ZecaiLiang_DataChallenge1_EmployeeRetention

February 21, 2019

0.0.1 Requirements

- python 3.6.2

```
conda install -c sebp scikit-survival
conda install -c anaconda seaborn
```

- numpy 1.15.4
- pandas 0.20.3
- scikit-learn 0.19.2
- scikit-survival 0.6.0
- seaborn 0.9.0

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
        %matplotlib inline
```

0.1 1. load data

```
In [2]: df = pd.read_csv("employee_retention_data.csv")
```

Data Example

- employee_id : id of the employee. Unique by employee per company
- company_id : company id.
- dept : employee dept
- seniority : number of yrs of work experience when hired
- salary: avg yearly salary of the employee during her tenure within the company
- join_date: when the employee joined the company, it can only be between 2011/01/24 and 2015/12/13
- quit_date: when the employee left her job (if she is still employed as of 2015/12/13, this field is NA)

```
In [3]: df.head()
```

```
Out[3]:    employee_id  company_id             dept  seniority    salary   join_date  \
        0      13021.0           7  customer_service         28   89000.0  2014-03-24
        1     825355.0           7        marketing         20  183000.0  2013-04-29
        2     927315.0           4        marketing         14  101000.0  2014-10-13
        3     662910.0           7  customer_service         20  115000.0  2012-05-14
        4     256971.0           2     data_science         23  276000.0  2011-10-17

           quit_date
        0  2015-10-30
        1  2014-04-04
        2         NaN
        3  2013-06-07
        4  2014-08-22
```

Check Missing Data

- no missing data
- NAN in join_date: still employed by 2015/12/13

```
In [14]: df.isnull().sum()
```

```
Out[14]: employee_id        0
         company_id         0
         dept               0
         seniority          0
         salary             0
         join_date          0
         quit_date      11192
         dtype: int64
```

Data Size

