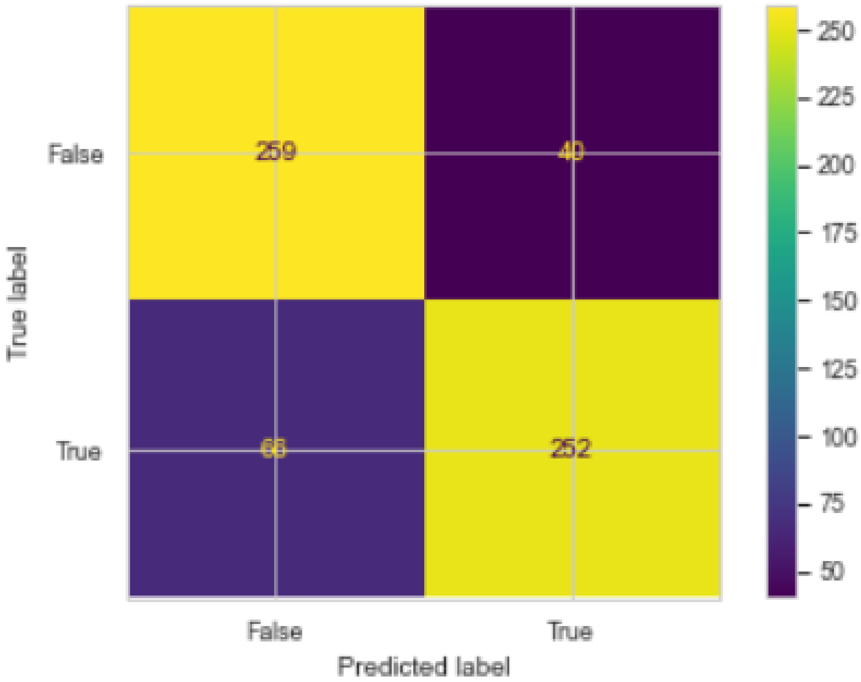
```
rf_Grid.best_params_
```

```
{'criterion': 'gini', 'max_depth': 5, 'n_estimators': 1000}
```

Train Accuracy: 88.91%

Test Accuracy: 83.14%

Plotting the Confusion Matrix



True Positives: 252

False Positives: 40

True Negatives: 259

False Negatives: 66

Conclusions:

On observing results of using Random Forest Classifier, applying GridCV Search on the same, studying the classification report and plotting the confusion matrix, the following conclusions can be derived:

i. The major reason of choosing Random Forest Classifier is all among the of the best f1 score and recall.

ii. Recall — What percent of the positive cases did you catch?

iii. Recall is the ability of a classifier to find all positive instances. For each class it is defined as the ratio of true positives to the sum of true positives and false negatives.

iv. Recall:- Fraction of positives that were correctly identified.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives+False Negatives}}$$

i. F1 score — What percent of positive predictions were correct?

ii. The F1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. F1 scores are lower than accuracy measures as they embed precision and recall into their computation.

iii. As a rule of thumb, the weighted average of F1 should be used to compare classifier models, not global accuracy.

$$\text{F1 Score} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall+Precision}}$$

Conclusions

i. Working working with Random Forest Hyper Tuning Parameter HyperTunning with both GridCV Search and RandomisedSearchCV, to integrate more parameters of random forest like bootstrap, max_depth, max_features, min_samples_leaf and min_samples_split.

ii. Including feature selection in the selected algorithm Feature selection using Random forest comes under the category of Embedded methods. Embedded methods combine the qualities of filter and wrapper methods. They are implemented by algorithms that have their own built-in feature selection methods.

iii. Integrating various research paper's on feature selection and embedded methods and working on employee retention strategies.

Referred Research Papers

A Study on Employee Attrition Effects and Causes

https://drive.google.com/file/d/17aRmygDKGNrXzTUiiXtFITKqErJMJO6a/view?usp=sharing

A Study on Reasons of Attrition and Strategies for Retention

https://drive.google.com/file/d/1LuNEiOZnJrRRHrkYnrPtDi_mRLKTP3ul/view?usp=sharing

Attrition Issues and Retention Challenges of Employees

https://drive.google.com/file/d/1be5ZKiISUvIvh2VDStD_or8VLcdInpkr/view?usp=sharing

All these research papers are publications in working with categorizing different important factors responsible for retention strategies.

Considering certain outcomes from this data will be beneficial in feature selection and also building various own features. Also, determining features importance to deliver maximum recall and f1 score of the algorithm.