EMPLOYEE PERFORMANCE ANALYSIS

Work done by

JOHNSON M	**(Student)**
SHAKTI KUMAR	**(Student)**
SAIDEEPAK R	**(Student)**
YOGESHWARAN MS	**(Student)**
GOURAV AWASTHI	**(Mentor)**

Report submitted in partial fulfillment of the requirements for the course

POST GRADUATE PROGRAM IN DATA SCIENCE AND ENGINEERING

APRIL 2019
CHENNAI

Table of Contents

CHAPTER 1

INTRODUCTION

1.1. Employee performance analysis

The Employee dataset includes features like employee_id, department, region, education, gender, recruitment_channel, no_of_trainings, age etc. In this project we will build a binary classifier that helps us predict what kind of employee will more likely to be promoted with some attributes and the dataset contains information of 54808 employees.

1.2 Need for the project

The performance of various employees in an organization varies and so is the probability of each employee getting promoted. And not getting promoted could have a direct bearing against employee attrition and hence the HR department would like to know the probability that an employee will get promoted. Such would help an organization to predict performance of an employee.

1.3 Objective

The objective of this project is to predict whether an employee will get promoted or not by using predictive model and also understand the factors which impact more on an employee promotion.

1.4 Features in dataset

The following variables are considered for the analysis:

- **employee_id:** Unique employee ID for each employees, there are total of 54808 employees in this data. It is categorical variable.
- **department:** Department in which the employee works, **9 unique departments are present in this dataset.** It is categorical variable.
- **region:** Different regions the employees are working, there are total of 34 unique regions are present in the dataset. It is categorical variable.
- **education:** Education level of the each employee are given by this variable, there are 4 different education levels are mentioned in this dataset. It is categorical variable.
- **gender:** Gender of the employee are given in this variable, male and female are two different categories in this data. It is categorical variable.

- **recruitment_channel:** Channel through which employee was recruited are given through this variable there are total of 3 unique recruitment channels are given in this dataset. It is categorical variable.
- **no_of_trainings:** No of training programs the employee has undergone in this company from joining. The number of training variable range from 0-10 in this dataset. It is categorical variable.
- **age:** Age of the each employee is mentioned in this dataset. It is a continuous variable.
- **previous_year_rating:** Performance rating of the employee in the previous year given by the company to each employee's. The rating ranges from 0-5 in this dataset. It is categorical variable.
- **length_of_service:** Length of each employee's have been working in this company. It is a continuous variable.
- **KPIs_met >80%:** This variable shows whether the employee met more than 80% of the KPIs or not. If met the value is 1, else 0. It is categorical variable.
- **awards_won:** This variable gives the information whether the employee has won any awards or not. If yes the value is 1, else 0. It is categorical variable.
- **avg_training_score:** Average score scored by the employee in the training process took in the company. It is continuous variable.
- **is_promoted:** It is variable that we are going to predict. If the value is 1, then the employee is promoted, if 0 the employee is not promoted.

CHAPTER 2

DATA CLEANING

2.1 Data set chosen

Generally, there are two types of datasets are available for analysis one is case centred data and justice centred data. The dataset chosen for the analysis is case centred data. Case Centred data provides case level information; i.e., each row in the database corresponds to a dispute. These data do not contain specific justice vote information.

2.2 Missing value treatment

The dataset which is collected by us, doesn't contain any missing values ,so we don't have any need to treat the missing values.

2.3 Outlier treatments

Out of 14 variables we have, only 3 variables are continuous. They are age of the employee's, average training score, length of service. In these variables, average training score doesn't contain any outliers. Only age and length of service contains outliers. Still now we have not done anything for outlier treatment. We plan to form clusters and remove the outliers based on that on future time of projects.

CHAPTER 3

EXPLORATORY DATA ANALYSIS

3.1 Introduction

EDA is a general approach to exploring datasets by means of simple summary statistics and graphic visualizations in order to gain a deeper understanding of the data.

3.2 EDA plots

The problem is of binary classification in which class ‘0’ represents the employee is not promoted and the class ‘1’ represents the employee is not promoted. The percentage of both classes present in the data is shown in the figure 1.

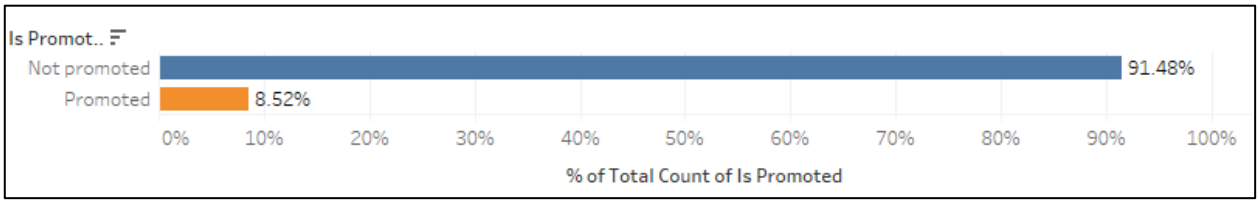

Figure 1: Bar plot of class imbalance

The percentage of both class ‘0’ and class ‘1’ is shown in the figure 1. The percentage of class ‘0’ is 91.48 and the percentage of class ‘1’ is 8.52. Since the percentage of class ‘1’ is very much less than class ‘0’ the problem is treated as imbalance classification problem.

3.2.1. Exploration of features

There are 13 input features and 1 output feature in the dataset. The data types of 13 features are shown in table 1.

