In [1]:
```
import pandas as pd
```

In [3]:
```
emp_data = pd.read_csv('Data/HR_comma_sep.csv.txt')
```

In [4]:
```
emp_data.head()
```

Out[4]:

	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

In [6]:
```
emp_data.rename(columns={'sales':'dept'}, inplace=True)
```

In [8]:
```
emp_data.head()
```

Out[8]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5years	dept	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

In [9]:
```
import numpy as np
import pandas as pd
import seaborn as sns; sns.set(color_codes=True)
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
%matplotlib inline
```

In [10]:
```
emp_data.describe()
```

Out[10]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5years
count	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000
mean	0.612834	0.716102	3.803054	201.050337	3.498233	0.144610	0.238083	0.021268
std	0.248631	0.171169	1.232592	49.943099	1.460136	0.351719	0.425924	0.144281
min	0.090000	0.360000	2.000000	96.000000	2.000000	0.000000	0.000000	0.000000
25%	0.440000	0.560000	3.000000	156.000000	3.000000	0.000000	0.000000	0.000000
50%	0.640000	0.720000	4.000000	200.000000	3.000000	0.000000	0.000000	0.000000
75%	0.820000	0.870000	5.000000	245.000000	4.000000	0.000000	0.000000	0.000000
max	1.000000	1.000000	7.000000	310.000000	10.000000	1.000000	1.000000	1.000000

Preprocessing

In [12]:
```
emp_data.select_dtypes('object').columns
```

Out[12]:
```
Index(['dept', 'salary'], dtype='object')
```

In [15]:
```
emp_data.dept.value_counts()
```

Out[15]:
```
sales          4140
technical      2720
support        2229
IT             1227
product_mng     902
marketing       858
RandD           787
accounting      767
hr              739
management      630
Name: dept, dtype: int64
```

In [16]:
```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
```

In [17]:
```
le = LabelEncoder()
```

In [19]:
```
dept = le.fit_transform(emp_data.dept)
```

In [20]:
```
ohe = OneHotEncoder()
```

In [23]:
```
ohe_dept = ohe.fit_transform(dept.reshape(-1,1))
```

In [25]:
```
ohe.active_features_
```

Out[25]:
```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9], dtype=int64)
```

In [26]:
```
le.classes_
```

Out[26]:
```
array(['IT', 'RandD', 'accounting', 'hr', 'management', 'marketing',
       'product_mng', 'sales', 'support', 'technical'], dtype=object)
```

In [34]:
```
dept_df = pd.DataFrame(ohe_dept.toarray(), dtype=int,columns=le.classes_)
```

In [37]:
```
emp_data['salary_tf'] = emp_data.salary.map({'low':1,'medium':2,'high':3})
```

In [47]:
```
from sklearn.preprocessing import StandardScaler,MinMaxScaler
```

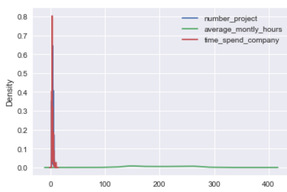
In [41]:
```
emp_data.columns
```

Out[41]:
```
Index(['satisfaction_level', 'last_evaluation', 'number_project',
       'average_montly_hours', 'time_spend_company', 'Work_accident', 'left',
       'promotion_last_5years', 'dept', 'salary', 'salary_tf'],
      dtype='object')
```

In [43]:
```
df = emp_data[['number_project','average_montly_hours', 'time_spend_company']]
```

In [44]:
```
df.plot.kde()
```

Out[44]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x159e3514ac8>
```



In [48]:
```
mm = MinMaxScaler()
```

In [53]:
```
scaled_np = mm.fit_transform(df)
```

In [54]:
```
dept_np = dept_df.values
```

In [57]:
```
emp_df = emp_data[['satisfaction_level','last_evaluation','Work_accident','promotion_last_5years','salary_tf']]
```

In [59]:
```
emp_np = emp_df.values
```

In [60]:
```
feature_data = np.hstack([emp_np, scaled_np, dept_np])
```

In [61]:
```
target_data = emp_data.left
```

In [63]:
```
feature_data.shape
```

Out[63]:
```
(14999, 18)
```

In [66]:
```
target_data.value_counts()
```

Out[66]:
```
0    11428
1     3571
Name: left, dtype: int64
```

Model Building

In [86]:
```
from sklearn.linear_model import LogisticRegression, SGDClassifier, PassiveAggressiveClassifier
from sklearn.ensemble import RandomForestClassifier
```

In [87]:
```
models = [ LogisticRegression(class_weight='balanced'), SGDClassifier(max_iter=10), PassiveAggressiveClassifier(max_iter=20), RandomForestClassifier(n_estimators=20)]
```

In [88]:
```
from sklearn.model_selection import train_test_split
```

In [89]:
```
trainX,testX,trainY,testY = train_test_split(feature_data,target_data)
```

In [90]:
```
for model in models:
    model.fit(trainX,trainY)
    print (model.score(testX,testY))
```

```
0.7626666666666667
0.8133333333333334
0.6269333333333333
0.9861333333333333
```

In [ ]:
```

```