```
In [15]: df.shape
```

```
Out[15]: (24702, 7)
```

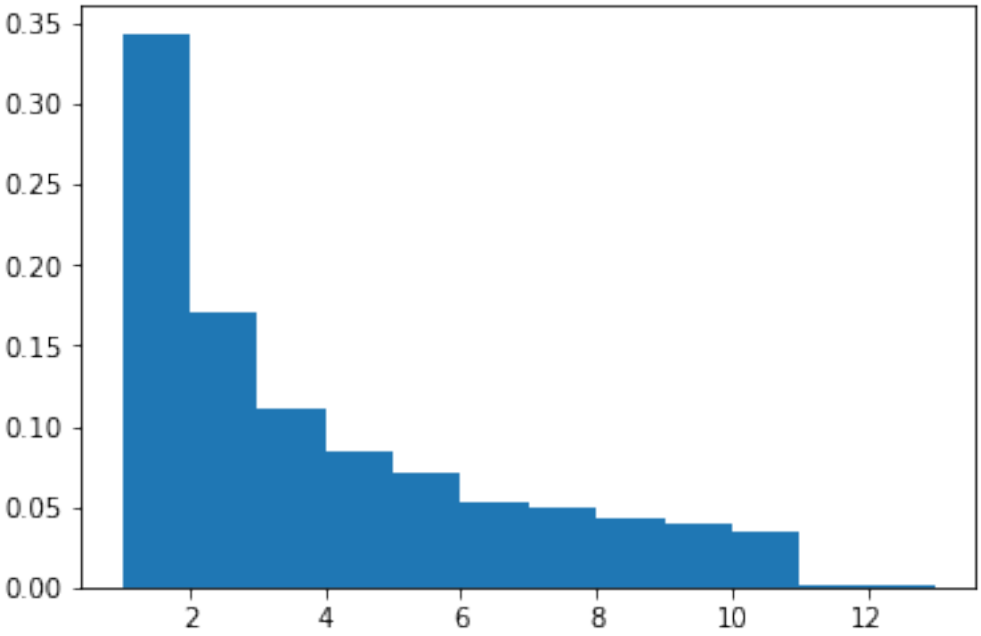
0.2 2. Exploratory Analysis: distribution

Distribution of company

- 34% data from one company

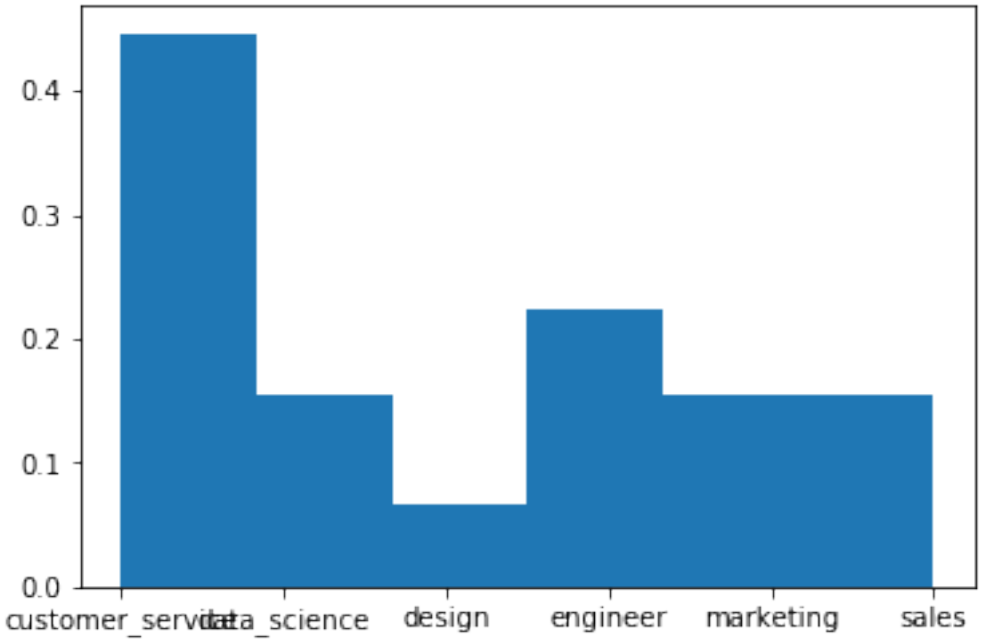
```
In [29]: plt.hist(df["company_id"], density = True, bins = range(1,14))
```

```
Out[29]: (array([0.34353494, 0.17091733, 0.11128654, 0.08347502, 0.07104688,
                 0.05226297, 0.04955064, 0.04238523, 0.03890373, 0.03501741,
                 0.00064772, 0.00097158]),
          array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13]),
          <a list of 12 Patch objects>)
```



Distribution of company

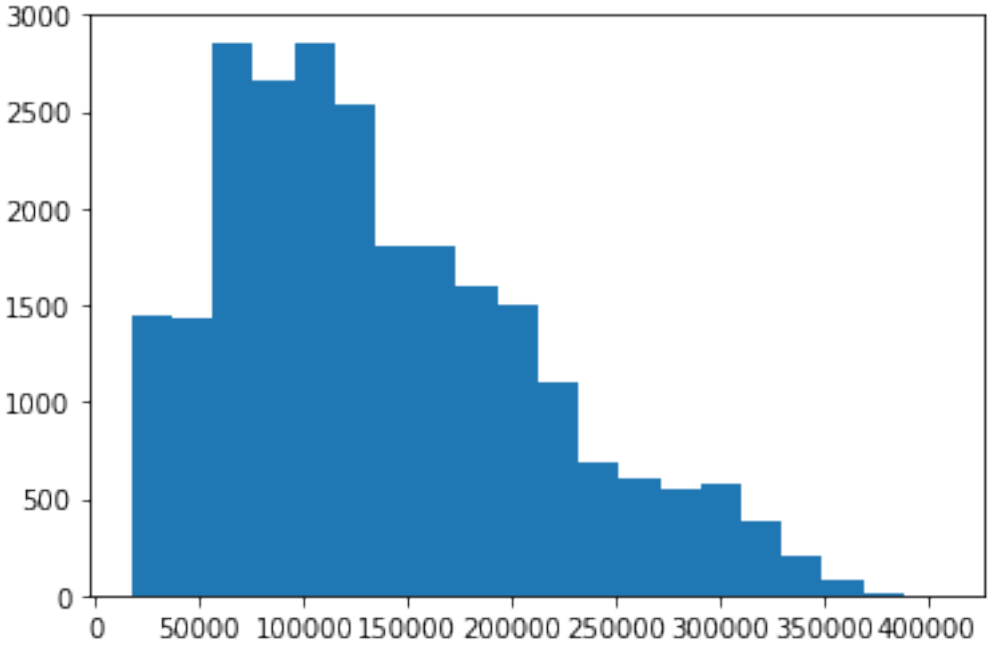
```
In [43]: plt.hist(df["dept"], density = True, bins = 6)

