CLARUSWAY
WAY TO REINVENT YOURSELF

الأكاديمية السعودية الرقمية
SAUDI DIGITAL ACADEMY

Employee Churn Analysis

G4

prediction

Table of Content

Employee Churn Analysis

Introduction

The dataset we are working on is an HR dataset that primarily focuses on employee-related information. It consists of 14,999 samples, each representing an employee in the company. Two main categories exist within the dataset: employees who stayed with the company and employees who left.

Introduction

Attributes of the dataset in detail:

- **Satisfaction Level:**
  Employee satisfaction point, ranging from 0 to 1.
- **Last Evaluation:**
  Evaluated performance by the employer, also ranging from 0 to 1.
- **Number of Projects:**
  Indicates the number of projects assigned to an employee.
- **Average Monthly Hours:**
  Average number of hours an employee works in a month.
- **Time Spent at the Company:**
  The number of years an employee has spent in the company.

Introduction

Attributes of the dataset in detail:

- **Work Accident:**
  Whether an employee has had a work accident or not.
- **Promotion in Last 5 Years:**
  Whether an employee has received a promotion in the last 5 years.
- **Departments:**
  Employee's working department or division.
- **Salary:**
  Salary level of the employee, categorized as low, medium, or high.
- **Left:**
  Binary variable indicating whether the employee has left the company or not.

Data Cleaning

Data Type

- **Categorical Variables:**
  ['Departments', 'salary' ,'Work_accident', 'left', 'promotion_last_5years']

- **Numerical Variables**:['satisfaction_level', 'last_evaluation','number_project', 'average_montly_hours', 'time_spend_company']

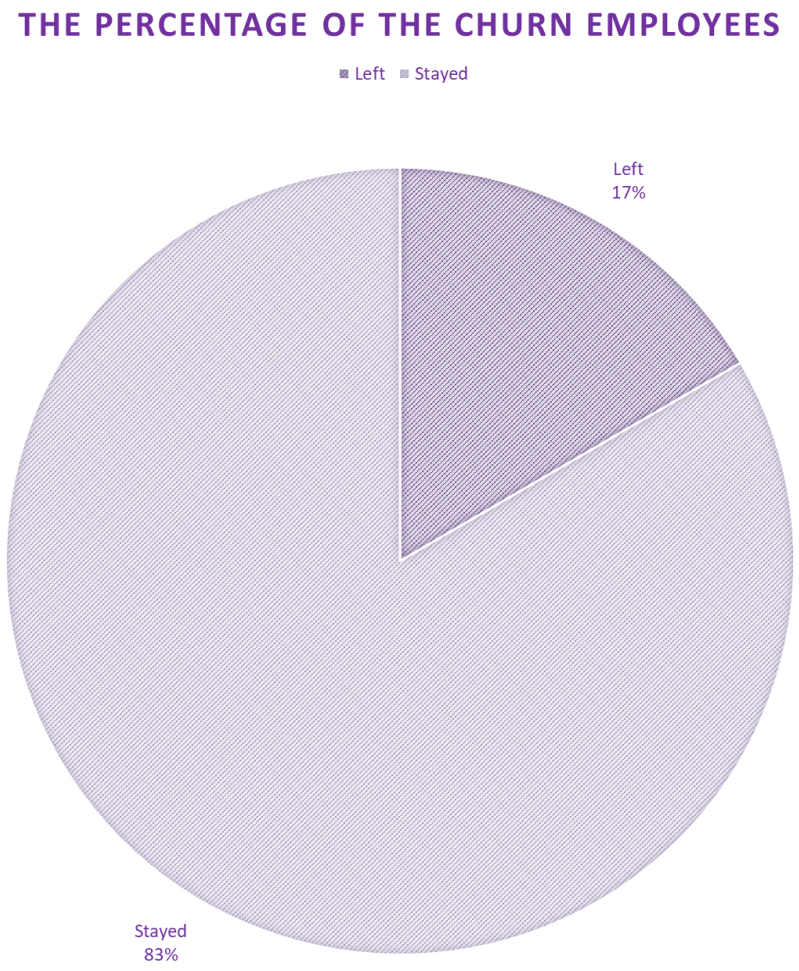
Duplicates

Num of Duplicated Values: 3008

Missing Values

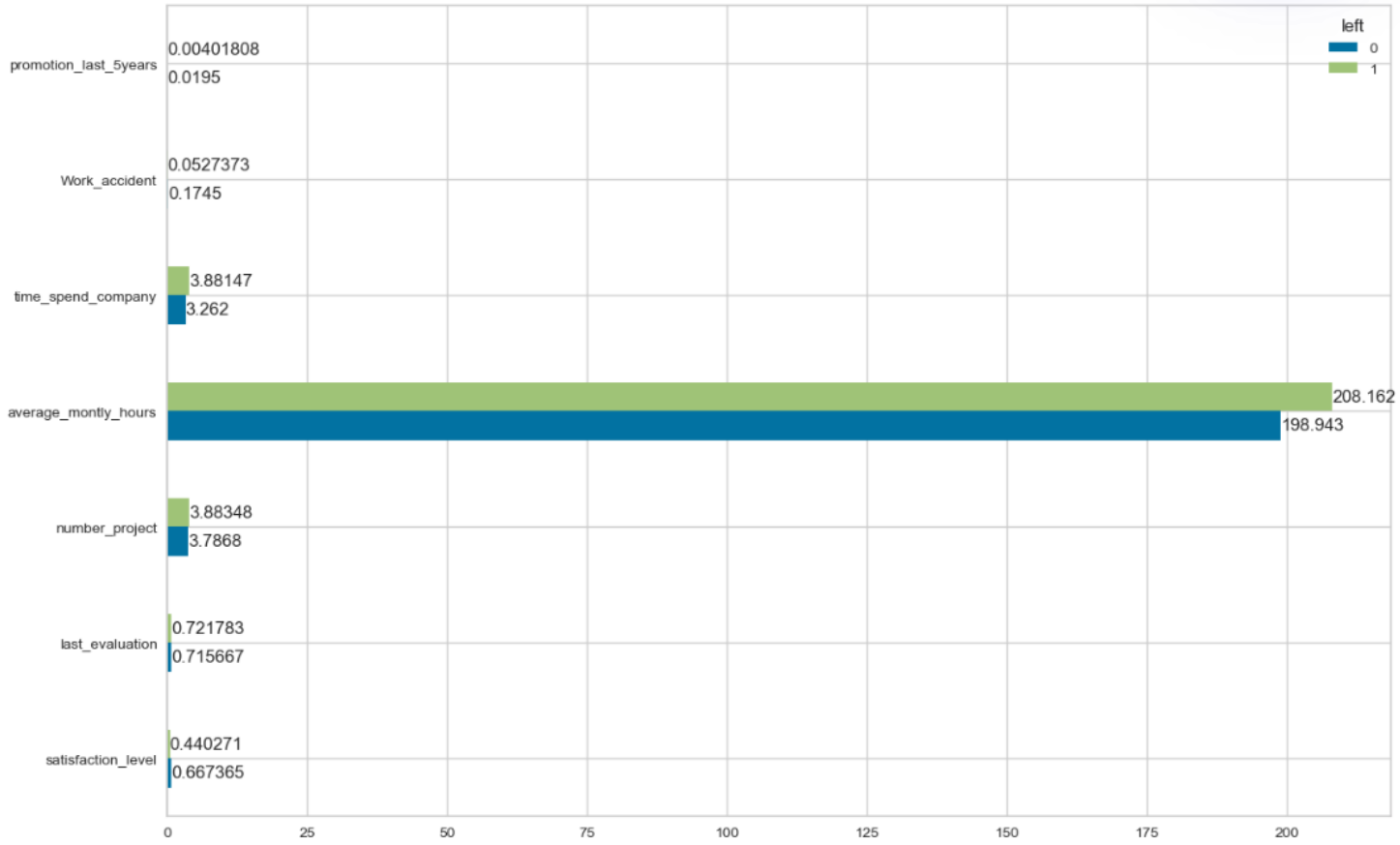
No Missing values

Explore Data Analysis



Explore Data Analysis

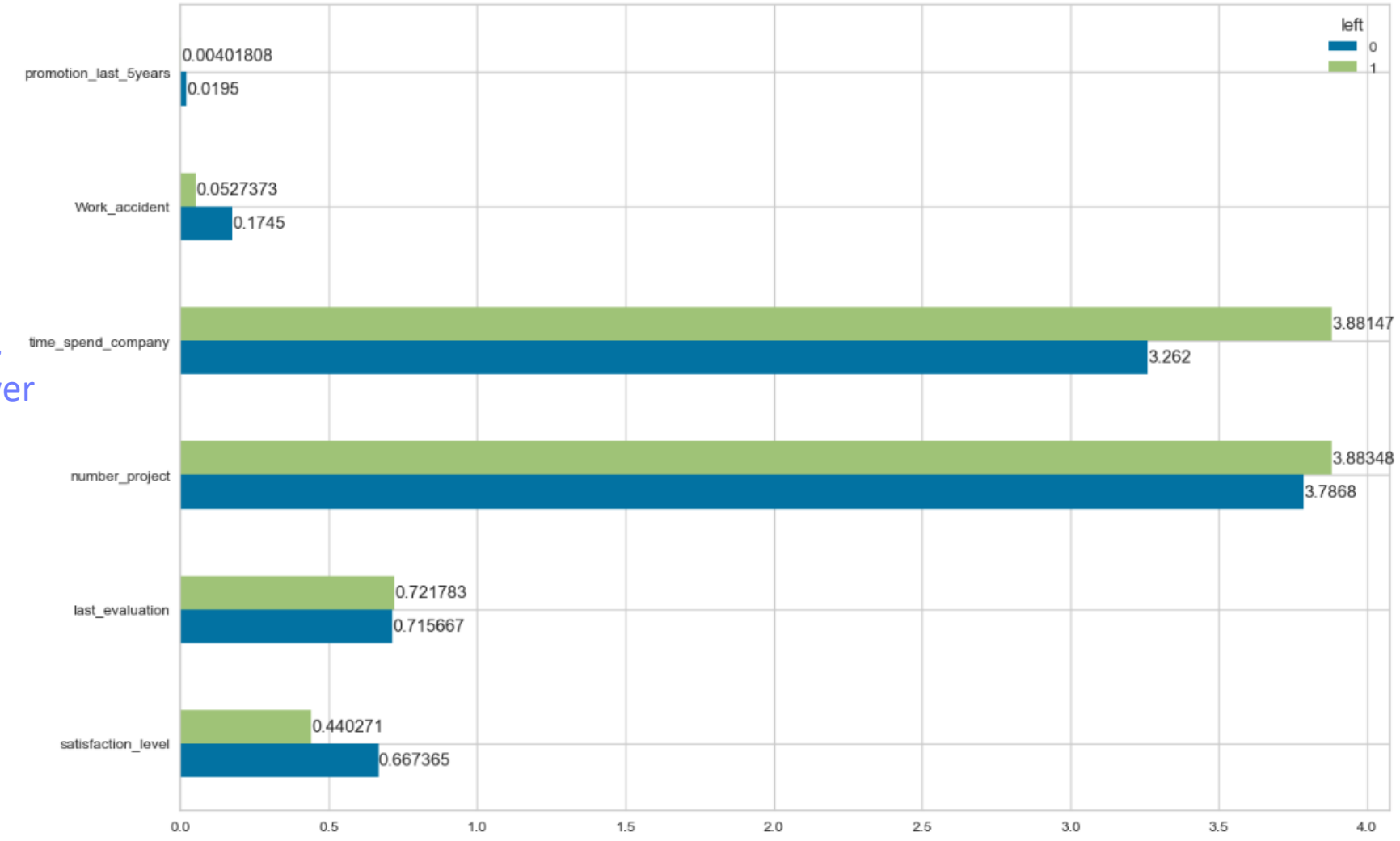
Employee who left have higher average monthly hours.



