Problem Statement and Data Description

Why are our best and most experienced employees leaving prematurely?

Have fun with this database and try to predict which valuable employees will leave next.

Fields in the dataset include:

1) Satisfaction Level
2) Last evaluation
3) Number of projects
4) Average monthly hours
5) Time spent at the company
6) Whether they have had a work accident
7) Whether they have had a promotion in the last 5 years
8) Departments (column sales)
9) Salary
10) Whether the employee has left

Data set information

RangeIndex: 14999 entries, 0 to 14998

Data columns (total 10 columns):

- satisfaction_level 14999 non-null float64
- last_evaluation 14999 non-null float64
- number_project 14999 non-null int64
- average_montly_hours 14999 non-null int64
- time_spend_company 14999 non-null int64
- Work_accident 14999 non-null int64
- left 14999 non-null int64
- promotion_last_5years 14999 non-null int64
- sales 14999 non-null object
- salary 14999 non-null object

dtypes: float64(2), int64(6), object(2)

Human Resource Analytics

Table

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5years	sales	salary
0	0.38	0.53	2	157	3	0	1	0	sales	low
1	0.80	0.86	5	262	6	0	1	0	sales	medium
2	0.11	0.88	7	272	4	0	1	0	sales	medium
3	0.72	0.87	5	223	5	0	1	0	sales	low
4	0.37	0.52	2	159	3	0	1	0	sales	low

sales and salary are categorical and let's see if "Work_accident" and "promotion_last_5years" are categorical or not

- `Work_accident [0 1]`
- `left [1 0]`
- `promotion_last_5years [0 1]`
- `sales ['sales' 'accounting' 'hr' 'technical' 'support' 'management' 'IT'`
- `'product_mng' 'marketing' 'RandD']`
- `salary ['low' 'medium' 'high']`

Clearly Work_accident and promotion_last_5years are categorical variables.

