Factors Influencing Employee Retention

Griselda Arevalo

Coral Rosa-Falero

Jiemin Sheng

Bootcamp: GWU-VIRT-DATA-PT-08-2022-U-B-MW

Topic: Employee Retention Prediction

Purpose

- Build a predictive model determine the factors that influence employee retention.

Applications

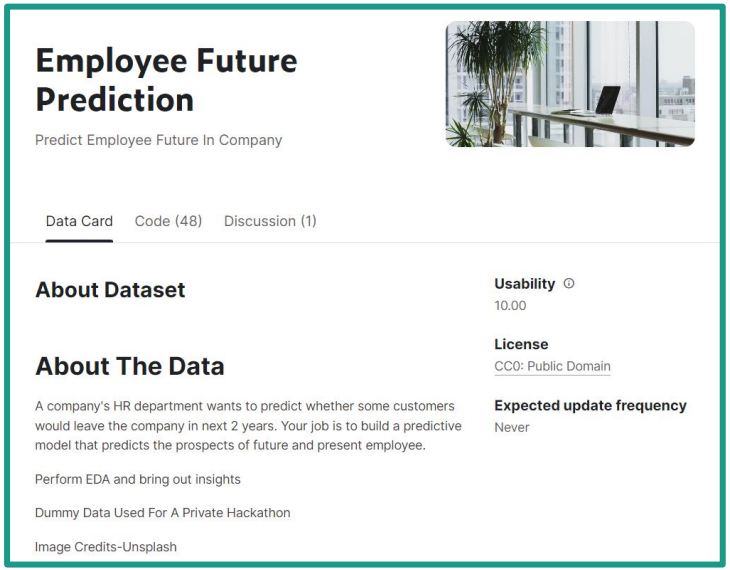
- Improve selection and retention of employees.
- Understand what factors that influence employee retention

Tools

- Jupyter NB
  - pandas
  - sqlite3
  - Sklearn.metrics
- Tableau
- GitHub

Data: Overview

csv file with 4,653 observations



Employee Future Prediction

Predict Employee Future In Company

Data Card | Code (48) | Discussion (1)

About Dataset

About The Data

A company's HR department wants to predict whether some customers would leave the company in next 2 years. Your job is to build a predictive model that predicts the prospects of future and present employee.