Table 1: Data types of variables

Variable name	**Variable type**
employee_id	Categorical
department	Categorical
region	Categorical
education	Categorical
gender	Categorical
recruitment_channel	Categorical
no_of_trainings	Categorical

age	Continuous
previous_year_rating	Categorical
length_of_service	Continuous
KPIs_met >80%	Categorical
awards_won	Categorical
Avg_training_score	Continuous

3.2.2. Department

The feature 'Department' is a categorical variable with 9 unique categories in it. The percentage contribution of each department to the class '0' and class '1' is shown in the figure 2.

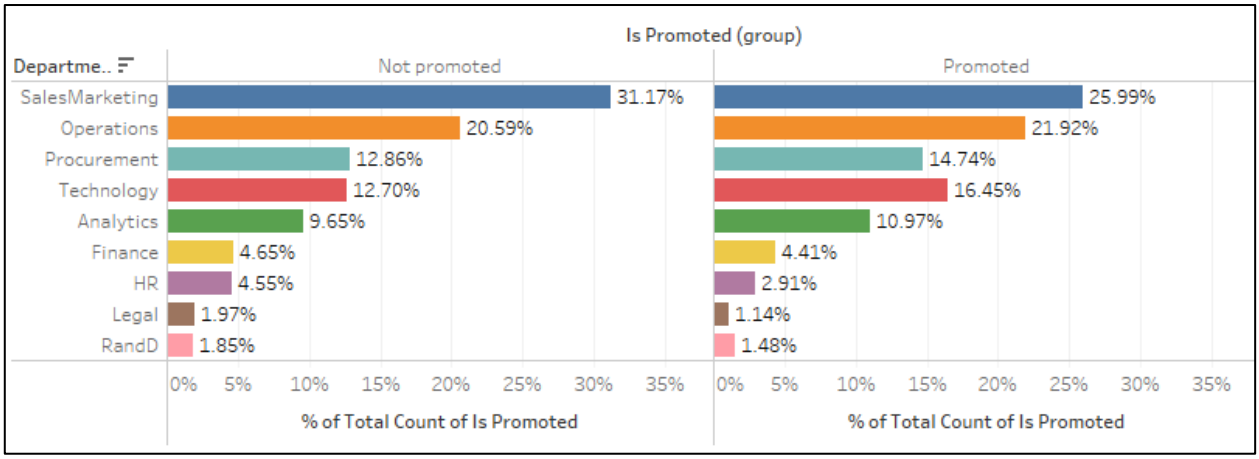

Figure 2: Bar plot of Department

There are 9 different departments in the company. Among all departments Sales/Marketing department contributes 25.99% in employees who got promotion. It is followed by Operations, Technology, Procurement, Analytics, HR, R and D and Legal department.

3.2.3. Region

The feature 'region' represents the region of employees and the bar plot is shown in figure 3. More employees got promotion from region 2, region 22 and region 7.

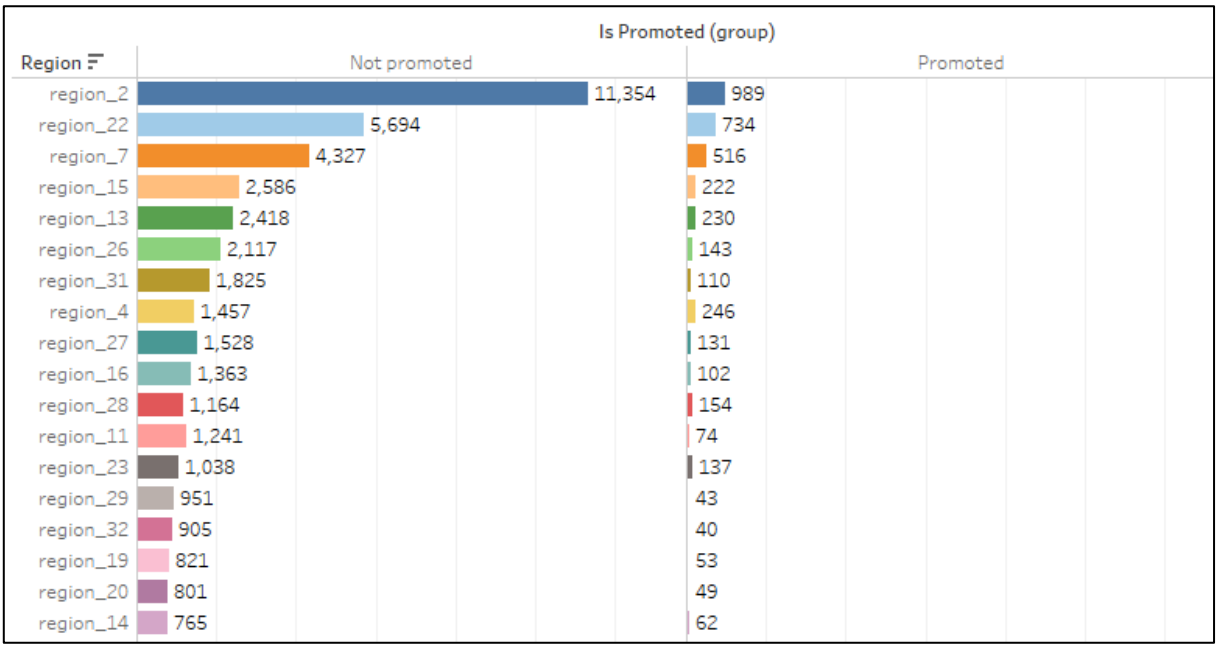

Figure 3: Bar plot of Region

3.2.4 Education

The percentage contribution of employee education in promotion is shown in figure 4.