- Sales: contains 10 unique features
- Salary: contains 3 unique features

Let's see if there is any null value in the dataset

satisfaction_level 0

last_evaluation 0

number_project 0

average_montly_hours 0

time_spend_company 0

Work_accident 0

left 0

promotion_last_5years 0

sales 0

salary 0

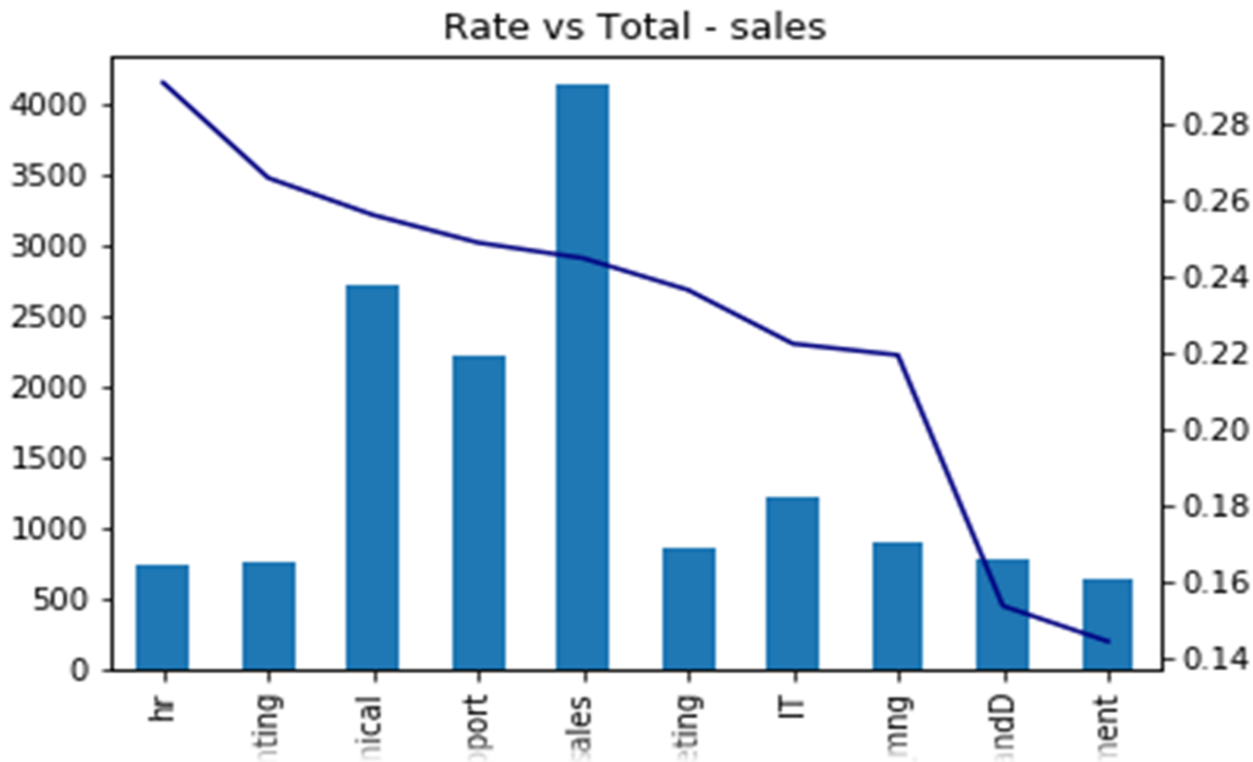
dtype: int64

There are no null values in dataset. In this dataset we don't need to impute any null values.

Exploratory Data Analysis : Categorical Variables

- **Sales:**

sales	left	not_left	Rate	total
hr	215	524	0.290934	739
accounting	204	563	0.265971	767
technical	697	2023	0.256250	2720
support	555	1674	0.248991	2229
sales	1014	3126	0.244928	4140
marketing	203	655	0.236597	858
IT	273	954	0.222494	1227
product_mng	198	704	0.219512	902
RandD	121	666	0.153748	787
management	91	539	0.144444	630

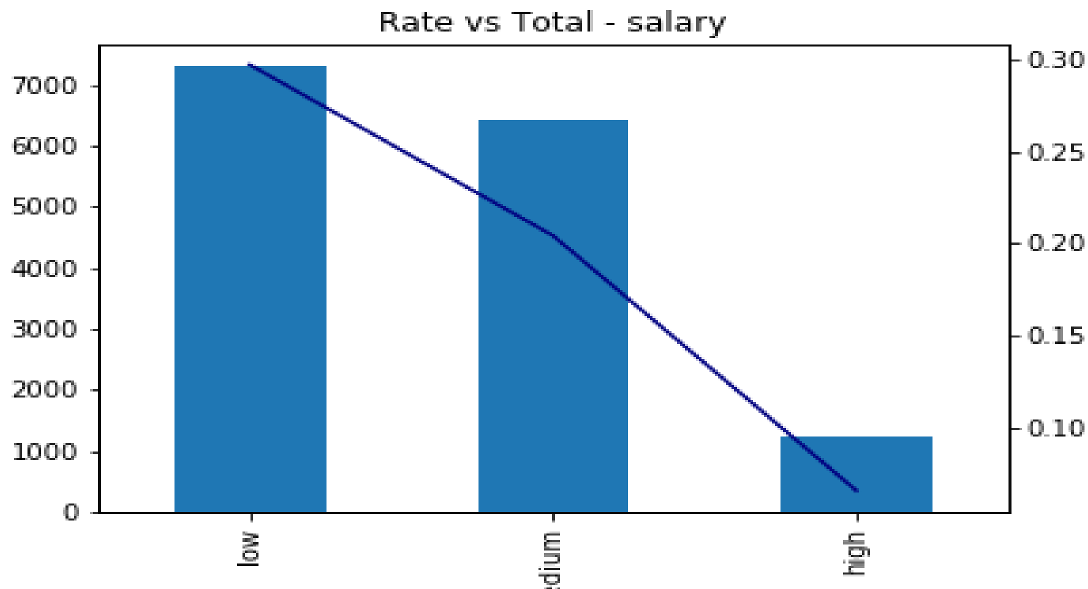


- People who are in HR and accounting departments are more likely to leave the company