0: Stayed 1: Left

Explore Data Analysis

Employees who left have higher time spend company, number of projects, and lower satisfaction

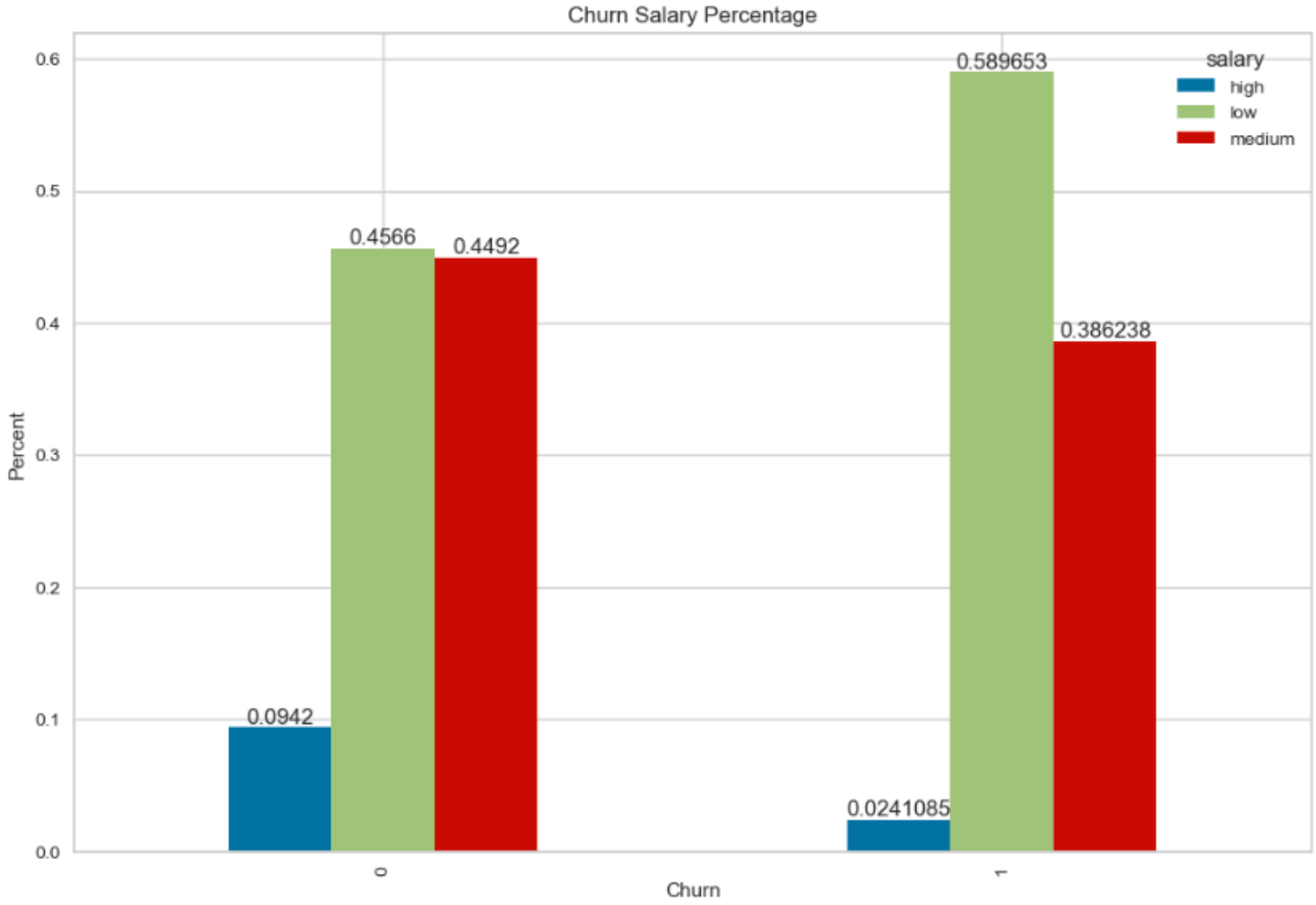


0: Stayed 1: Left

Explore Data Analysis

It seems that most the employees who left work the longest hours, have more project, have the highest time spent in the company, and the lowest salary.

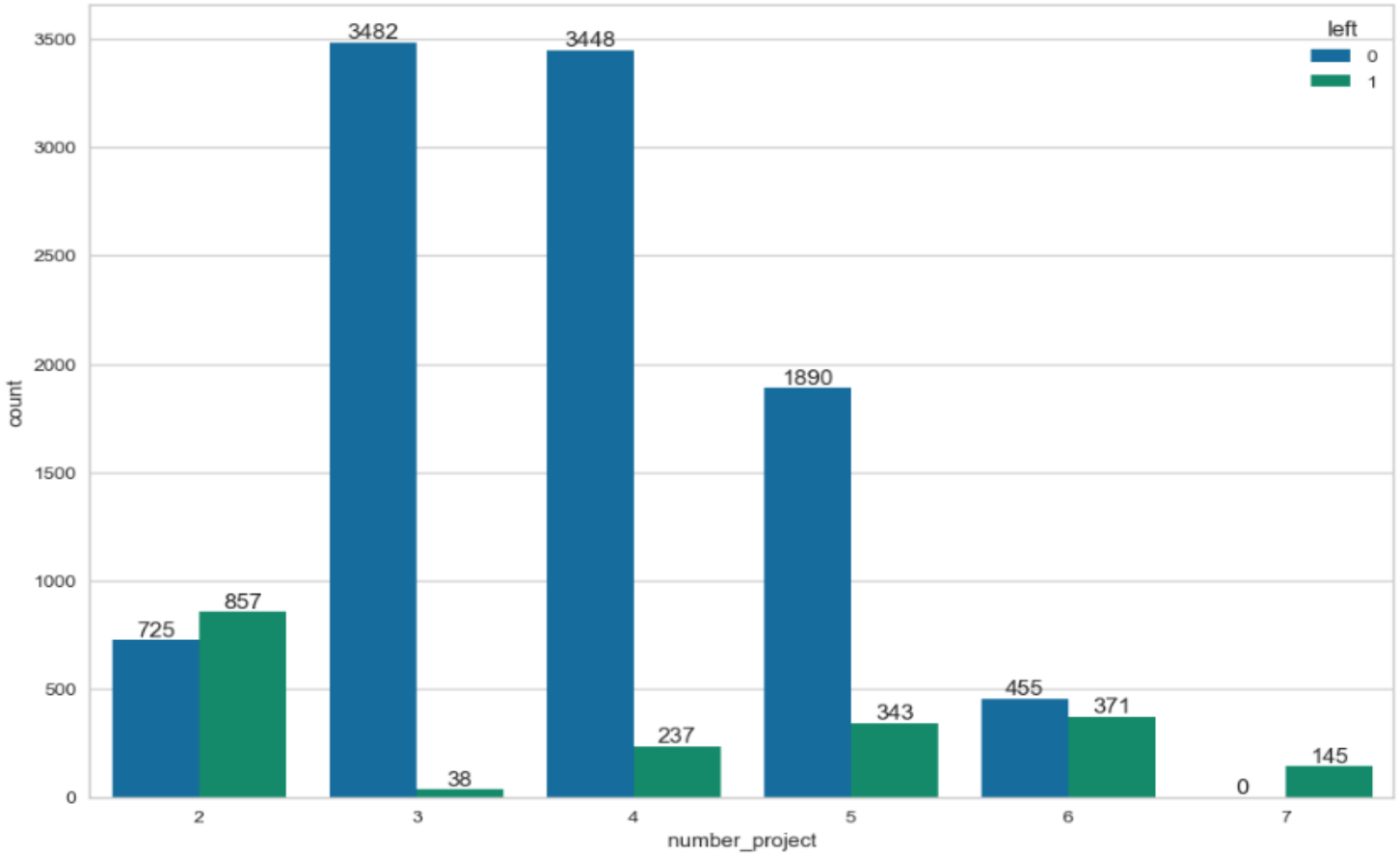
That may be the reason for the low satisfaction!



