Employees promotion prediction

Business Problem

- HR Team in JMD company need to take a decision on employee promotion for upcoming year.
- There are so many variables involved in choosing right employee for promotion, year by year this is getting difficult for HR Team.
- There are so many employees working on different region, departments, task of finding employee for a promotion is monumental task for HR team.

The Solution

- Find a simple way to filter employee for a promotion based on available variable.
- We can sort employees based on different categories and generate meaningful inputs for both HR team and employees.
- We can then able to integrate final results directly on employee portal for better decision making and it saves ton of time for a HR Team.

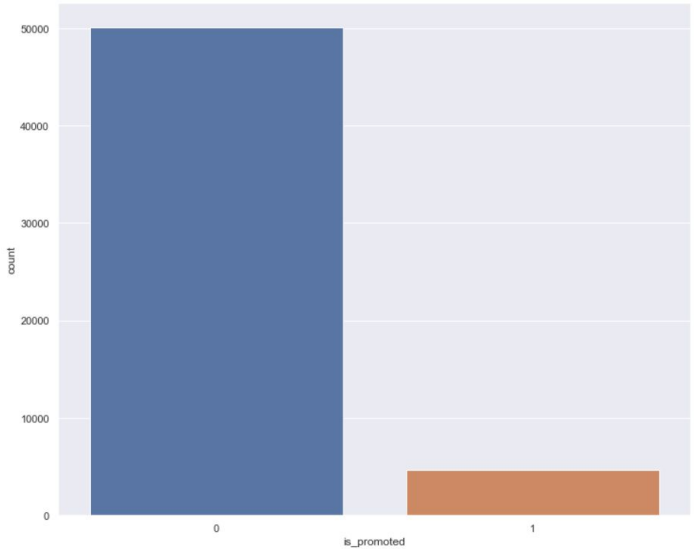
Why is automating important?

- Scalability
- Real-time analysis
- Consistent Criteria

Dataset

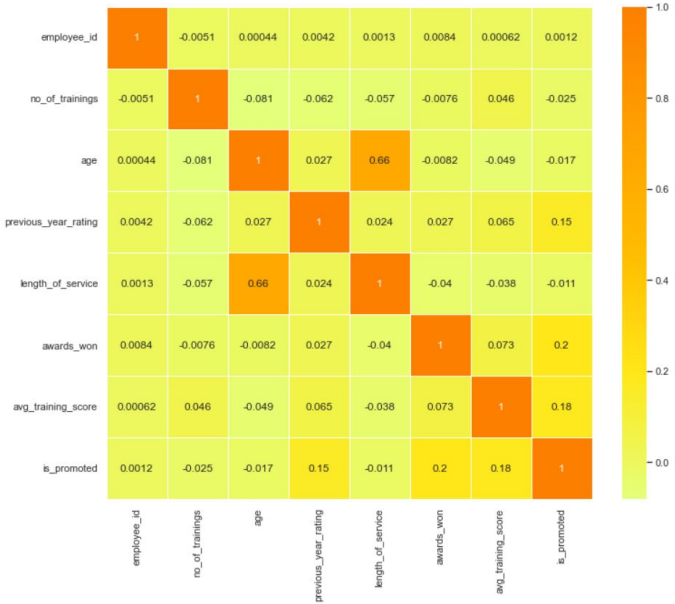
- Previous year employee promotion database taken into consideration for this problem.
- Data contains **54808 * 13** samples.
- Employee data spread across 9 departments, 20+ regions and contain both male and female employees.

Look inside data - 1



- The bar chart represents how much the data is balanced, and clearly it's not balanced data set.
- Promoted employees in the data set are way less compare to others.
- This will create highly biased result if this didn't treated properly.
- We used oversampling and undersampling technique to solve this problem.

Look inside data - 2



- Length of service of a employee is correlated with age,
- Other than above mentioned correlation we didn't find any other variable which related to each other.

Cleaning

- 20% data as test dataset
- 80% data as train and validation dataset
- Used imputation techniques to treat missing values.
- After cleaning the data set samples stands as 43842 * 10(1 target variable & 9 dependent variable)

- department
- education
- gender
- no_of_trainings
- age
- previous_year_rating
- length_of_service
- awards_won
- avg_training_score

Model performance - Random forest

- Random Forest is a great algorithm, for both classification and regression problem
- Its default hyperparameters already return great results and the system is great at avoiding overfitting.
- The result based on over sampled data

```
Training Accuracy : 0.986949200112265
Testing Accuracy : 0.9350130971685169
              precision    recall  f1-score   support

           0       0.93      0.94      0.93      7931
           1       0.94      0.93      0.94      8103

    accuracy                           0.94     16034
   macro avg       0.94      0.94      0.94     16034
weighted avg       0.94      0.94      0.94     16034
```

Challenges

- Unbalanced data samples
- Domain knowledge on various department.
- Less features to generate any meaningful results.

Future works

- Gather more and balanced data across different features like region, gender, education etc.,
- Add more features like attendance, immediate manager rating etc., to get more accurate result as possible.

Thankyou!