Out[43]: (array([0.44595579, 0.15496721, 0.06703911, 0.22409521, 0.15384989,
                 0.15409279]),
          array([0.        , 0.83333333, 1.66666667, 2.5       , 3.33333333,
                 4.16666667, 5.        ]),
          <a list of 6 Patch objects>)
```



Distribution of salary

- left skewed

```
In [33]: plt.hist(df["salary"], bins = 20)
         plt.show()
```

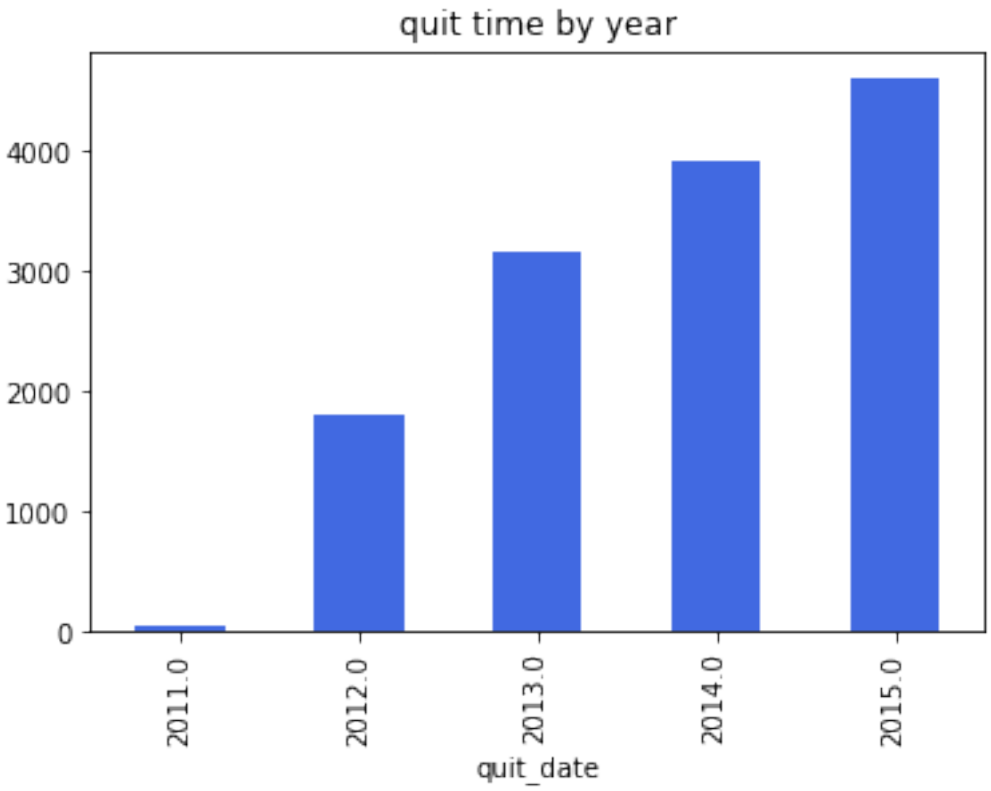


0.3 3. Exploratory Analysis: quit time trend