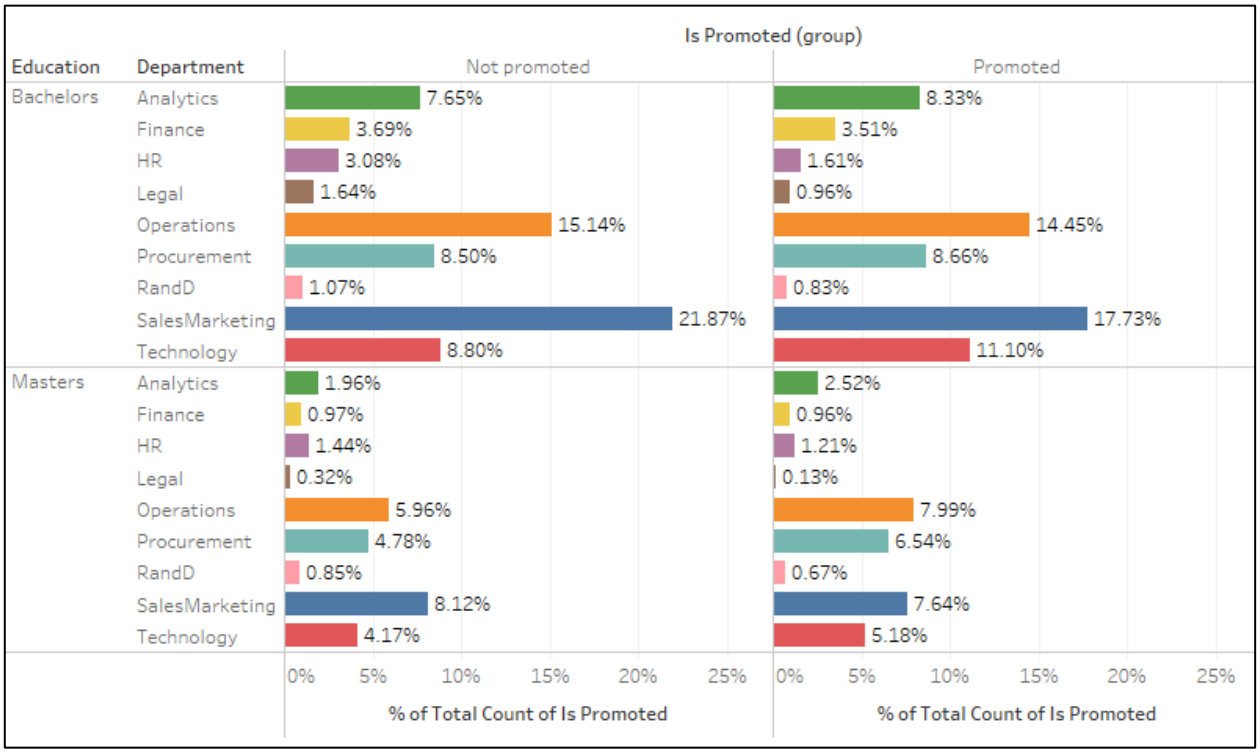

Figure 4: Bar plot of Education

Employees who did Bachelor got more promotion than employees with Master's degree.

3.2.5. Gender

The percentage contribution of gender in getting promotion is shown in figure 5.