Perform EDA and bring out insights

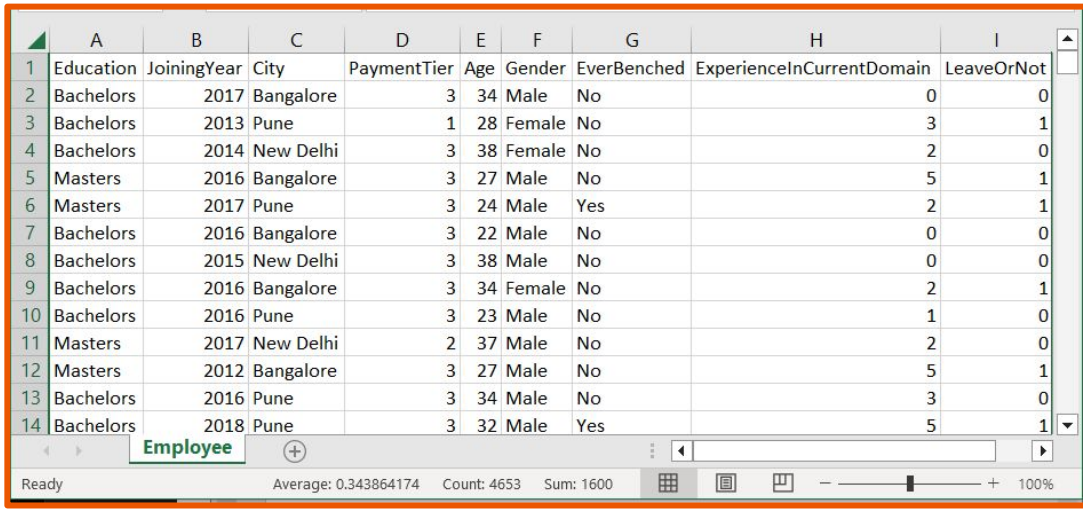
Dummy Data Used For A Private Hackathon

Image Credits-Unsplash

Usability 10.00

License CC0: Public Domain

Expected update frequency Never



	A	B	C	D	E	F	G	H	I
1	Education	JoiningYear	City	PaymentTier	Age	Gender	EverBenched	ExperienceInCurrentDomain	LeaveOrNot
2	Bachelors	2017	Bangalore	3	34	Male	No	0	0
3	Bachelors	2013	Pune	1	28	Female	No	3	1
4	Bachelors	2014	New Delhi	3	38	Female	No	2	0
5	Masters	2016	Bangalore	3	27	Male	No	5	1
6	Masters	2017	Pune	3	24	Male	Yes	2	1
7	Bachelors	2016	Bangalore	3	22	Male	No	0	0
8	Bachelors	2015	New Delhi	3	38	Male	No	0	0
9	Bachelors	2016	Bangalore	3	34	Female	No	2	1
10	Bachelors	2016	Pune	3	23	Male	No	1	0
11	Masters	2017	New Delhi	2	37	Male	No	2	0
12	Masters	2012	Bangalore	3	27	Male	No	5	1
13	Bachelors	2016	Pune	3	34	Male	No	3	0
14	Bachelors	2018	Pune	3	32	Male	Yes	5	1

Employee

Ready Average: 0.343864174 Count: 4653 Sum: 1600 100%

https://www.kaggle.com/datasets/tejashvi14/employee-future-prediction

Data

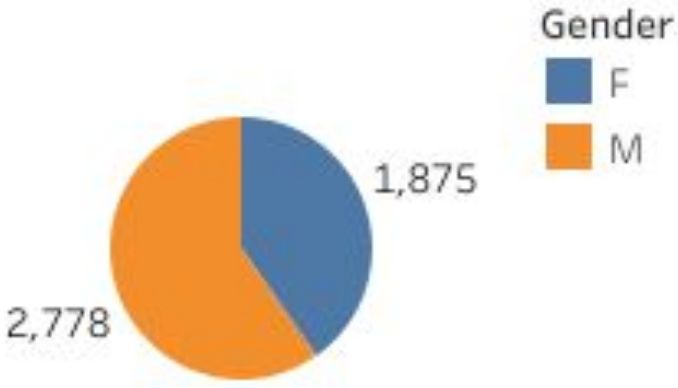
- Education (Degrees)
- Joining Year (Hiring Date)
- City (Office Location)
- Payment Tier (Salary Level)
- Age
- Gender
- Ever Benched (Productivity/Profitability)
- Experience In Current Domain (# Years)
- Leave Or Not (Retention)

```
employee_data_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4653 entries, 0 to 4652
Data columns (total 9 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Education                  4653 non-null   object
 1   JoiningYear                4653 non-null   int64
 2   City                       4653 non-null   object
 3   PaymentTier                4653 non-null   int64
 4   Age                        4653 non-null   int64
 5   Gender                     4653 non-null   object
 6   EverBenched                4653 non-null   object
 7   ExperienceInCurrentDomain  4653 non-null   int64
 8   LeaveOrNot                 4653 non-null   int64
dtypes: int64(5), object(4)
memory usage: 327.3+ KB
```

Data: Variables

Gender



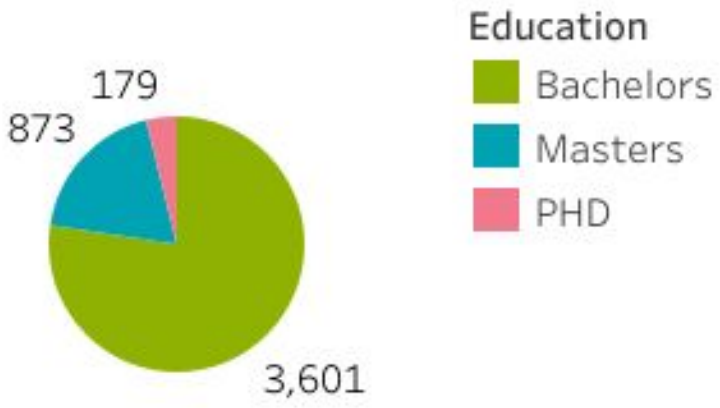
Age

Age	Gender F	Gender M
22	19	30
23	20	28
24	156	229
25	171	247
26	243	402
27	254	371
28	274	356
29	93	137
30	83	137
31	44	81
32	47	85
33	48	76
34	55	81
35	51	72
36	63	76
37	57	84
38	53	83
39	61	70
40	57	77
41	26	56