- People in RandD and management department are less likely to leave

- **Salary**

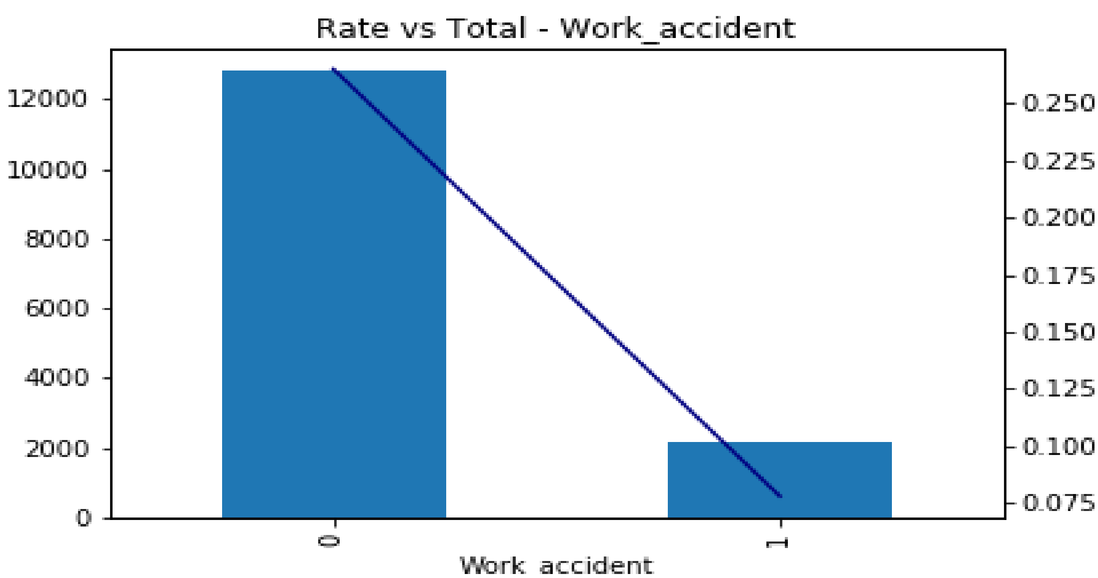
	left	not_left	Rate	total
Salary				
low	2172	5144	0.296884	7316
medium	1317	5129	0.204313	6446
high	82	1155	0.066289	1237



- People who have low salary are more likely to leave than medium and high salary.
- As obvious people with higher salary are very less likely to leave the company

- **Work_accident**

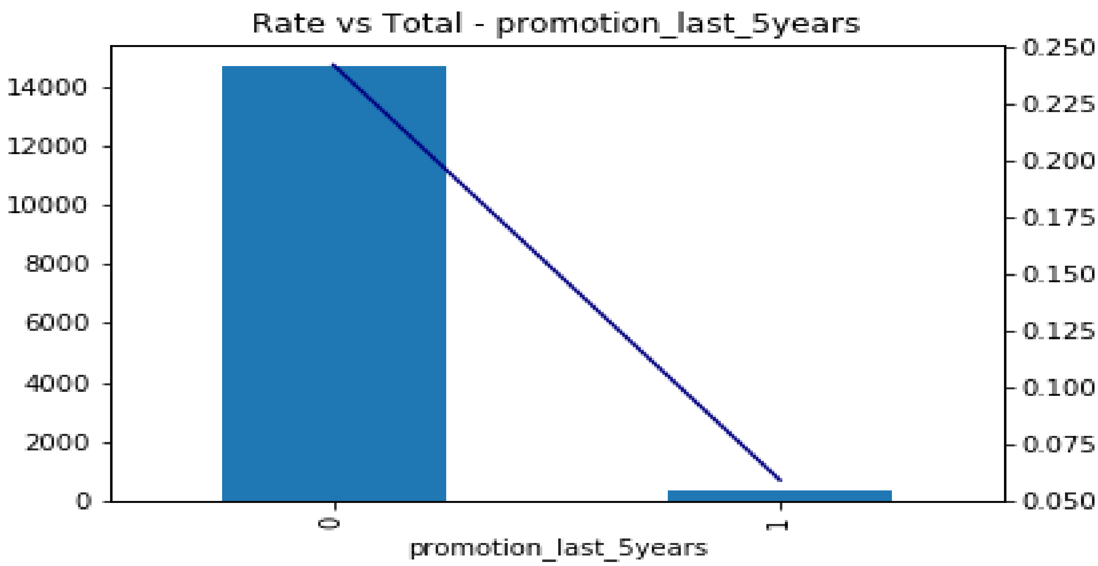
	left	not_left	rate	total
Work_accident				
0	3402	9428	0.265160	12830
1	169	2000	0.077916	2169