```
In [36]: df['quit_date'] = pd.to_datetime(df['quit_date'])
```

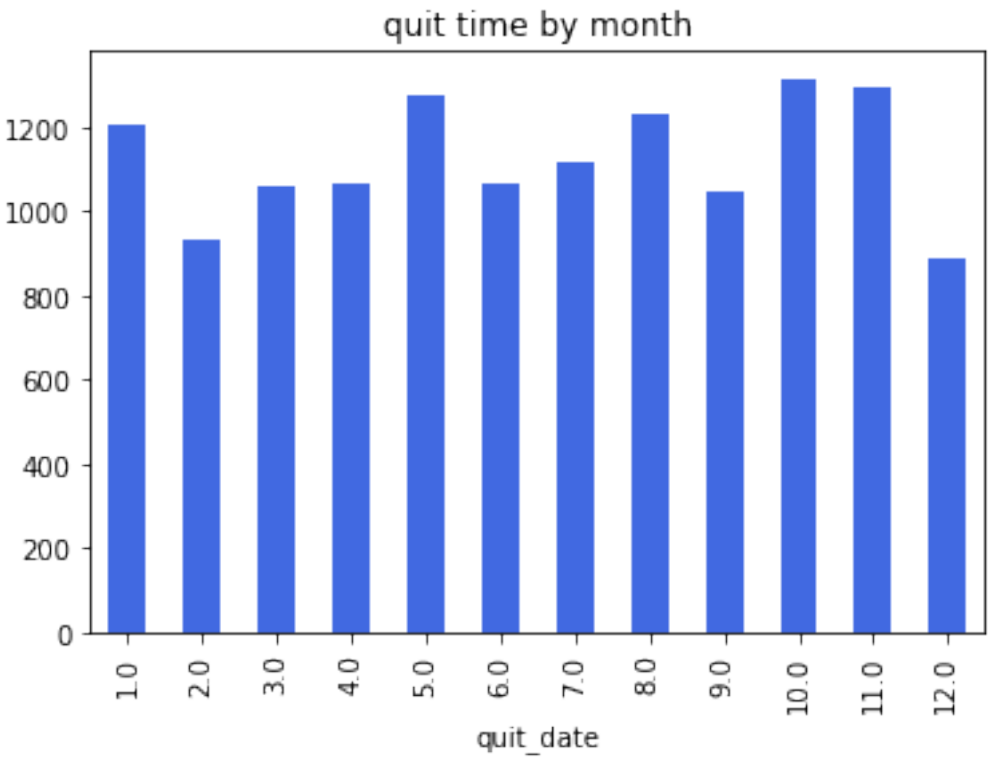
```
In [48]: df['quit_date'].groupby(df["quit_date"].dt.year).count().plot(kind="bar", color = "ro
         plt.title("quit time by year")
```

```
Out[48]: Text(0.5,1,'quit time by year')
```




```
In [49]: df['quit_date'].groupby(df["quit_date"].dt.month).count().plot(kind="bar", color = "r
         plt.title("quit time by month")
```

```
Out[49]: Text(0.5,1,'quit time by month')
```



0.4 4. survival analysis: feature engineer

df_y - "status": True if quit - survival_days: days between join and quit

```
In [129]: df["status"] = df["quit_date"].notnull()

In [130]: df['quit_date'] = pd.to_datetime(df['quit_date'])
          df['join_date'] = pd.to_datetime(df['join_date'])

In [131]: df["survival_days"] = (df["quit_date"] - df["join_date"]).dt.days

In [132]: df_y = pd.DataFrame({"status": df["status"],
                               "survival_days": df["survival_days"]})

In [133]: ### survival time for people who haven't quit
          df_y["survival_days"] = df_y["survival_days"].fillna((pd.Timestamp("2015/12/13 ") - p