0: Stayed 1: Left

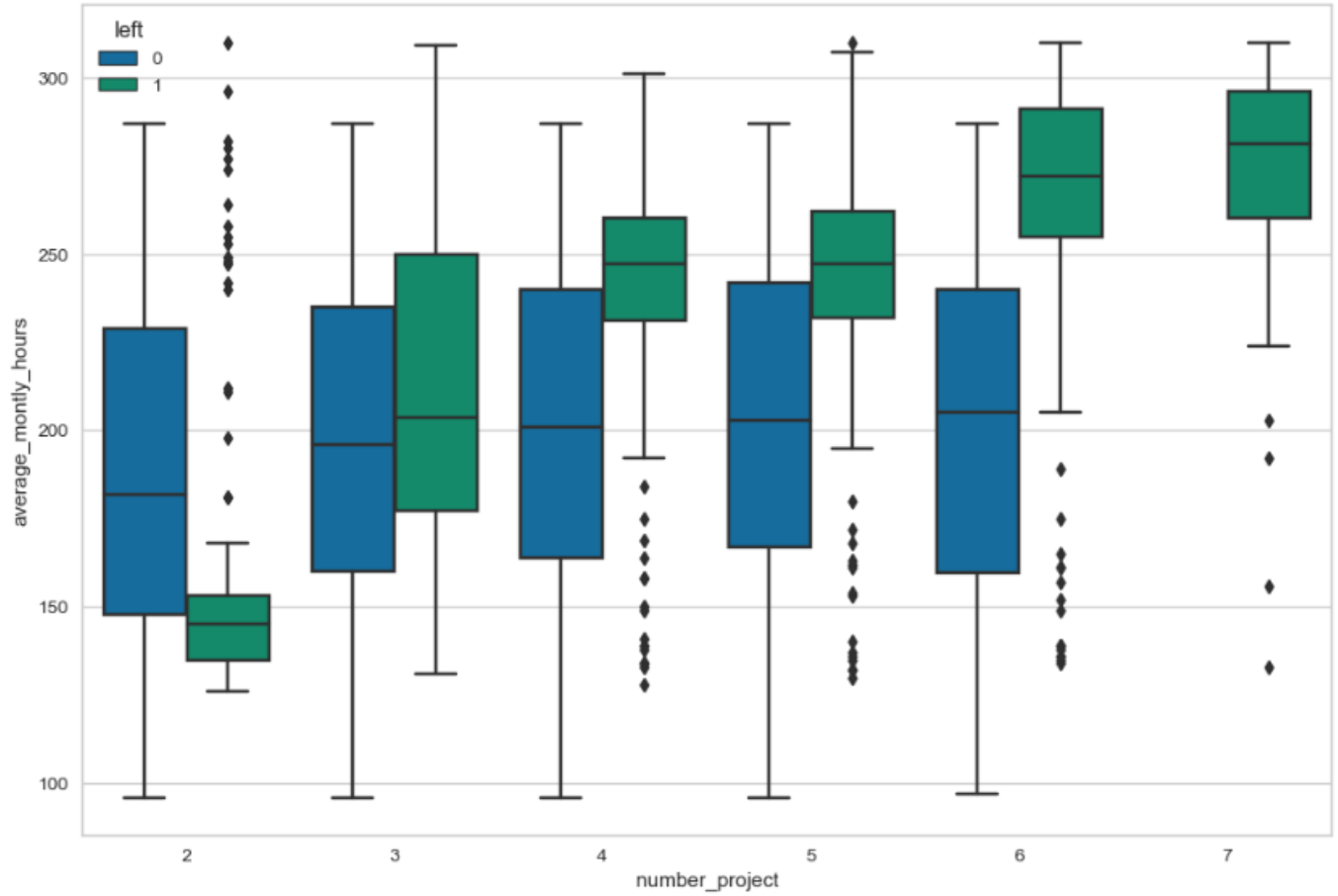
Explore Data Analysis

- All employees who did 7 project had left
- Employees who left with number of projects 2 and 6 are higher than half who stayed



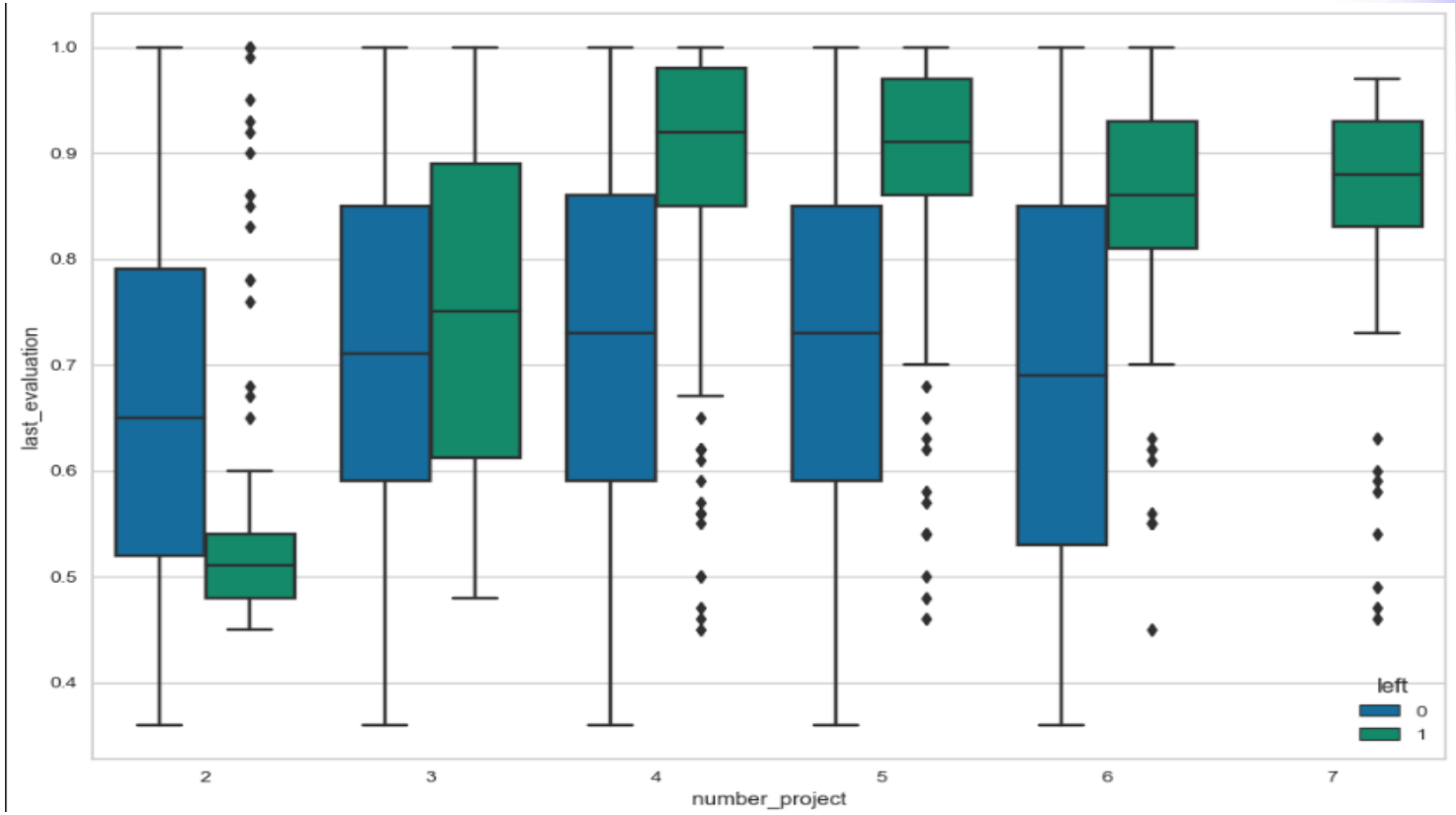
Explore Data Analysis

- For employees who left, the average monthly hours are increasing with the number of projects to be higher than the employees who stayed
- for employees who stayed, the average monthly hours are similar to each other and doesn't change much with the number of projects, it is lower than the employees who left



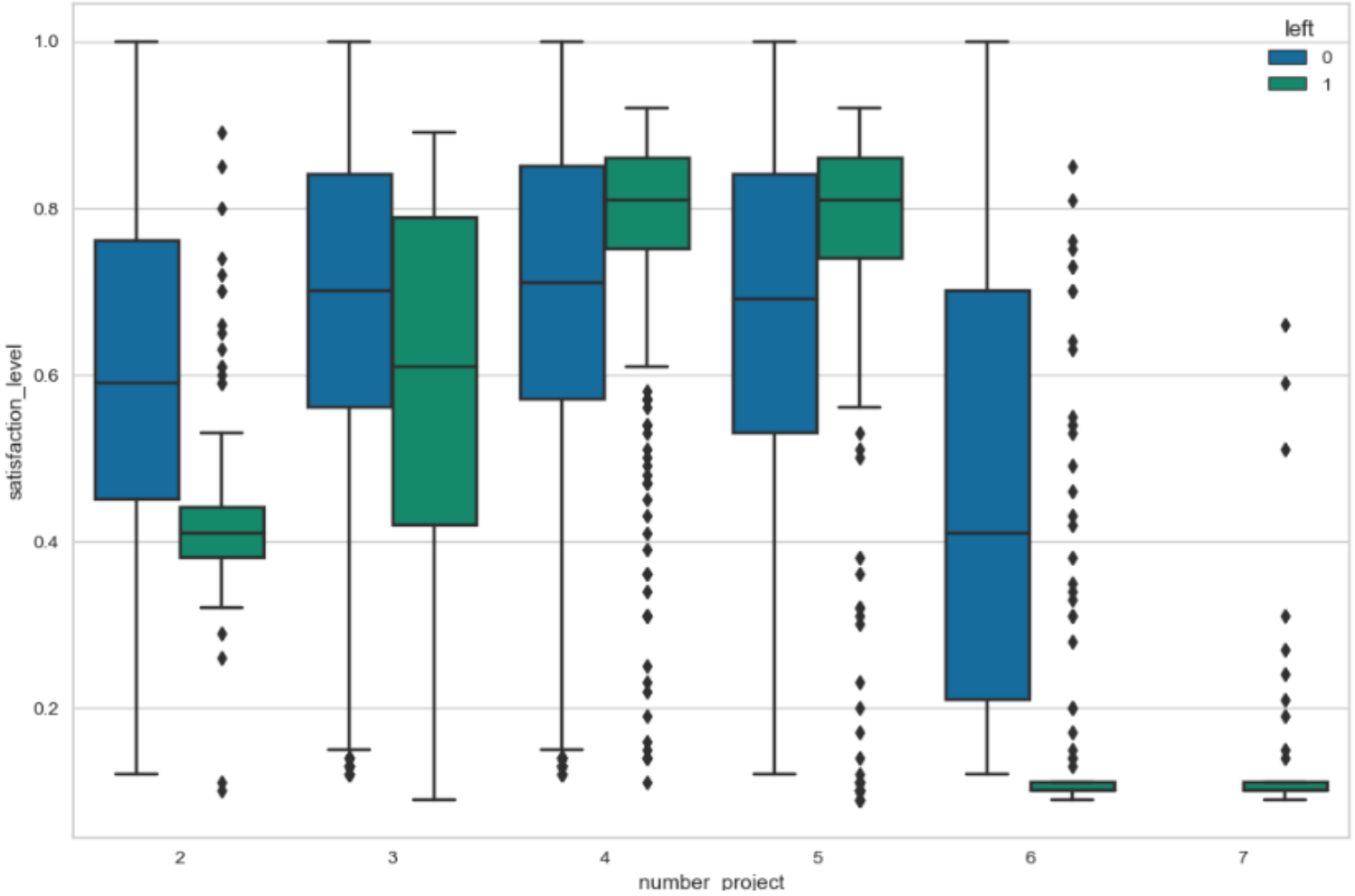
Explore Data Analysis

- Similar to the distribution of average monthly hours
- For employees who left the last evaluation are increasing with the number of projects to be higher than the employees who stayed
- For employees who stayed the last evaluation are similar to each other and doesn't change much with the number of projects and it is lower than the employees who left