- In this case people who didn't had any accidents are more likely to leave.

- **promotion_last_5years**

	left	not_left	rate	total
promotion_last_5years				
0	3552	11128	0.241962	14680
1	19	300	0.059561	319

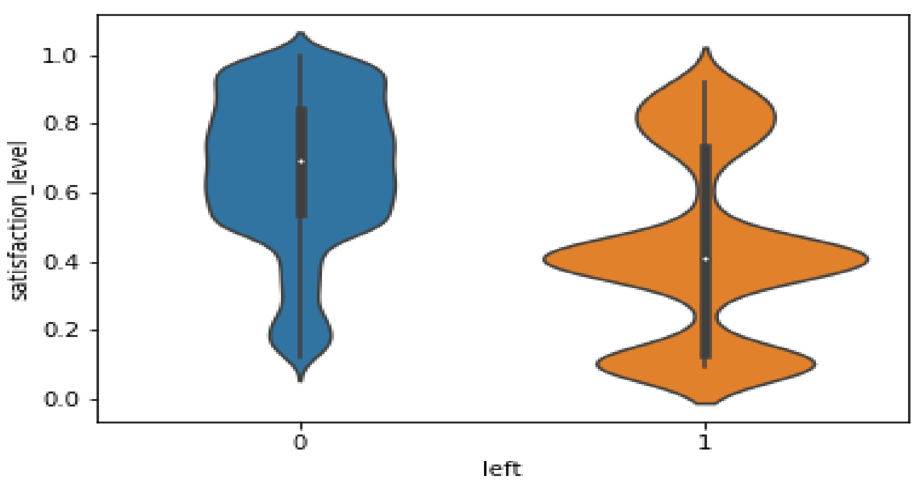


- As obvious People who didn't had promotion in last 5 years are very much likely to leave.

Exploratory Data Analysis -- Numerical variable

- satisfaction_level

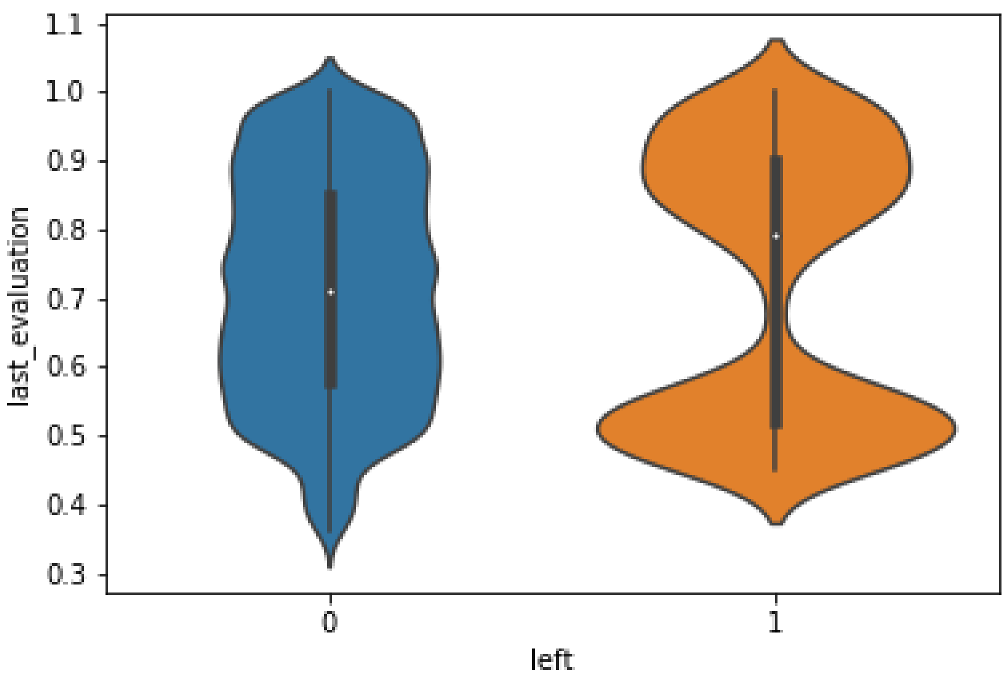
	satisfaction_level_x	satisfaction_level_y
count	3571.000000	11428.000000
mean	0.440098	0.666810
std	0.263933	0.217104
min	0.090000	0.120000
25%	0.130000	0.540000
50%	0.410000	0.690000
75%	0.730000	0.840000
max	0.920000	1.000000