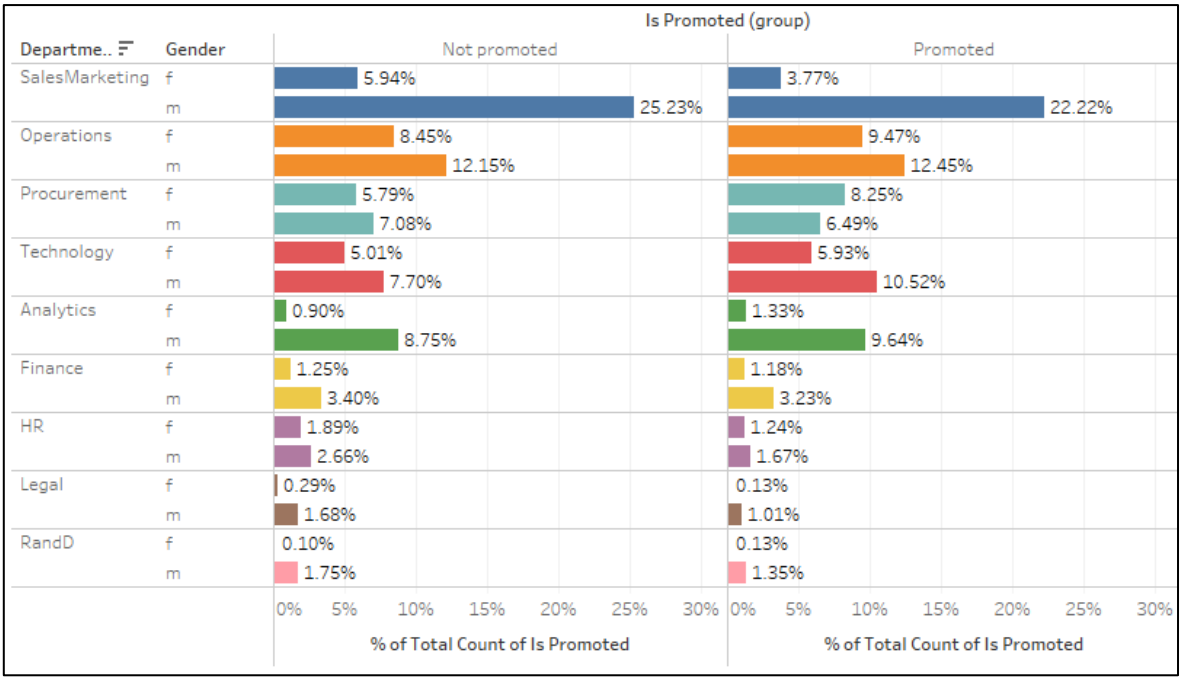

Figure 5: Bar plot of gender

3.2.6. Recruitment channel

The feature 'Recruitment_channel' represents the mode of recruitment of the employees. The percentage contribution of employees with respect to mode of recruitment is shown in figure 6.

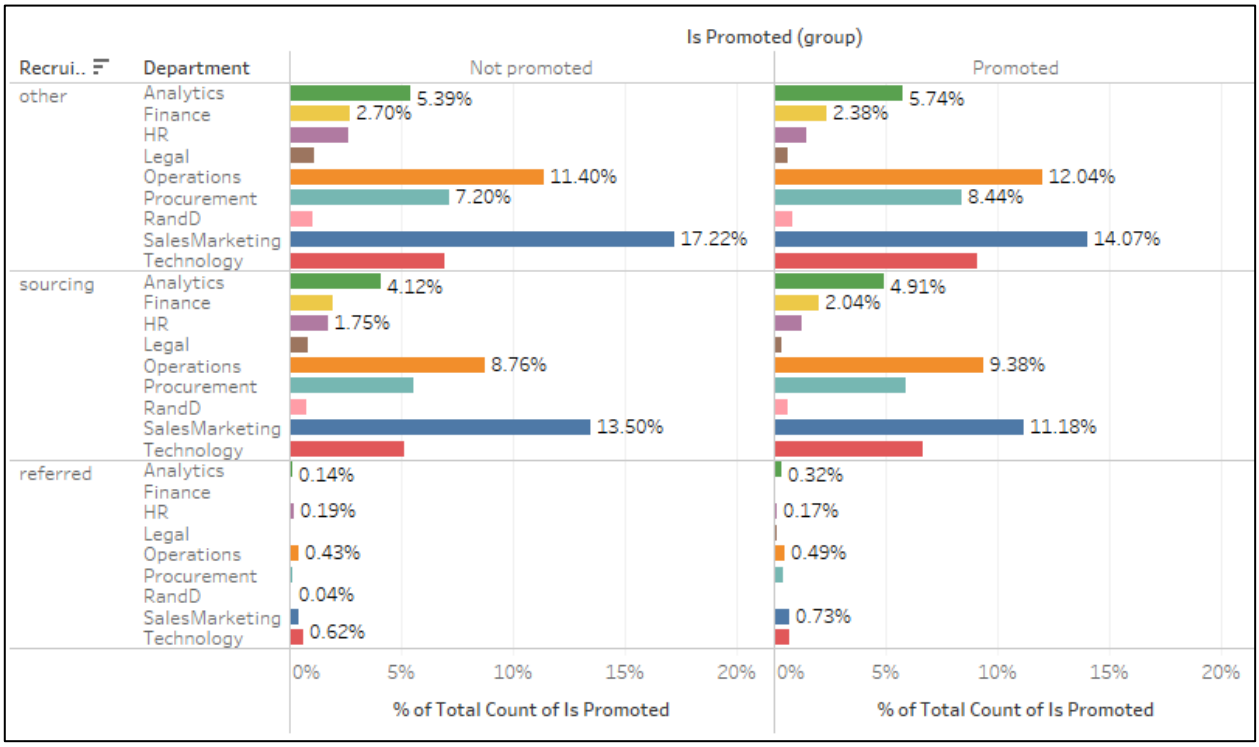

Figure 6: Bar plot of recruitment channel

Very less number of employees got promotion from the referral mode of recruitment.

3.2.7. Age

The feature ‘age’ represents the age of the employees and it is a continuous variable in this dataset. The distribution of age of employees is shown in figure 7.