Data: Variables

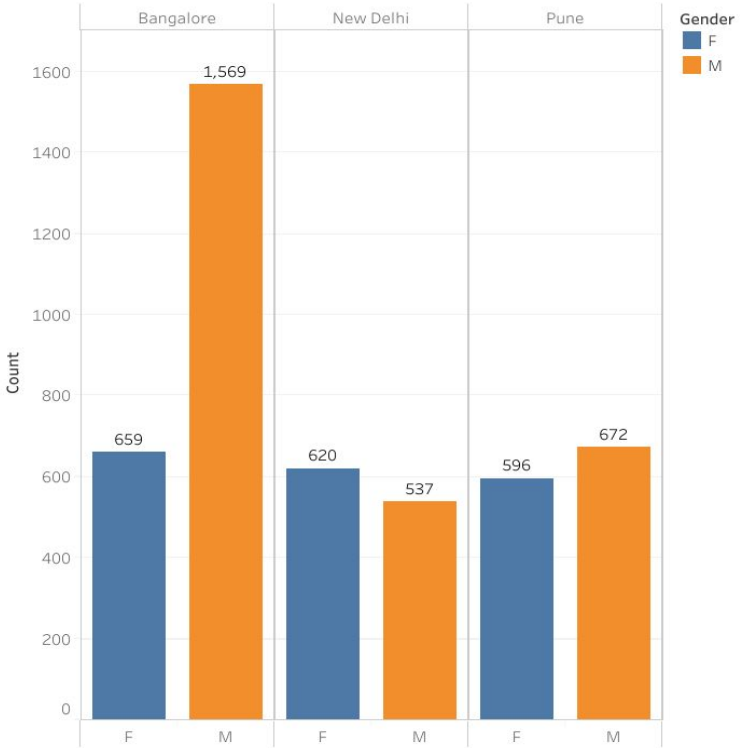
Education

Degrees



City

Office location



Data: Variables

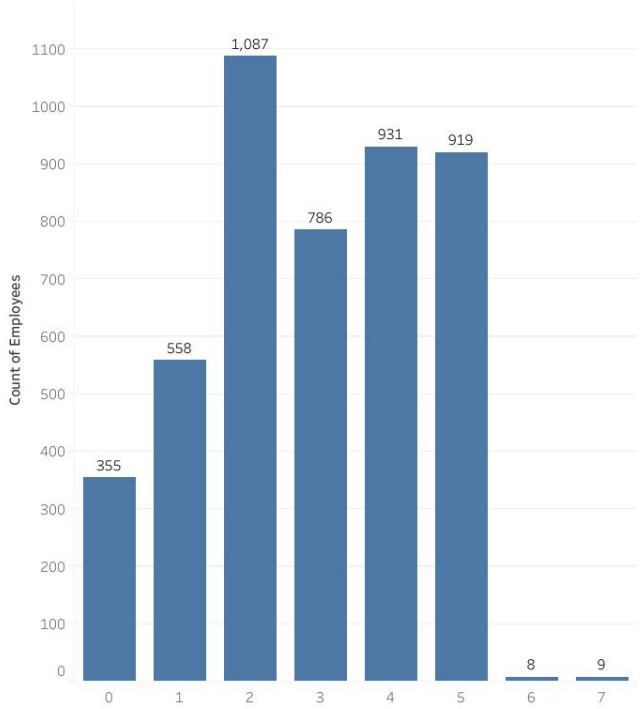
Join Year

Hiring Date

Join Year Distribution

Joining Year	Gender F	Gender M
2012	180	324
2013	253	416
2014	246	453
2015	440	341
2016	178	347
2017	450	658
2018	128	239

