employee attrition

June 2, 2020

[29]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

[30]:
```
pd.set_option('display.max_columns', 100)
attrition = pd.read_csv('WA_Fn-UseC_-HR-Employee-Attrition.csv')
attrition.head()
```

[30]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	Gender	HourlyRate	JobInvolvement	JobLevel	JobRole	JobSatisfaction	MaritalStatus	MonthlyIncome
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	2	Female	94	3	2	Sales Executive	4	Single	5993
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	3	Male	61	2	2	Research Scientist	2	Married	5130
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	4	Male	92	2	1	Laboratory Technician	3	Single	2090
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	4	Female	56	3	1	Research Scientist	3	Married	2909
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	1	Male	40	3	1	Laboratory Technician	2	Married	3468

```
   MonthlyRate  NumCompaniesWorked Over18 OverTime  PercentSalaryHike  \
0        19479                   8      Y      Yes                 11
1        24907                   1      Y       No                 23
2         2396                   6      Y      Yes                 15
3        23159                   1      Y      Yes                 11
4        16632                   9      Y       No                 12

   PerformanceRating  RelationshipSatisfaction  StandardHours  \
0                  3                         1             80
1                  4                         4             80
2                  3                         2             80
3                  3                         3             80
4                  3                         4             80

   StockOptionLevel  TotalWorkingYears  TrainingTimesLastYear  \
0                 0                  8                      0
1                 1                 10                      3
2                 0                  7                      3
3                 0                  8                      3
4                 1                  6                      3

   WorkLifeBalance  YearsAtCompany  YearsInCurrentRole  \
0                1               6                   4
1                3              10                   7
2                3               0                   0
3                3               8                   7
4                3               2                   2

   YearsSinceLastPromotion  YearsWithCurrManager
0                        0                     5
1                        1                     7
2                        0                     0
3                        3                     0
4                        2                     2
```

[31]:
```
attrition.isnull().sum()
```

[31]:
```
Age                         0
Attrition                   0
BusinessTravel              0
DailyRate                   0
Department                  0
DistanceFromHome            0
Education                   0
EducationField              0
EmployeeCount               0
EmployeeNumber              0
```

```
EnvironmentSatisfaction     0
Gender                      0
HourlyRate                  0
JobInvolvement              0
JobLevel                    0
JobRole                     0
JobSatisfaction             0
MaritalStatus               0
MonthlyIncome               0
MonthlyRate                 0
NumCompaniesWorked          0
Over18                      0
OverTime                    0
PercentSalaryHike           0
PerformanceRating           0
RelationshipSatisfaction    0
StandardHours               0
StockOptionLevel            0
TotalWorkingYears           0
TrainingTimesLastYear       0
WorkLifeBalance             0
YearsAtCompany              0
YearsInCurrentRole          0
YearsSinceLastPromotion     0
YearsWithCurrManager        0
dtype: int64
```