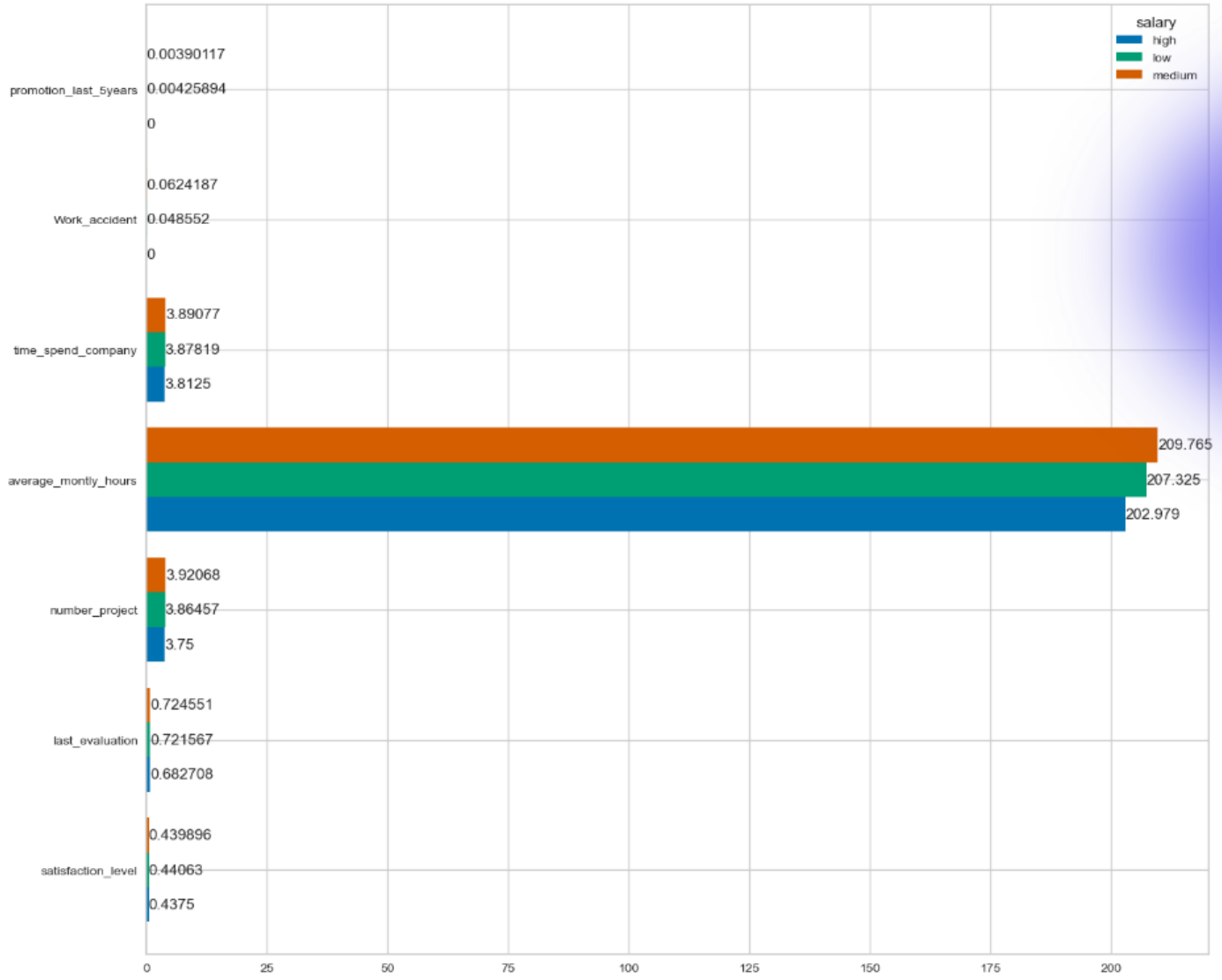
Explore Data Analysis

- For employees who left the satisfaction level are the lowest at number of project 2, 6, and 7
- For employees who stayed the satisfaction level are similar to each other and doesn't change much with the number of projects except for number of project 6



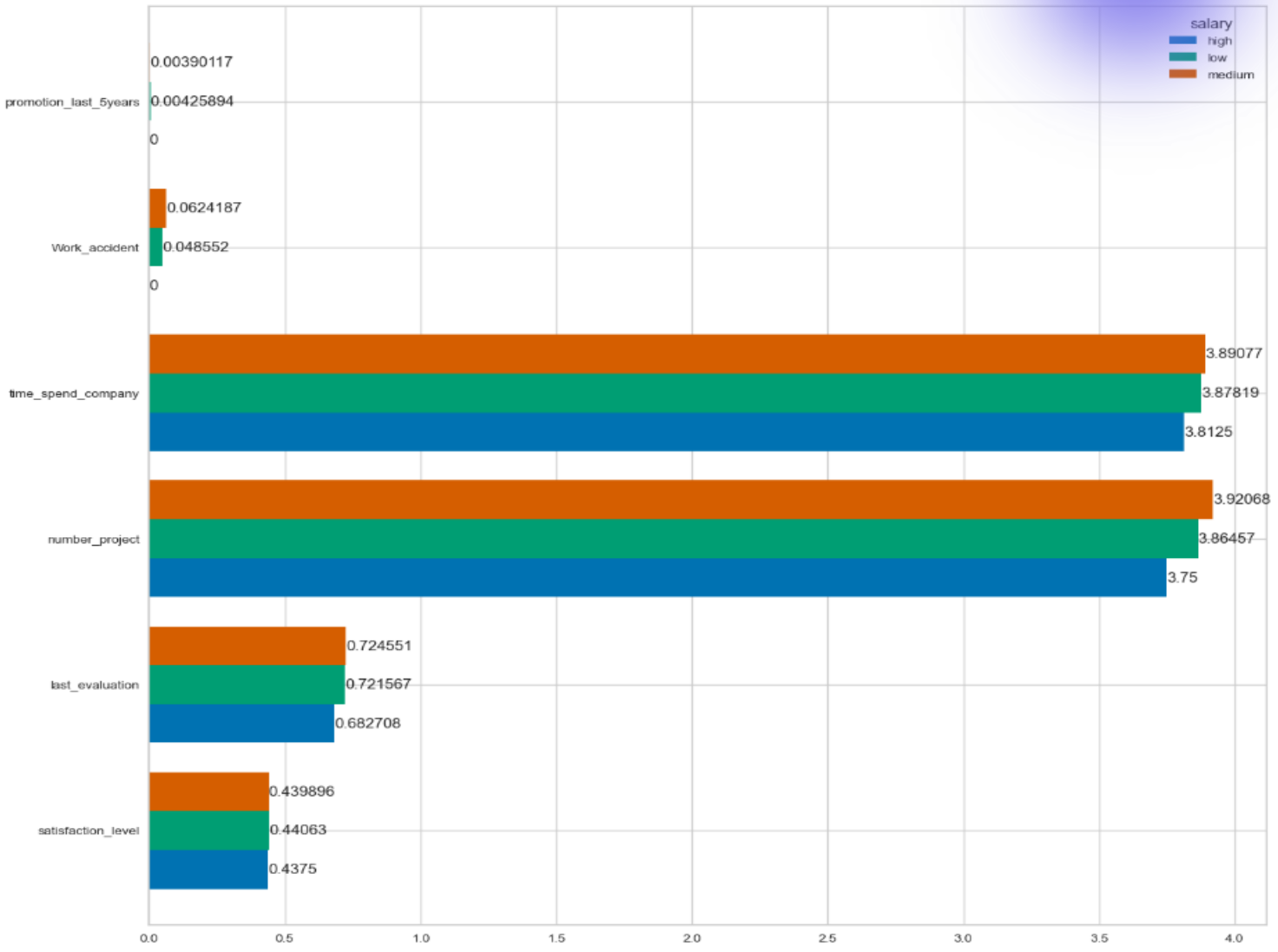
Explore Data Analysis

From average monthly hours it seams that the high salary are the lowest average monthly hours