In [134]: df_y.head()

Out[134]:    status  survival_days
          0    True          585.0
          1    True          340.0
          2   False         1784.0
          3    True          389.0
          4    True         1040.0
```

df_x

- numeric
- categorical

```
In [77]: df.columns
```

```
Out[77]: Index(['employee_id', 'company_id', 'dept', 'seniority', 'salary', 'join_date',
                'quit_date', 'status', 'survival_days'],
               dtype='object')
```

```
In [78]: df_x = df[['company_id', 'dept', 'seniority', 'salary']]
```

```
In [79]: df_x['company_id'] = df_x['company_id'].astype('category')
         df_x['dept'] = df_x['dept'].astype('category')
```

```
/anaconda3/envs/datachallenge/lib/python3.6/site-packages/ipykernel_launcher.py:1: SettingWithC
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html
  """Entry point for launching an IPython kernel.
/anaconda3/envs/datachallenge/lib/python3.6/site-packages/ipykernel_launcher.py:2: SettingWithC
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html
```

```
In [80]: from sksurv.preprocessing import OneHotEncoder

         df_x_numeric = OneHotEncoder().fit_transform(df_x)
         df_x_numeric.head()
```

```
Out[80]:    company_id=2  company_id=3  company_id=4  company_id=5  company_id=6  \
         0           0.0           0.0           0.0           0.0           0.0
         1           0.0           0.0           0.0           0.0           0.0
         2           0.0           0.0           1.0           0.0           0.0
         3           0.0           0.0           0.0           0.0           0.0
         4           1.0           0.0           0.0           0.0           0.0

            company_id=7  company_id=8  company_id=9  company_id=10  company_id=11  \
         0           1.0           0.0           0.0            0.0            0.0
         1           1.0           0.0           0.0            0.0            0.0
         2           0.0           0.0           0.0            0.0            0.0
         3           1.0           0.0           0.0            0.0            0.0
         4           0.0           0.0           0.0            0.0            0.0
```

```
   company_id=12  dept=data_science  dept=design  dept=engineer  \
0            0.0                0.0          0.0            0.0
1            0.0                0.0          0.0            0.0
2            0.0                0.0          0.0            0.0
3            0.0                0.0          0.0            0.0
4            0.0                1.0          0.0            0.0

   dept=marketing  dept=sales  seniority    salary
0             0.0         0.0         28   89000.0
1             1.0         0.0         20  183000.0
2             1.0         0.0         14  101000.0
3             0.0         0.0         20  115000.0
4             0.0         0.0         23  276000.0
```

0.5 5. survival analysis: linear model

Cox's proportional hazard's model

```
In [135]: df_y_array = df_y.to_records(index = False)
```

```
In [141]: from sksurv.linear_model import CoxPHSurvivalAnalysis

          estimator = CoxPHSurvivalAnalysis()
          estimator.fit(df_x_numeric, df_y_array)
```

```
Out[141]: CoxPHSurvivalAnalysis(alpha=0, n_iter=100, tol=1e-09, verbose=0)
```

accuracy

```
In [152]: estimator.score(df_x_numeric, df_y_array)
```

```
Out[152]: 0.5249647502977128
```

feature importance:

```
In [144]: pd.Series(estimator.coef_, index=df_x_numeric.columns)
```

```
Out[144]: company_id=2        -0.060092
          company_id=3        -0.007914
          company_id=4         0.008704
          company_id=5        -0.003471
          company_id=6        -0.014468
          company_id=7         0.018409
          company_id=8        -0.002403
          company_id=9        -0.023196
          company_id=10       -0.007008
          company_id=11        0.497423
          company_id=12       -0.082669
          dept=data_science    0.068453
```

```
dept=design          0.090938
dept=engineer        0.027032
dept=marketing       0.092390
dept=sales           0.131716
seniority            0.007566
salary              -0.000001
dtype: float64
```