As per above table and violin distribution graph we can see a obvious thing as people who left were not much satisfied. Most of the people who stay (75%) had satisfaction level more than 0.5. Whereas people who left are categorized into three groups. First who had very less satisfaction level. Second group had satisfaction level around 0.4 and last group had around 0.9

• last_evaluation

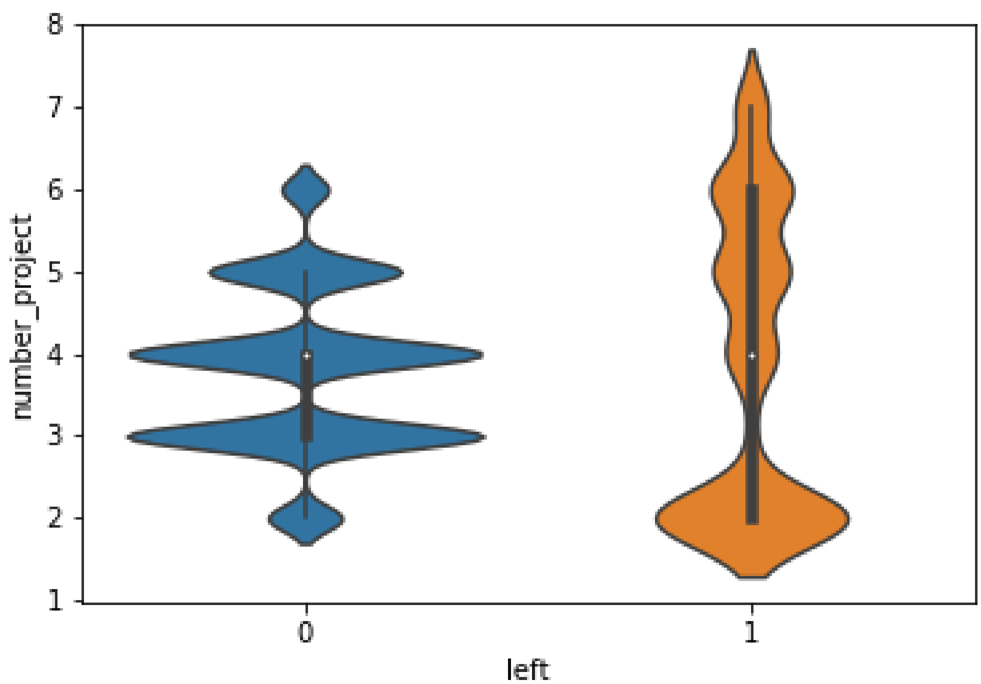
	last_evaluation_x	last_evaluation_y
• count	3571.000000	11428.000000
• mean	0.718113	0.715473
• std	0.197673	0.162005
• min	0.450000	0.360000
• 25%	0.520000	0.580000
• 50%	0.790000	0.710000
• 75%	0.900000	0.850000
• max	1.000000	1.000000



Distribution for people who left shows bi-modal graph. Here some people had very low evaluation, so of course they left. Few people had higher evaluation but still left.