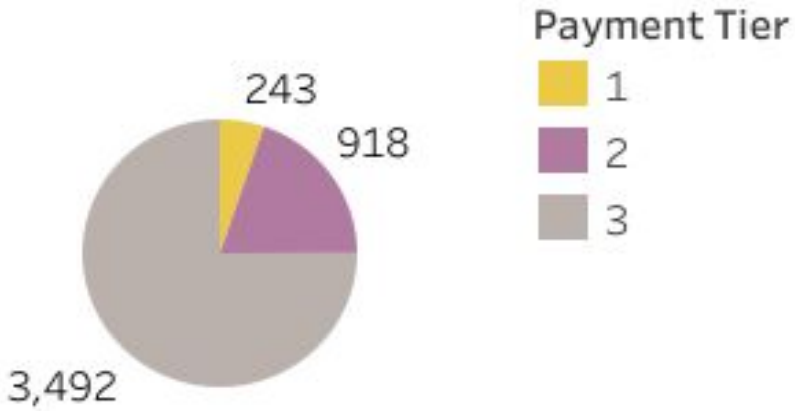
Experience In Current Domain



Data: Variables

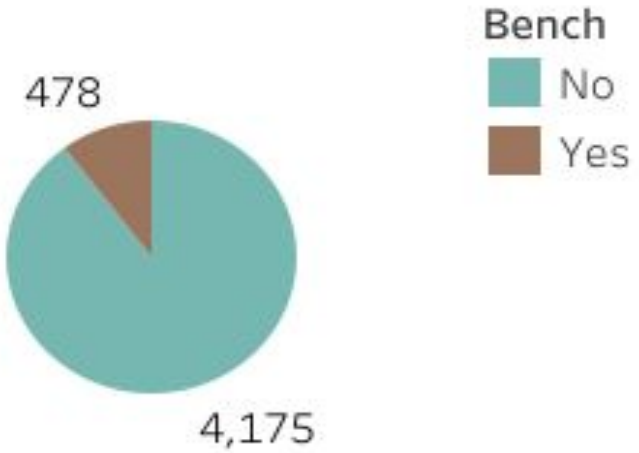
Payment Tier

Salary



Ever Benched

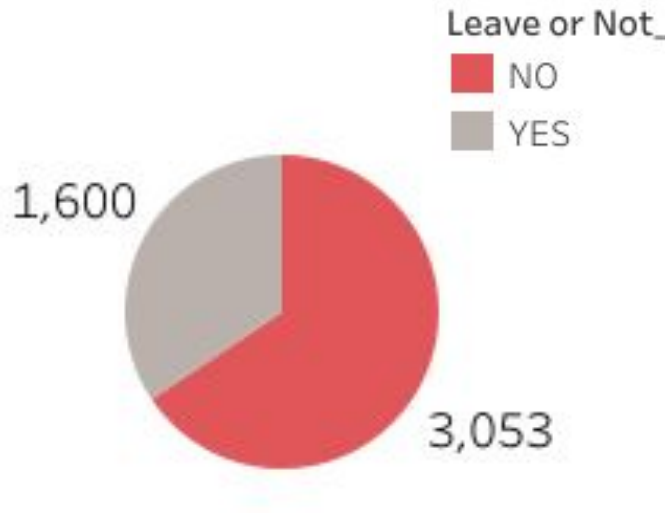
Productivity/Profitability

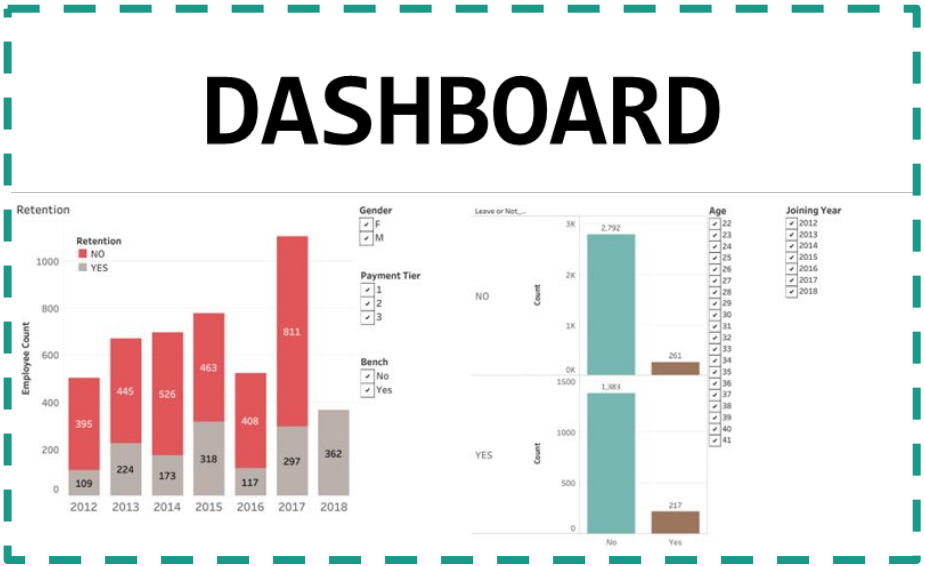


Retention Dashboard

Leave Or Not

Retention





Overview of Analysis

- **Connect to sqlite3 database**
- **Asses the properties of the database**
  - Establish columns/target and remove unwanted columns
  - Check the balance of the target values
  - Establish the training set and train the model
  - Resample the training data
  - Calculate training accuracy score
  - Calculate testing set accuracy score
  - Calculate confusion matrix
  - Evaluate the classification report
  - Determine the importance rating of each feature

```
employee_data_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4653 entries, 0 to 4652
Data columns (total 9 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Education                  4653 non-null   object
 1   JoiningYear                4653 non-null   int64
 2   City                       4653 non-null   object
 3   PaymentTier                4653 non-null   int64
 4   Age                        4653 non-null   int64
 5   Gender                     4653 non-null   object
 6   EverBenched                4653 non-null   object
 7   ExperienceInCurrentDomain  4653 non-null   int64
 8   LeaveOrNot                 4653 non-null   int64
dtypes: int64(5), object(4)
memory usage: 327.3+ KB
```

Overview of Analysis

- Connect to sqlite3 database
- Asses the properties of the database
- Establish columns/target and remove unwanted columns
- **Check the balance of the target values**
- **Establish the training set and train the model**
- Calculate training accuracy score
- Calculate testing set accuracy score
- Calculate confusion matrix
- Evaluate the classification report
- Determine the importance rating of each feature

```
# Check the balance of our target values (1 = yes or 0 = no)
y.value_counts()
```

```
0    3053
1    1600
Name: LeaveOrNot, dtype: int64
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y,
                                                    random_state=1,
                                                    stratify=y)
Counter(y_train)
```

```
Counter({1: 1200, 0: 2289})
```

```
X_train.shape
```

```
(3489, 14)
```

Overview of Analysis

- Connect to sqlite3 database
- Asses the properties of the database
- Establish columns/target and remove unwanted columns
- Check the balance of the target values
- Establish the training set and train the model