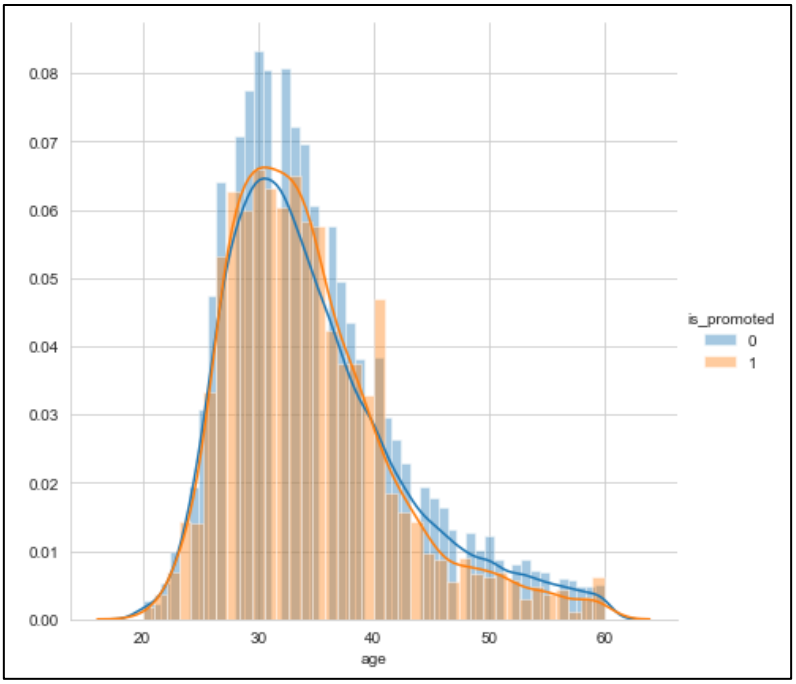

Figure 7: Histogram of age

Histogram of class ‘0’ and class ‘1’ overlaps to greater extent which is very difficult to separate from one another.

3.2.8. Previous year rating

The feature ‘previous_year_rating’ represents the rating of the employees for the previous year and it is a categorical variable. The contribution of employees in promotion with respect to previous year rating is shown in figure 8. Employees with rating 5 and 3 got more promotion compared to ratings 1, 2 and 4.

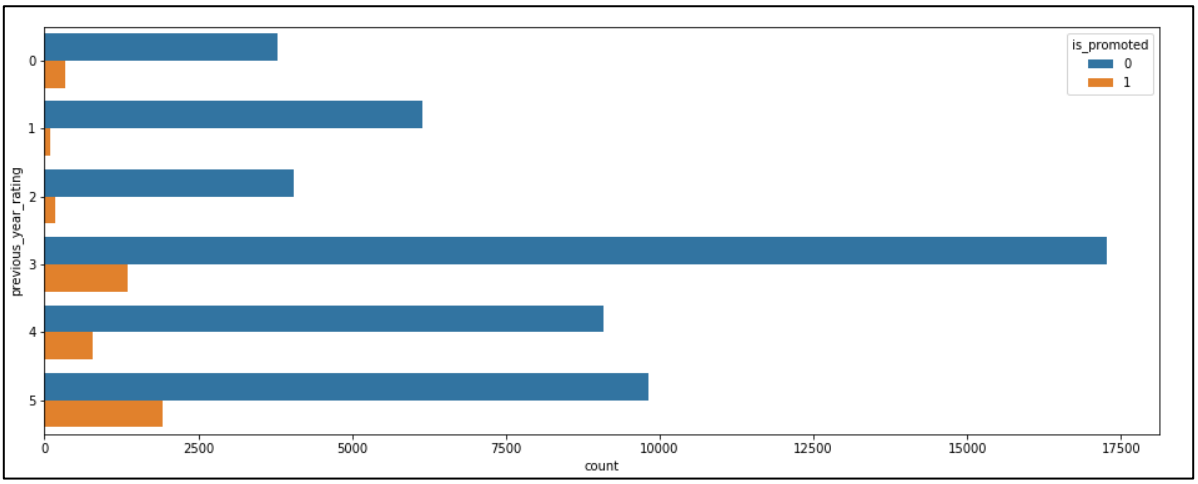

Figure 8: Bar plot of previous year rating

3.2.9. KPI's met>80%

The feature 'KPI's met>80%' key performance indication of employees. It is a categorical variable with class '0' represents KPI's not met and class '1' represents KPI's met. The percentage contribution of employees with respect to KPI is shown in figure 9.