• number_project

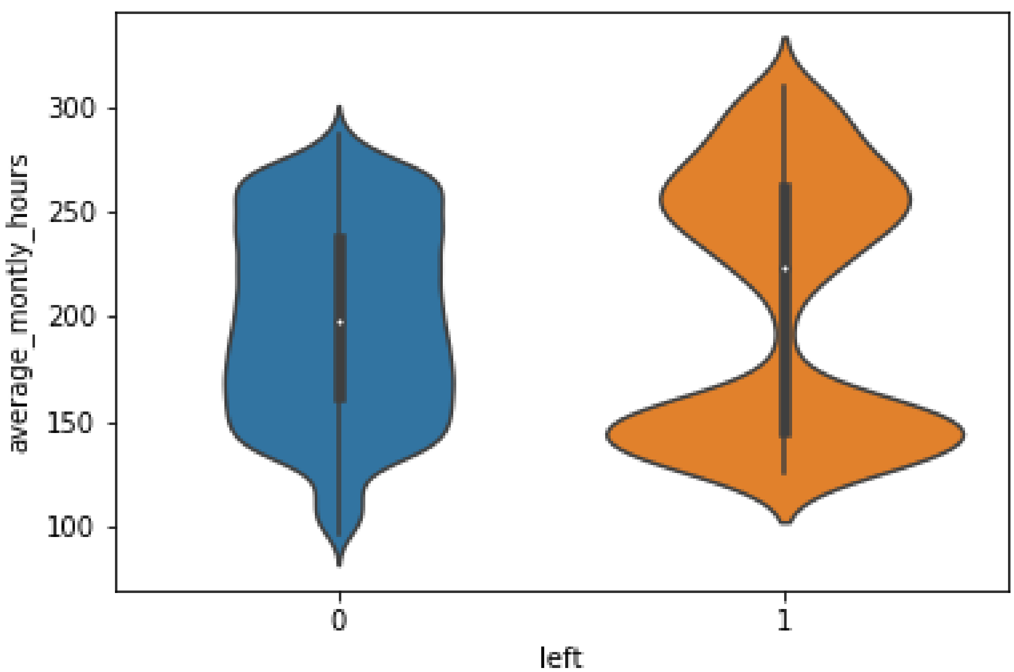
	number_project_x	number_project_y
count	3571.000000	11428.000000
mean	3.855503	3.786664
std	1.818165	0.979884
min	2.000000	2.000000
25%	2.000000	3.000000
50%	4.000000	4.000000
75%	6.000000	4.000000
max	7.000000	6.000000



People who had very less projects left and even people who had many projects left due to over pressure.

- ## average_montly_hours

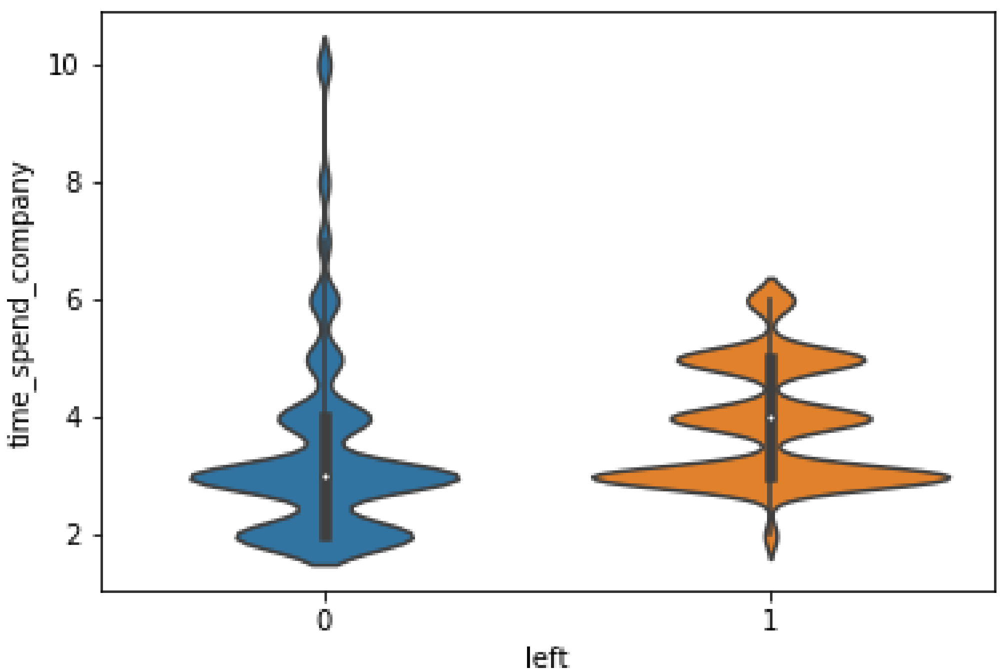
	average_montly_hours_x	average_montly_hours_y
count	3571.000000	11428.000000
mean	207.419210	199.060203
std	61.202825	45.682731
min	126.000000	96.000000
25%	146.000000	162.000000
50%	224.000000	198.000000
75%	262.000000	238.000000
max	310.000000	287.000000