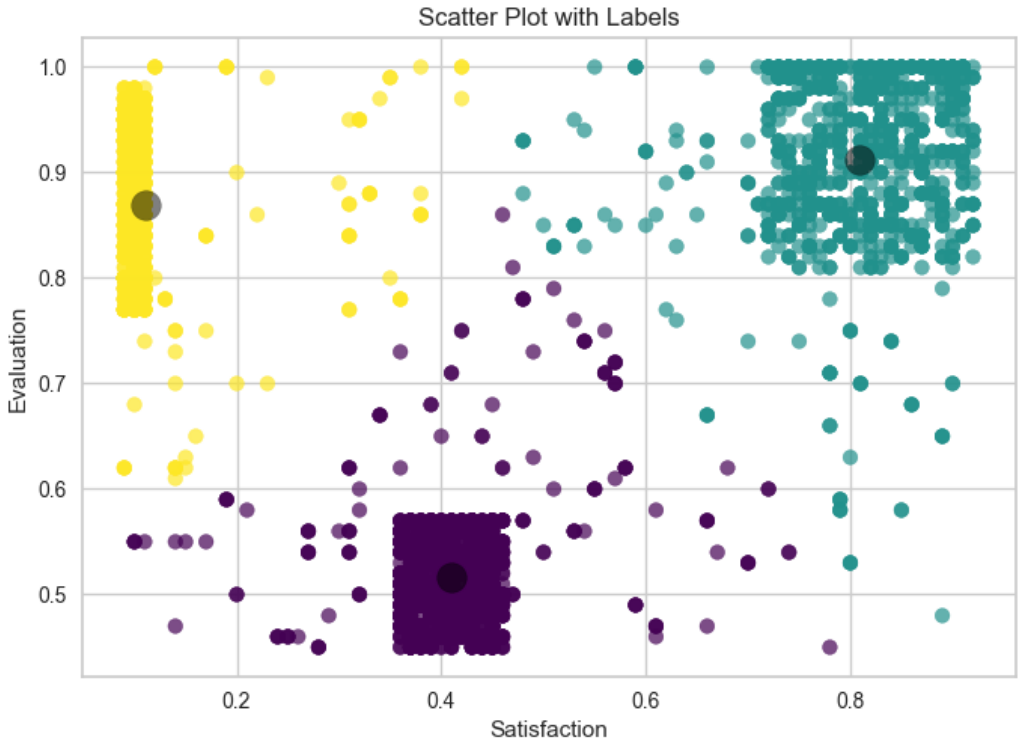


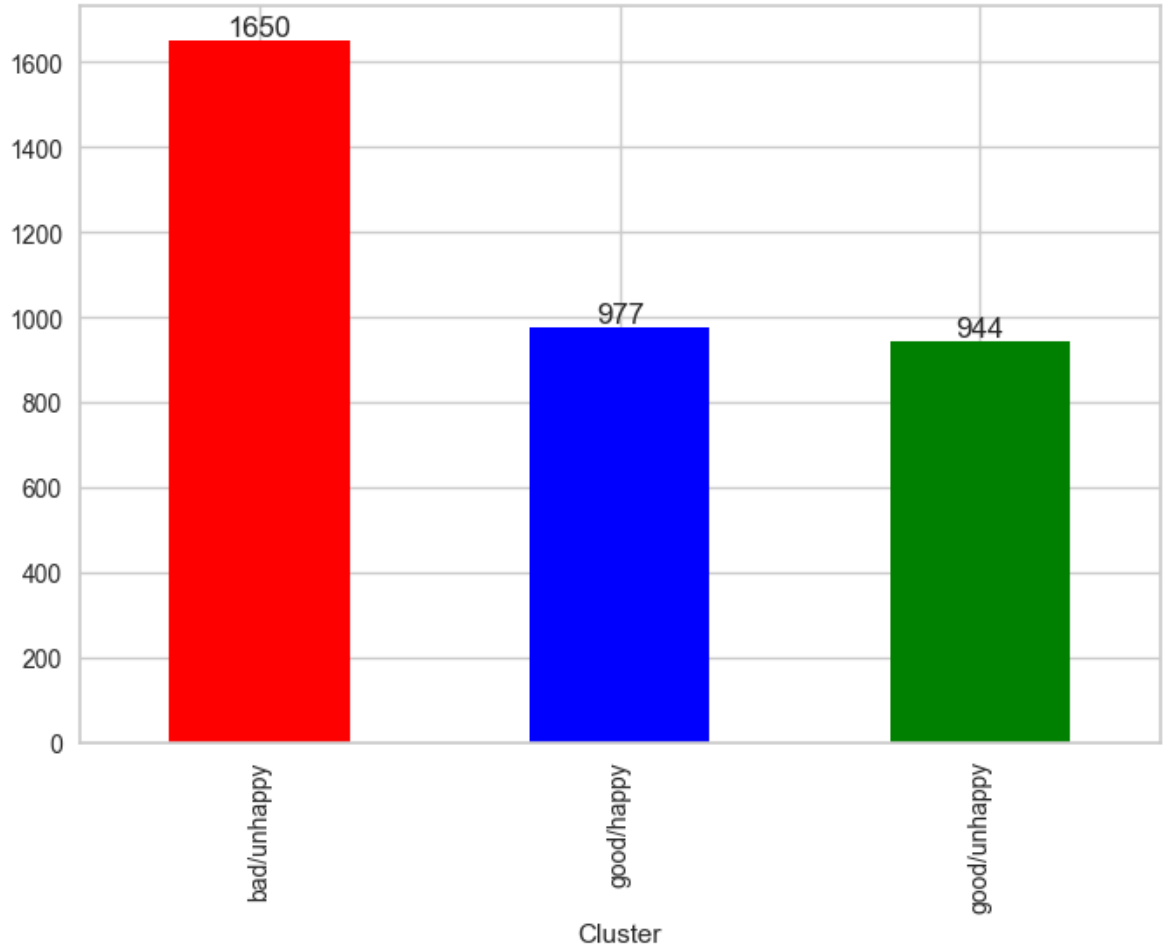
Explore Data Analysis

- It seems that the high salary have the lowest values at everything
- They might be working less and have high salaries



Cluster Analysis





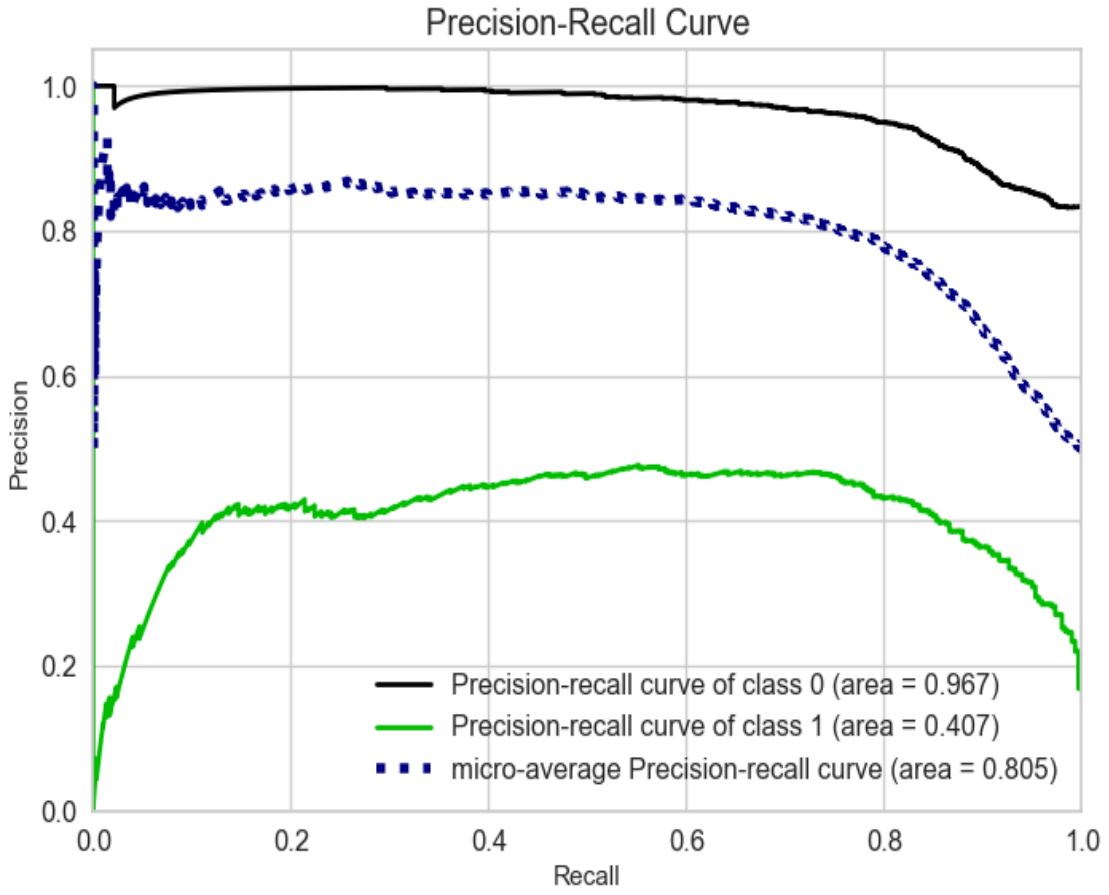
- Most of who leave are bad/unhappy
- Since we saw that the higher the number of project the higher the evaluation, good/unhappy they may be exhausted by the high workload

Predictive Models Building
Preprocessing

- We made the Test size = 15% and Train size = 85%
- We Apply Scaling on the dataset with type Stander Scaler
- We decide the Random state = 101
- we Apply Encoding technique for categorical data
  - One Hot Encoding for Department
  - Ordinal Encoding for Salary