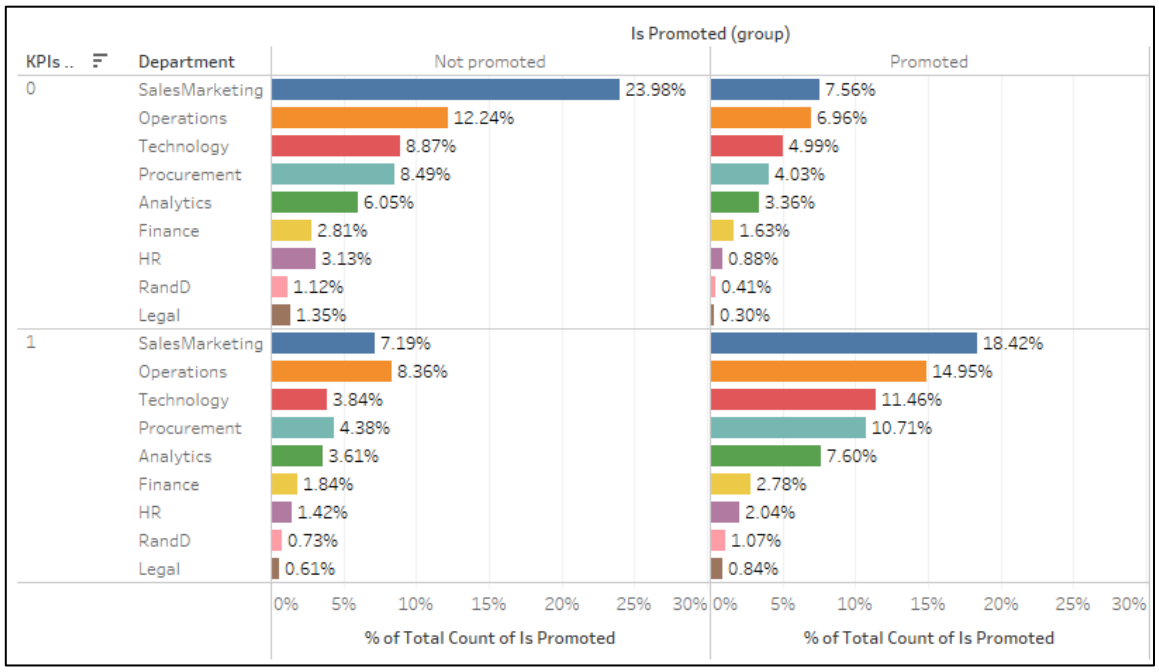

Figure 9: Bar plot of KPI>80%

3.2.10. Awards won

The feature 'awards_won' is a categorical variable in which class '0' represents the employees did not won any award and class '1' represents employee has won award. The percentage contribution of awards in promotion of the employee is represented in figure 10. More people from awards not won got promoted compared to employees who won award.

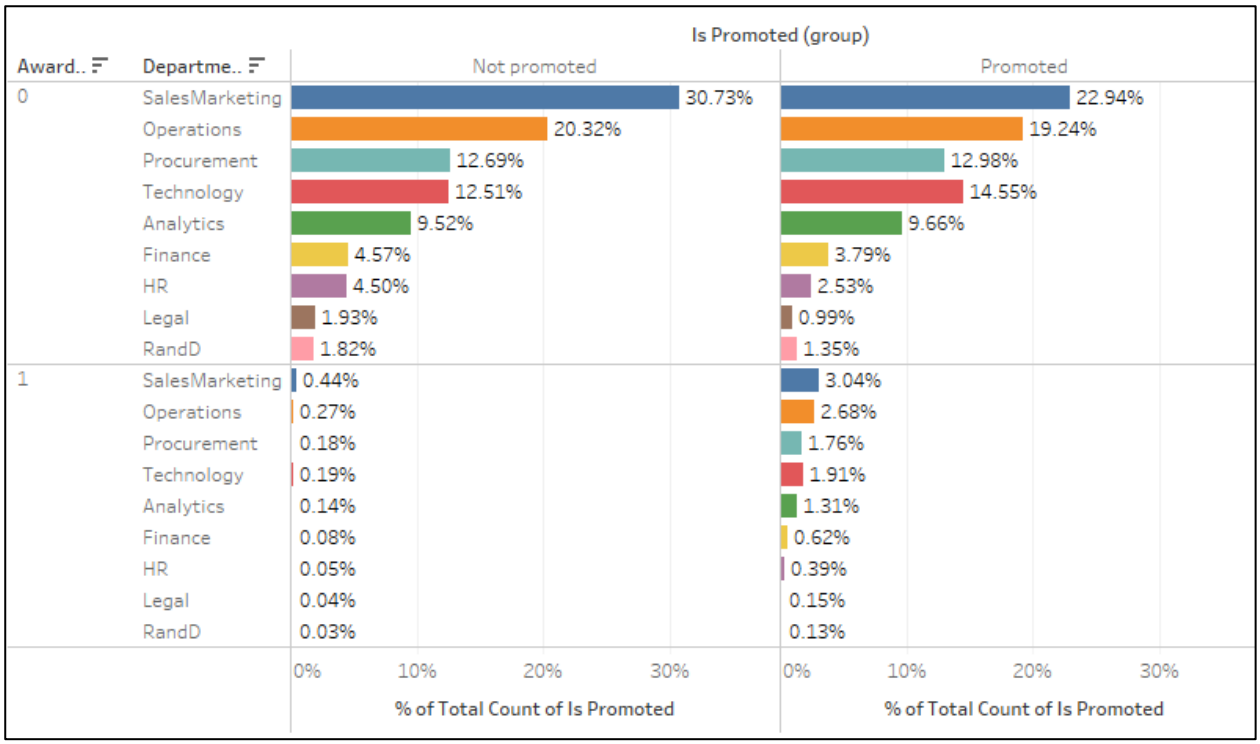

Figure 10: Bar plot of Awards won

3.2.11. Average training score

The feature 'average_training_score' represent the training score obtained by the employees and it is a continuous variable. Histogram of average training score is shown in figure 11. The distribution of both the classes overlap to greater extent and it is tedious to separate.