- **Calculate training accuracy score**
- **Calculate testing set accuracy score**
- **Calculate confusion matrix**
- **Evaluate the classification report**
- Determine the importance rating of each feature

```
# Calculated the balanced accuracy score: TRAINING
y_tr = brfc.predict(X_train)
balanced_accuracy_score(y_train, y_tr)
```

```
0.9137117737003058
```

```
# Calculated the balanced accuracy score: TESTING
y_pred = brfc.predict(X_test)
balanced_accuracy_score(y_test, y_pred)
```

```
0.7848625654450262
```

```
# Calculate confusion matrix.
cm = confusion_matrix(y_test, y_pred)
cm
```

```
array([[632, 132],
       [103, 297]], dtype=int64)
```

```
# classification report
from sklearn.metrics import classification_report
print( classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.86      0.83      0.84       764
           1       0.69      0.74      0.72       400

    accuracy                           0.80      1164
   macro avg       0.78      0.78      0.78      1164
weighted avg       0.80      0.80      0.80      1164
```

Overview of Analysis

- Connect to sqlite3 database
- Asses the properties of the database
- Establish columns/target and remove unwanted columns
- Check the balance of the target values
- Establish the training set and train the model
- Calculate training accuracy score
- Calculate testing set accuracy score
- Calculate confusion matrix
- Evaluate the classification report
- **Determine the importance rating of each feature**

```
# List the features sorted in descending order by feature importance
importances = brfc.feature_importances_
sorted(zip(brfc.feature_importances_, X.columns), reverse=True)
```

```
[(0.3086914210245113, 'JoiningYear'),
 (0.19190147118381745, 'Age'),
 (0.09890156348096112, 'ExperienceInCurrentDomain'),
 (0.09325636092904764, 'PaymentTier'),
 (0.05727410779116132, 'City_Pune'),
 (0.054031392148976856, 'Education_Masters'),
 (0.046093613205681915, 'Gender_Male'),
 (0.03876883320695893, 'Education_Bachelors'),
 (0.03204738082388843, 'Gender_Female'),
 (0.025990119766903574, 'City_Bangalore'),
 (0.025687810889074857, 'City_New Delhi'),
 (0.009464044355872172, 'EverBenched_No'),
 (0.009311635638471467, 'EverBenched_Yes'),
 (0.008580245554672804, 'Education_PHD')]
```

Summary

Accuracy scores

- Training Score: 91%
- Testing Score: 78%

Precision, Sensitivity and F1 score of 80%.

Next Steps

Anything the team would have done differently

- Use a more robust database

Recommendation for future analysis

- Adjustments to current model:
  - Modify the split size
  - Number of decision making trees
  - Remove the less important features