Predictive Models Building
Machine Learning - Classification

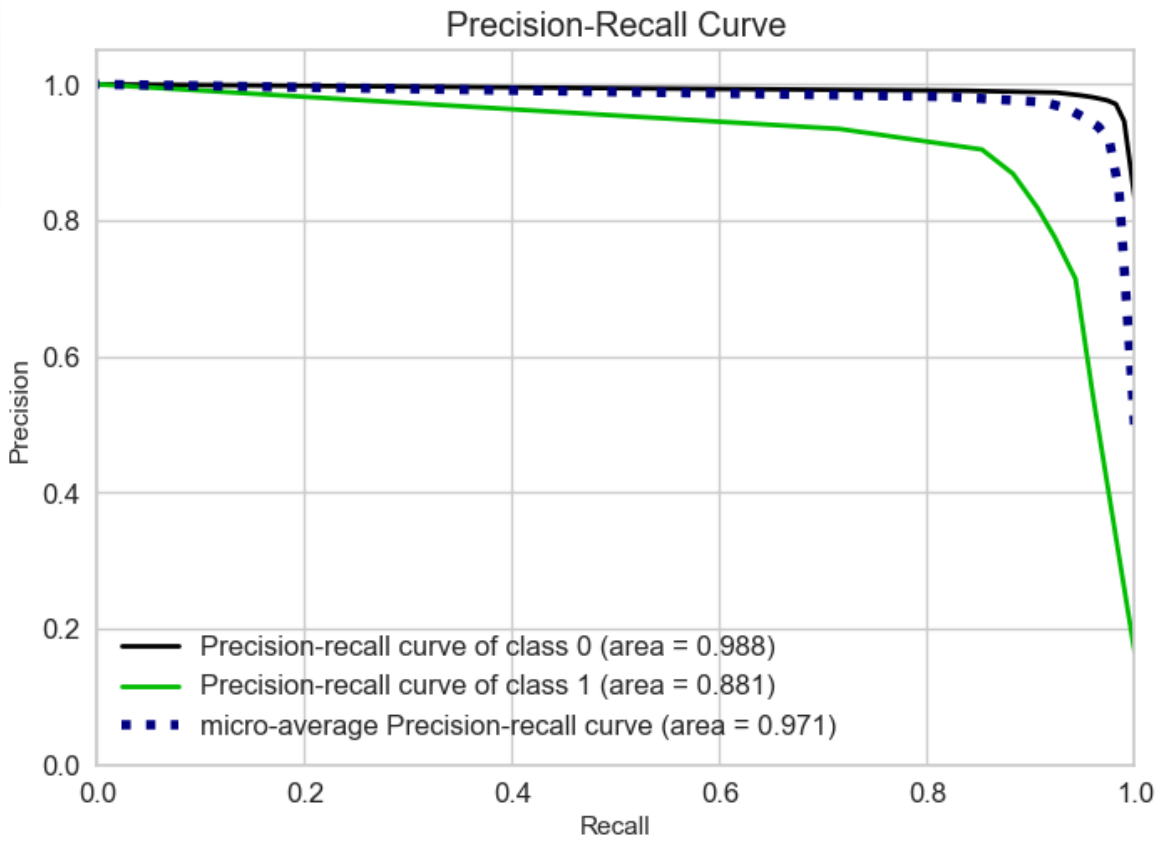
We used Best Model object from GridSearch for Logistic Regression model




```
Test_Set
[[1175  325]
 [  55  244]]
              precision    recall  f1-score   support

           0       0.96      0.78      0.86      1500
           1       0.43      0.82      0.56       299

    accuracy                           0.79      1799
   macro avg       0.69      0.80      0.71      1799
weighted avg       0.87      0.79      0.81      1799


Train_Set
[[6447 2053]
 [ 288 1404]]
              precision    recall  f1-score   support

           0       0.96      0.76      0.85      8500
           1       0.41      0.83      0.55      1692

    accuracy                           0.77     10192
   macro avg       0.68      0.79      0.70     10192
weighted avg       0.87      0.77      0.80     10192
```

Predictive Models Building
Machine Learning - Classification

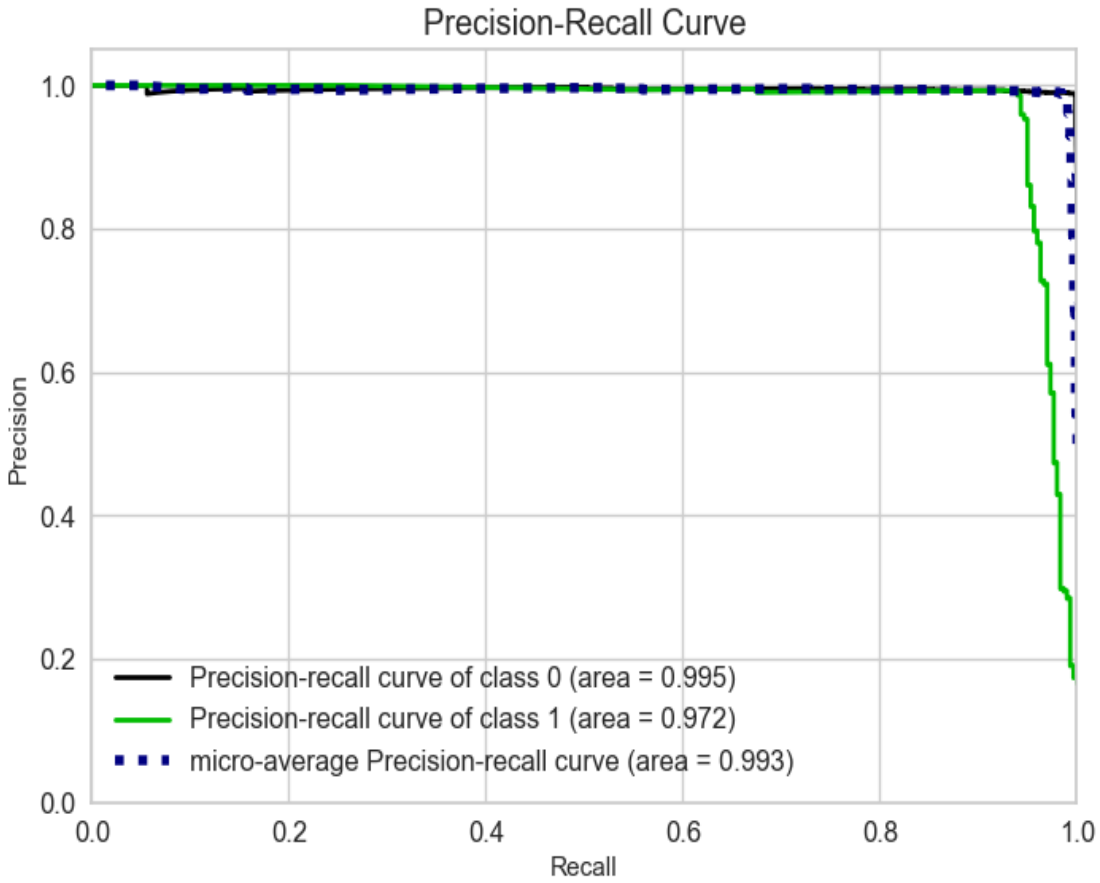
We used k= 7 for KNN model




```
Test_Set
[[1440   60]
 [  28  271]]
              precision    recall  f1-score   support

           0       0.98      0.96      0.97      1500
           1       0.82      0.91      0.86       299

    accuracy                           0.95      1799
   macro avg       0.90      0.93      0.92      1799
weighted avg       0.95      0.95      0.95      1799


Train_Set
[[8268  232]
 [ 162 1530]]
              precision    recall  f1-score   support

           0       0.98      0.97      0.98      8500
           1       0.87      0.90      0.89      1692

    accuracy                           0.96     10192
   macro avg       0.92      0.94      0.93     10192
weighted avg       0.96      0.96      0.96     10192
```

Predictive Models Building
Machine Learning - Classification

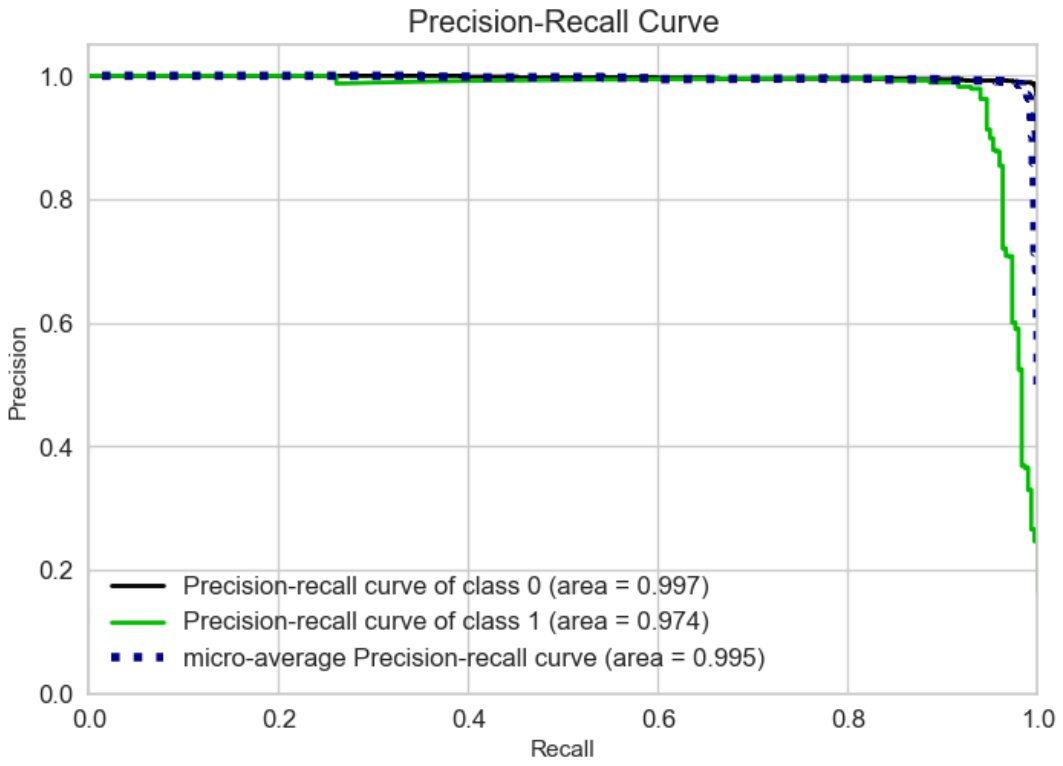
We used **RF** best model from GridSearch




```
Test_Set
[[1495    5]
 [  17  282]]
              precision    recall  f1-score   support

           0       0.99      1.00      0.99      1500
           1       0.98      0.94      0.96       299

    accuracy                           0.99      1799
   macro avg       0.99      0.97      0.98      1799
weighted avg       0.99      0.99      0.99      1799


Train_Set
[[8489   11]
 [ 100 1592]]
              precision    recall  f1-score   support

           0       0.99      1.00      0.99      8500
           1       0.99      0.94      0.97      1692

    accuracy                           0.99     10192
   macro avg       0.99      0.97      0.98     10192
weighted avg       0.99      0.99      0.99     10192
```

Predictive Models Building
Machine Learning - Classification

We used GridSearch for XGBoost with param_grid and we focused on Recall



Test_Set

```
[[1489   11]
 [  18  281]]
```

	precision	recall	f1-score	support
0	0.99	0.99	0.99	1500
1	0.96	0.94	0.95	299
accuracy			0.98	1799
macro avg	0.98	0.97	0.97	1799
weighted avg	0.98	0.98	0.98	1799

Train_Set

```
[[8468   32]
 [ 122 1570]]
```

	precision	recall	f1-score	support
0	0.99	1.00	0.99	8500
1	0.98	0.93	0.95	1692
accuracy			0.98	10192
macro avg	0.98	0.96	0.97	10192
weighted avg	0.98	0.98	0.98	10192

Predictive Models Building
Deep Learning - **without weight**

Layers used:

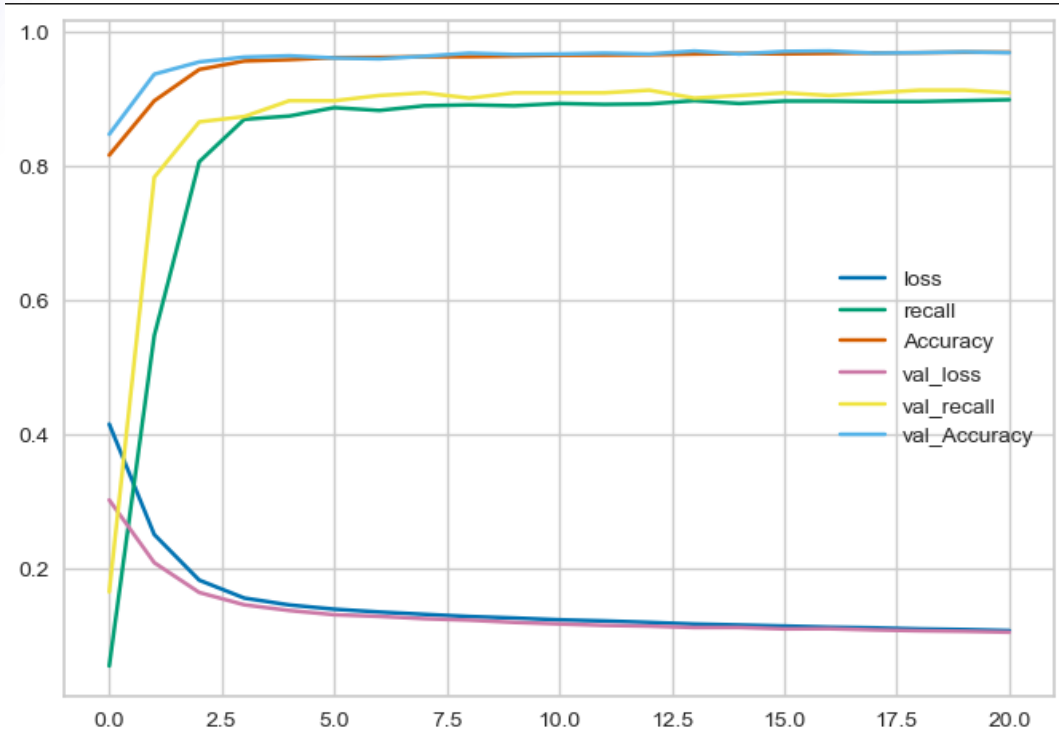
- First layer = 16 with Relu
- Second layer = 8 with Relu
- Output layer = 1 with sigmoid

- Adam optimizer is used
- Loss binary cross entropy since it is a binary classification,
- Early stopping with monitor on Max val_recall

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 16)                304

 dense_1 (Dense)             (None, 8)                 136

 dense_2 (Dense)             (None, 1)                 9

=================================================================
Total params: 449
Trainable params: 449
Non-trainable params: 0
_________________________________________________________________
```

Predictive Models Building
Deep Learning

Evaluation




```
271/271 [==============================] - 0s 639us/step
57/57 [==============================] - 0s 761us/step
Test_Set
[[1457   43]
 [  30  269]]
              precision    recall  f1-score   support

           0       0.98      0.97      0.98      1500
           1       0.86      0.90      0.88       299

    accuracy                           0.96      1799
   macro avg       0.92      0.94      0.93      1799
weighted avg       0.96      0.96      0.96      1799


Train_Set
[[7076  149]
 [ 145 1293]]
              precision    recall  f1-score   support

           0       0.98      0.98      0.98      7225
           1       0.90      0.90      0.90      1438

    accuracy                           0.97      8663
   macro avg       0.94      0.94      0.94      8663
weighted avg       0.97      0.97      0.97      8663
```

Predictive Models Building

Deep Learning - with class weight

Layers used:

- First layer = 16 with Relu
- Second layer = 8 with Relu
- Output layer = 1 with sigmoid

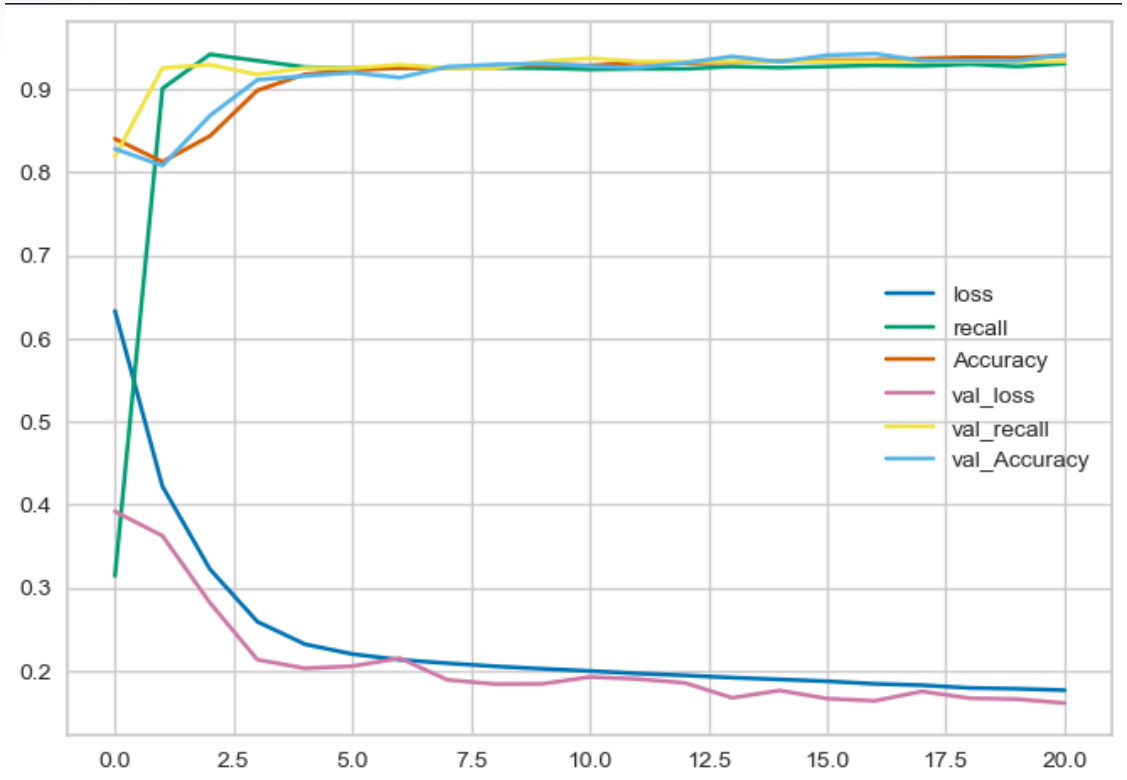
- Adam optimizer is used
- Loss binary cross entropy since it is a binary classification,
- Early stopping with monitor on Max val_recall
- We add class weight = “balanced”

```
{0: 0.599515570934256, 1: 3.0121696801112656}
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 16)                304

 dense_1 (Dense)             (None, 8)                 136

 dense_2 (Dense)             (None, 1)                 9

=================================================================
Total params: 449
Trainable params: 449
Non-trainable params: 0
_________________________________________________________________
```

Predictive Models Building
Deep Learning

Evaluation



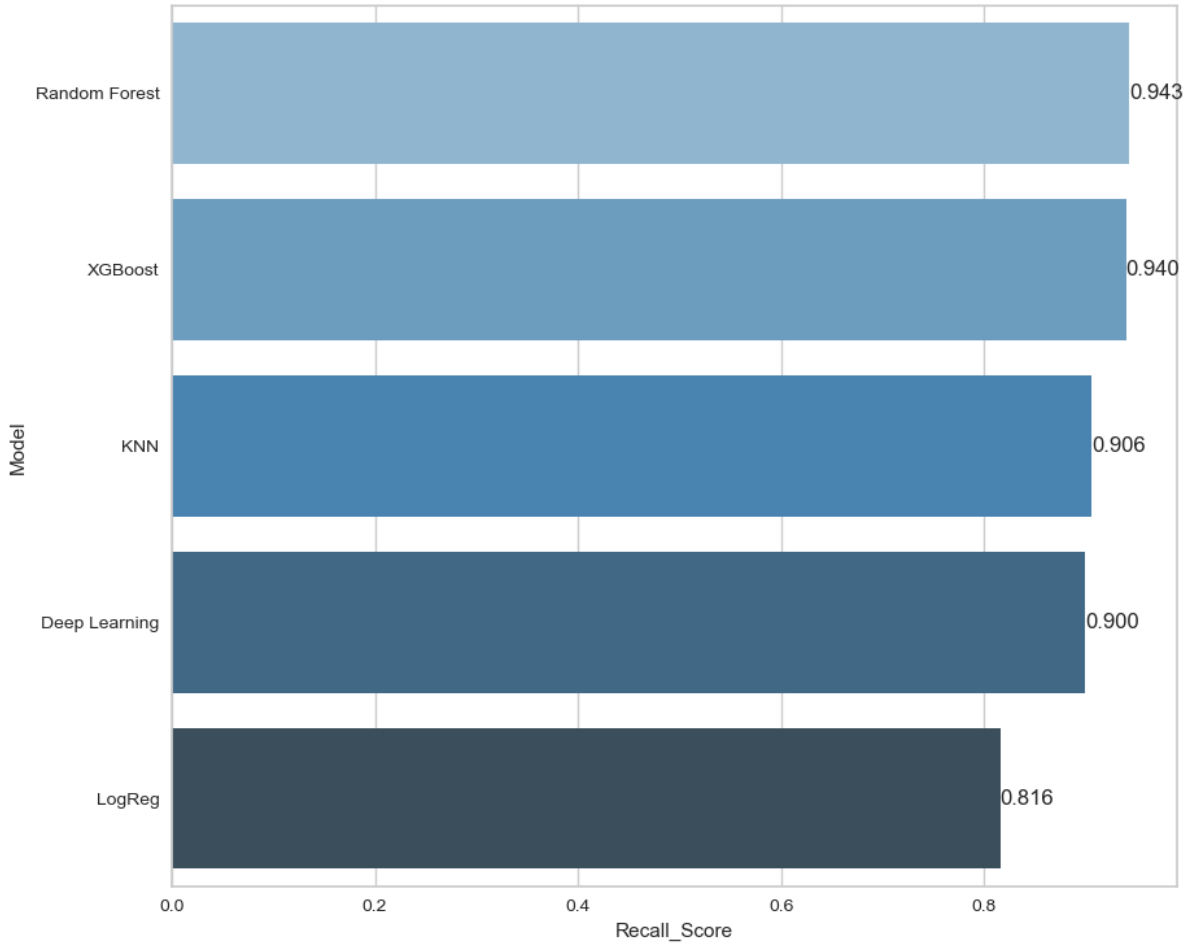