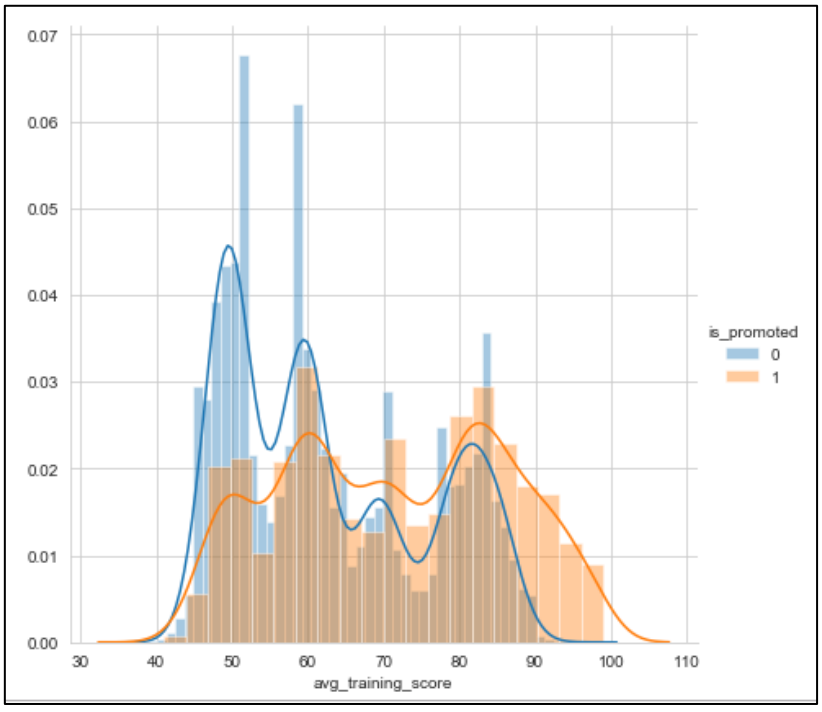

Figure 11: Histogram of Average training score

3.2.12. Length of service

The feature ‘length_of_service’ represent the years the employees are working in that company. Histogram of length of service is shown in figure 12. The distribution is right skewed for both the classes.

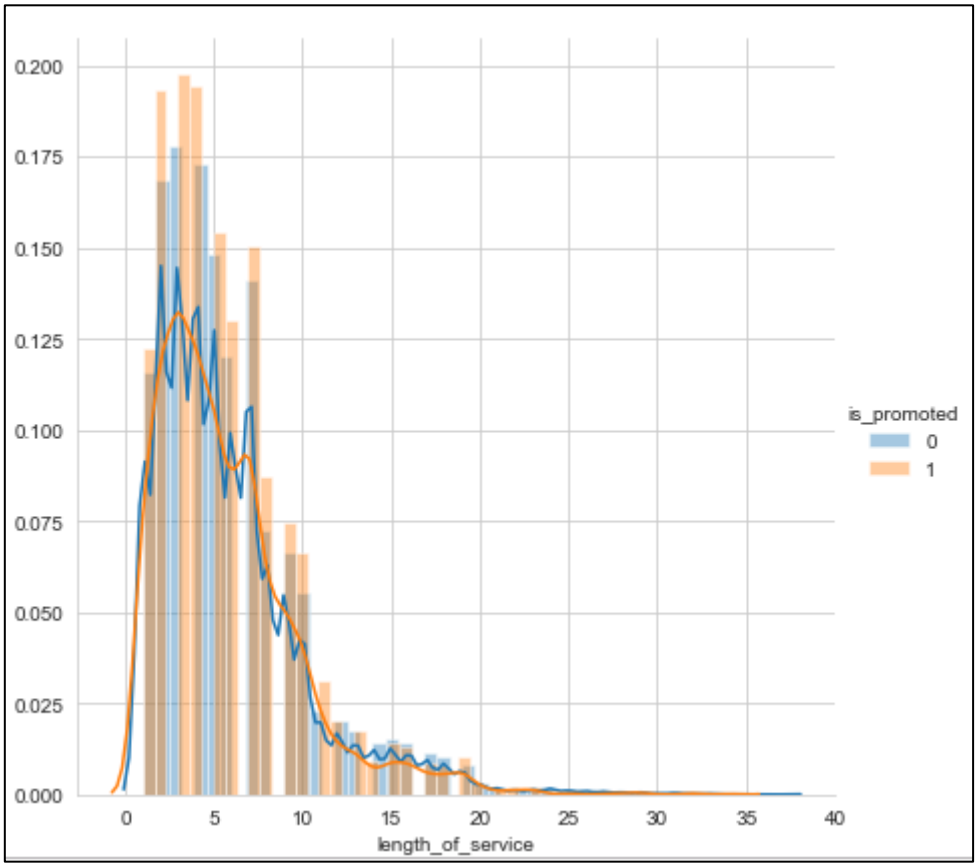

Figure 12: Histogram of Length of service

CHAPTER 4

MODEL BUILDING

4.1. Base model

The base model is created using logistic regression by including all the features in the dataset. The categorical features are one hot encoded before creating the model. The data frame after encoding is shown in figure 13. After encoding the column of the data frame becomes 67. The continuous variables present in the data are standardized with mean and standard deviation before building the model.

employee_id	age	length_of_service	avg_training_score	is_promoted	department_Finance	department_HR	department_Legal	department_Operations	departmer
65438	35	8	49	0	0	0	0	0	
65141	30	4	60	0	0	0	0	1	
7513	34	7	50	0	0	0	0	0	
2542	39	10	50	0	0	0	0	0	
48945	45	2	73	0	0	0	0	0	

5 rows × 67 columns

Figure 13: Data frame after encoding

4.1.1. Performance metric

Since the data is highly imbalanced performance metrics like accuracy, ROC AUC score cannot be used for evaluating the model performance. In this case f1 score is used for evaluating the model performance.