Some people were working very hard and spending extra monthly hours left due to work pressure.

- **time_spend_company**

	time_spend_company_x	time_spend_company_y
count	3571.000000	11428.000000
mean	3.876505	3.380032
std	0.977698	1.562348
min	2.000000	2.000000
25%	3.000000	2.000000
50%	4.000000	3.000000
75%	5.000000	4.000000
max	6.000000	10.000000



Clearly who worked less than 4 years were more likely to leave

Building a Model

- We have **Sale** and **Salary** columns that contain textual categorical variable.
- So we use **LabelEncoder** and **OneHotEncode**r from **preprocessing** module of **sklearn** library.
- Remember to avoid **dummy variable trap** by removing one column.
- Hence we now have 18 columns in comparison to 10 columns that we had in our dataset earlier.
- Now we split data our data into **x_train,x_test,y_train, y_test** using **train_test_split** from **sklearn.model_selection.**
- As our main goal is to predict that who will leave the company and who will not ,so for that we need an appropriate classifier and hence we use **RandomForestClassifier** for this.
- Now to tune our hyper parameter we use **GridSearchCV** from **sklearn.model_selection**.
- Then we use best parameter given by **GridSeachCV** to train our model via R**andomForestClassifier.**
- And once trained then we use it to predict value for **x_test**
- Now we check how well our model performed by looking at **accuracy score**, **confusion matrix** and **f1 score**.

sklearn.ensemble.RandomForestClassifier

class sklearn.ensemble.**RandomForestClassifier**(*n_estimators=10*, *criterion='gini'*, *max_depth=None*, *min_samples_split=2*, *min_samples_leaf=1*, *min_weight_fraction_leaf=0.0*, *max_features='auto'*, *max_leaf_nodes=None*, *min_impurity_decrease=0.0*, *min_impurity_split=None*, *bootstrap=True*, *oob_score=False*, *n_jobs=1*, *random_state=None*, *verbose=0*, *warm_start=False*, *class_weight=None*)

n_estimators=[10,100,500]

max_depth=[None,6,8]

max_features=[2,5,12,18]

min_samples_leaf=[1,3,5]

rfc_param={'criterion':['gini','entropy'],'n_estimators':n_estimators,'max_depth':max_depth
,'max_features':max_features,'min_samples_leaf':min_samples_leaf}

Now best score and best parameters returned by Grid Search are:

```
rfc_best_score 0.989582465205
rfc_best_param {'min_samples_leaf': 1,
'max_depth': None, 'n_estimators': 500,
'max_features': 12, 'criterion': 'entropy'}
```

Model evaluation

Random Forest Classifier **Confusion Matrix**:

	Left	Stayed
Left	2294	5
Stayed	17	684

Accuracy Score:

0.992666666667

F1_Score:

0.984172661871

Created by:

Anubhav Shukla

http://www.github.com/anushuk