```
271/271 [==============================] - 1s 2ms/step
57/57 [==============================] - 0s 670us/step
Test_Set
[[1391  109]
 [  20  279]]
              precision    recall  f1-score   support

           0       0.99      0.93      0.96      1500
           1       0.72      0.93      0.81       299

    accuracy                           0.93      1799
   macro avg       0.85      0.93      0.88      1799
weighted avg       0.94      0.93      0.93      1799


Train_Set
[[6703  522]
 [ 106 1332]]
              precision    recall  f1-score   support

           0       0.98      0.93      0.96      7225
           1       0.72      0.93      0.81      1438

    accuracy                           0.93      8663
   macro avg       0.85      0.93      0.88      8663
weighted avg       0.94      0.93      0.93      8663
```

We used - without class weight - for better results!

Evaluating Models Performance

We will compare model performances according to metrics you choose for the problem

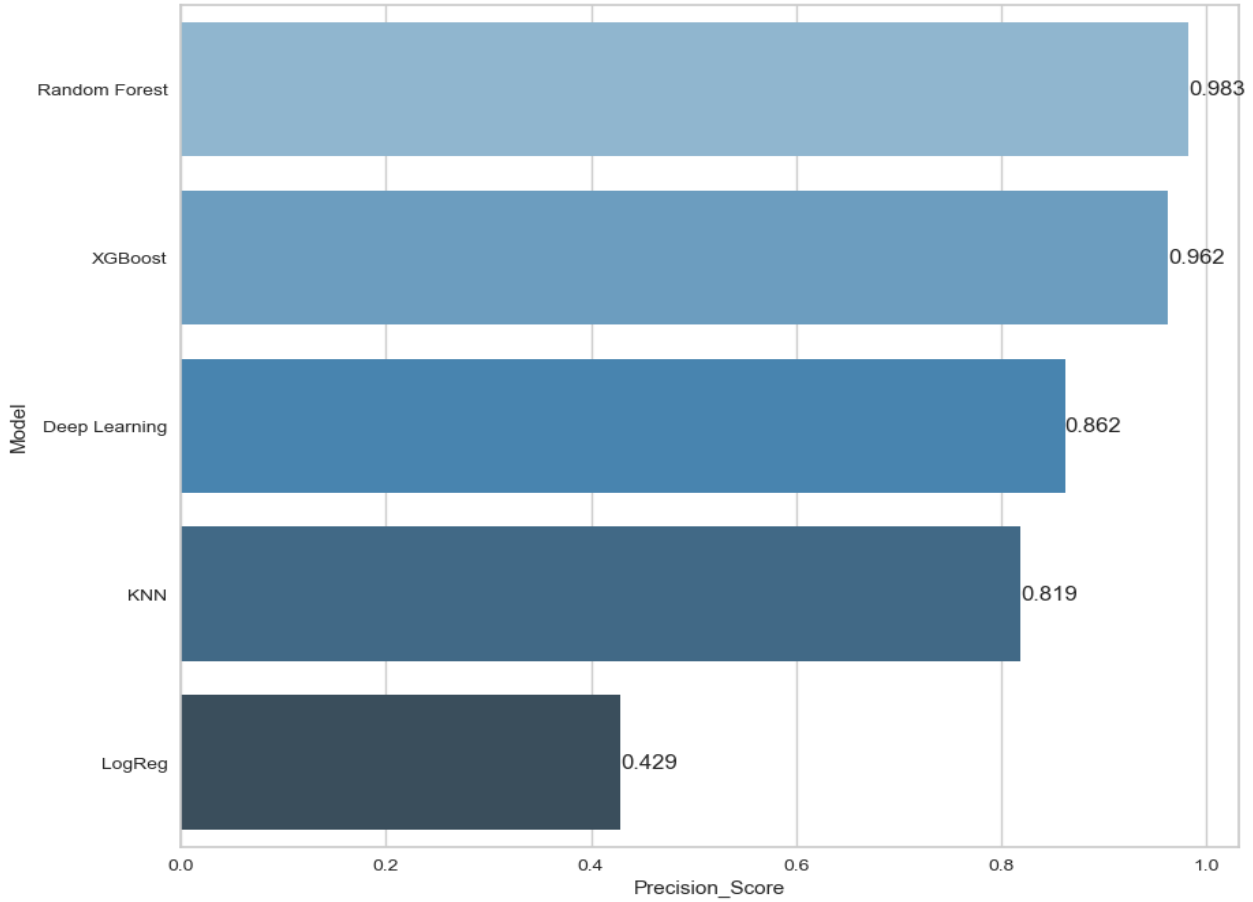
Recall score



Evaluating Models Performance

We will compare model performances according to metrics you choose for the problem

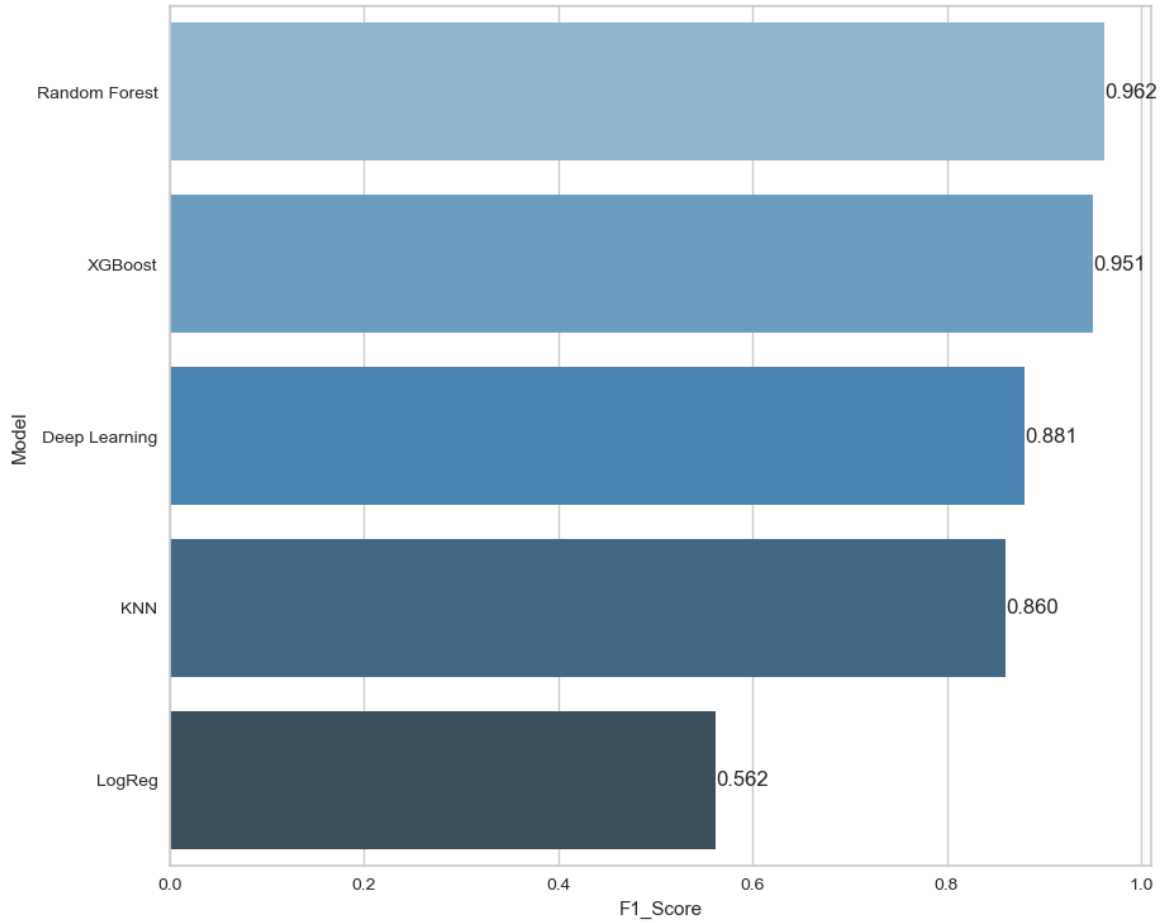
Precision score



Evaluating Models Performance

We will compare model performances according to metrics you choose for the problem

F1 score



Evaluating Models Performance

Based on evaluating graphs we chooses Random Forest

```
RandomForestClassifier
RandomForestClassifier(max_depth=10, max_features=10, n_estimators=300,
                       random_state=101)
```

```
[[9986   14]
 [ 112 1879]]
--------------------------------------------------------
              precision    recall  f1-score   support

           0       0.99      1.00      0.99     10000
           1       0.99      0.94      0.97      1991

    accuracy                           0.99     11991
   macro avg       0.99      0.97      0.98     11991
weighted avg       0.99      0.99      0.99     11991
```

Model Deployment

Good Morning

Employee Churn

What is the employee satisfaction

0 0 100

The level : %0

What was the employee last evaluation

0 0 100

The level : %0

Time spent in the company

2

How many projects does the employee worked on ?

2

Does an employee has a work accident ?

- (•) Yes
- ( ) No

Did the employee had a promotion in the last 5 years ?

- (•) Yes
- ( ) No

Select the Departments of the employee

sales

Select Salary level of employee

high

The average monthly hours

96

prediction

Conclusion

The main reasons for employees churn to leave the company are:

- Their salaries on average are low
  - Group 1
    - They work on many projects
    - They work for highest hours
    - But their evaluations were higher
  - Group 2
    - Have low evaluation
    - Have low number of projects
    - Have low working hours

In conclusion, we're addressing the challenges of both highly evaluated but overworked employees and those with poor evaluations. Prioritizing the well-being of the former and investing in the development of the latter will create a balanced, motivated workforce.
Through open communication, targeted training, and strategic task management, we aim to foster an environment that values and supports every team member, driving overall organizational success.

Thank You For Listening!

Do you have any question?