4.1.2. Base model using Logistic Regression

The Logistic Regression is performed on the train data and the result is shown in the figure 14. f1 score for this model is found to be 0.36. The hyperparameter of the Logistic Regression which is ‘C’ is tuned by using GridSearchCV and the result is shown in figure 15. The f1 score after tuning the hyperparameter is found to be 0.39.

```
Confusion Matrix of the model:
[[9977   51]
 [ 712  222]]
----------------------------------------
Accuracy Score: 0.9303959131545338
----------------------------------------
Classification report:
              precision    recall  f1-score   support

           0       0.93      0.99      0.96     10028
           1       0.81      0.24      0.37       934

   micro avg       0.93      0.93      0.93     10962
   macro avg       0.87      0.62      0.67     10962
weighted avg       0.92      0.93      0.91     10962

----------------------------------------
f1 score: 0.36785418392709196
----------------------------------------
ROC AUC score: 0.6163008031473332
----------------------------------------

Cross Validation using KFold:
Accuracy score using KFold cross validation:
cross_val_score: 0.3829113924050633
cross_val_score: 0.37795275590551186
cross_val_score: 0.37437603993344426
cross_val_score: 0.380178716490658
cross_val_score: 0.3648315529991783
Mean Acuuracy Score: 0.37605009154677116
```

Figure 14: Result of Logistic Regression

```
Confusion Matrix of the model:
[[9958   70]
 [ 689  245]]
----------------------------------------
Accuracy Score: 0.9307608100711549
----------------------------------------
Classification report:
              precision    recall  f1-score   support

           0       0.94      0.99      0.96     10028
           1       0.78      0.26      0.39       934

   micro avg       0.93      0.93      0.93     10962
   macro avg       0.86      0.63      0.68     10962
weighted avg       0.92      0.93      0.91     10962

----------------------------------------
f1 score: 0.3923138510808647
----------------------------------------
ROC AUC score: 0.6276660895531058
----------------------------------------

Cross Validation using KFold:
Accuracy score using KFold cross validation:
cross_val_score: 0.40853658536585363
cross_val_score: 0.4003392705682782
cross_val_score: 0.40927258193445243
cross_val_score: 0.41809672386895474
cross_val_score: 0.39171974522292996
Mean Acuuracy Score: 0.4055929813920939
```

Figure 15: Result of Logistic Regression after hyperparameter tuning

4.1.3. Logistic Regression using GLM

The statistical significance of the features is found by using Logistic Regression in Generalized Linear Models (GLM). The hypothesis for this testing is as follows

- Null hypothesis, H0: There is no relationship between dependent and independent variables
- Alternate hypothesis, H1: There is relationship between dependent and independent variables

The features which have p values > 0.05 indicates that the null hypothesis is true and hence it can be removed from building the logistic regression model. The p value for each feature is shown in figure 16.

Model:	GLM	AIC:	21486.3099
Link Function:	logit	BIC:	-575959.1175
Dependent Variable:	is_promoted	Log-Likelihood:	-10676.
Date:	2019-04-07 12:50	LL-Null:	-15961.
No. Observations:	54808	Deviance:	21352.
Df Model:	66	Pearson chi2:	4.40e+04
Df Residuals:	54741	Scale:	1.0000
Method:	IRLS		

	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
const	-10.4052	0.2344	-44.3970	0.0000	-10.8646	-9.9459
department_Finance	7.0614	0.1604	44.0367	0.0000	6.7471	7.3757
department_HR	9.9529	0.2117	47.0164	0.0000	9.5380	10.3678

Figure 16: GLM result for Logistic Regression 1

department_Operations	7.2777	0.1417	51.3680	0.0000	7.0000	7.5553
department_Procurement	4.4102	0.1058	41.6783	0.0000	4.2028	4.6176
department_RandD	-0.5515	0.1478	-3.7325	0.0002	-0.8411	-0.2619
department_SalesMarketing	10.4652	0.1883	55.5865	0.0000	10.0962	10.8342
department_Technology	1.7745	0.0766	23.1719	0.0000	1.6244	1.9246
region_region_10	0.1519	0.2388	0.6362	0.5247	-0.3161	0.6200
region_region_11	-0.3142	0.2168	-1.4490	0.1473	-0.7392	0.1108
region_region_12	-0.4122	0.2888	-1.4274	0.1535	-0.9782	0.1538
region_region_13	0.0583	0.1872	0.3116	0.7553	-0.3086	0.4253
region_region_14	0.0209	0.2248	0.0928	0.9260	-0.4197	0.4615
region_region_15	0.0851	0.1875	0.4537	0.6501	-0.2825	0.4527
region_region_16	-0.1001	0.2073	-0.4827	0.6293	-0.5065	0.3063

Figure 17: GLM result for Logistic Regression 2

The features which have p value < 0.05 is used for building the next Logistic Regression model.

Future work

- Feature engineering
- Partial Least Square Discriminant analysis
- Building different machine learning models
- Model evaluation
- Insight derivation from model

Python code

```
def model(clf,X_train,Y_train):
    x_train, x_test, y_train, y_test = train_test_split(X_train, Y_train, test_size=0.20, stratify =
Y_train, random_state = 99)
    clf.fit(x_train,y_train)
    y_pred = clf.predict(x_test)
    print('Scores of the Model')
    print('Confusion Matrix of the model:')
    print(confusion_matrix(y_test,y_pred))
    print('----------------------------------------')
    print('Accuracy Score:',accuracy_score(y_test,y_pred))
    print('----------------------------------------')
    print('Classification report:')
    print(classification_report(y_test, y_pred))
```

```
print('----------------------------------------')
print('f1 score:',f1_score(y_test,y_pred))
print('----------------------------------------')
print('ROC AUC score:',roc_auc_score(y_test,y_pred))
print('----------------------------------------')
print('')
print('Cross Validation using KFold:')
kf = KFold(n_splits=5,random_state=99)
print('Accuracy score using KFold cross validation:')
score = cross_val_score(clf, X_train, Y_train, cv=kf,scoring='f1', n_jobs=1)
for i in score:
    print('cross_val_score:',i)
print('Mean Acuuracy Score:',score.mean())
```