[32]:
```
attrition.dtypes
```

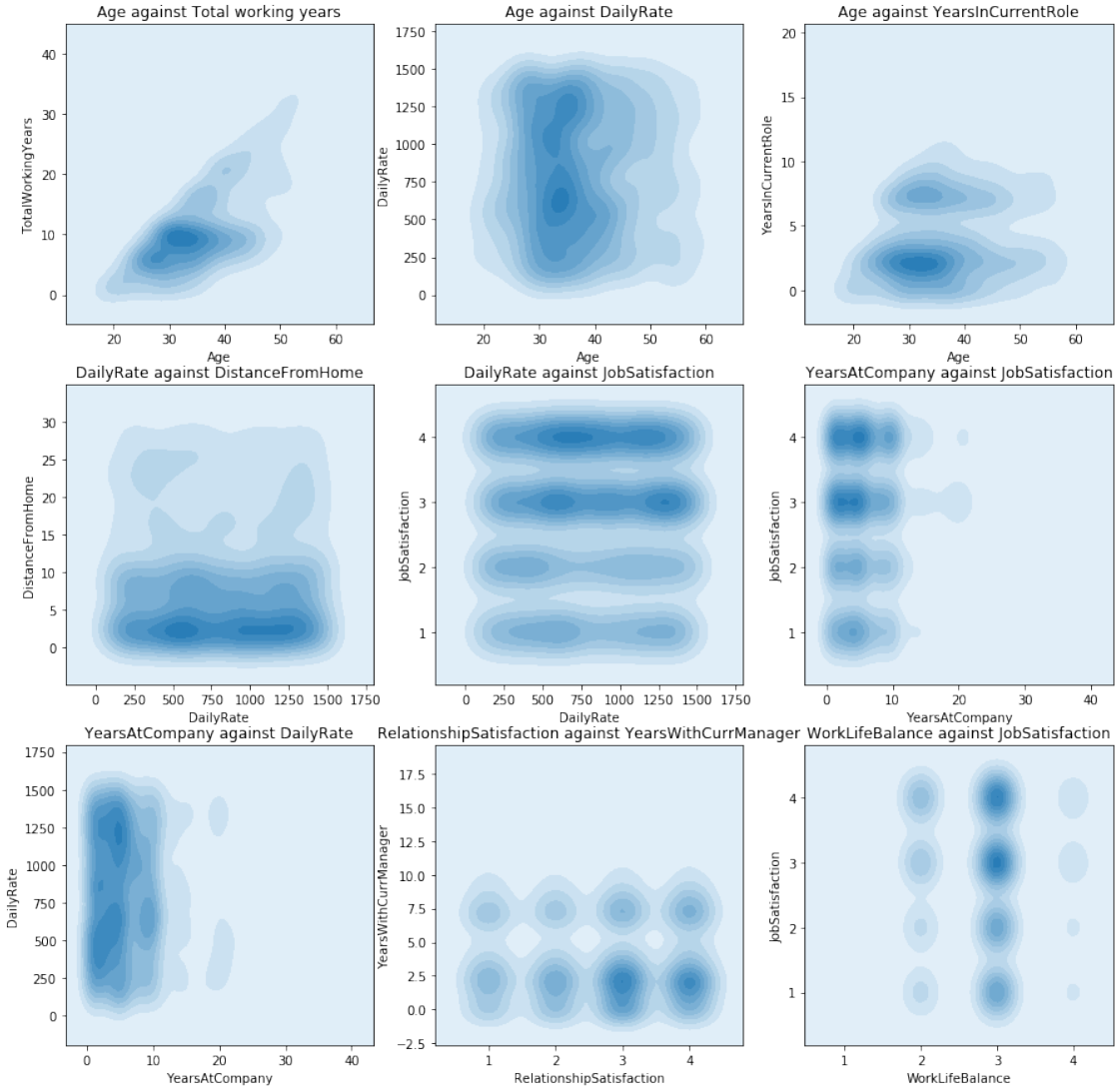
[32]:
```
Age                         int64
Attrition                  object
BusinessTravel             object
DailyRate                   int64
Department                 object
DistanceFromHome            int64
Education                   int64
EducationField             object
EmployeeCount               int64
EmployeeNumber              int64
EnvironmentSatisfaction     int64
Gender                     object
HourlyRate                  int64
JobInvolvement              int64
JobLevel                    int64
JobRole                    object
JobSatisfaction             int64
MaritalStatus              object
MonthlyIncome               int64
```

```
MonthlyRate                int64
NumCompaniesWorked         int64
Over18                    object
OverTime                  object
PercentSalaryHike          int64
PerformanceRating          int64
RelationshipSatisfaction   int64
StandardHours              int64
StockOptionLevel           int64
TotalWorkingYears          int64
TrainingTimesLastYear      int64
WorkLifeBalance            int64
YearsAtCompany             int64
YearsInCurrentRole         int64
YearsSinceLastPromotion    int64
YearsWithCurrManager       int64
dtype: object
```

[33]:
```
f, axes = plt.subplots(3, 3, figsize=(15, 15), sharex=False, sharey=False)
sns.kdeplot(attrition['Age'], attrition['TotalWorkingYears'], shade=True,␣
 ↪ax=axes[0,0])
axes[0,0].set( title = 'Age against Total working years')

sns.kdeplot(attrition['Age'], attrition['DailyRate'], shade=True, ax=axes[0,1])
axes[0,1].set( title = 'Age against DailyRate')

sns.kdeplot(attrition['Age'], attrition['YearsInCurrentRole'], shade=True,␣
 ↪ax=axes[0,2])
axes[0,2].set( title = 'Age against YearsInCurrentRole')

sns.kdeplot(attrition['DailyRate'], attrition['DistanceFromHome'], shade=True,␣
 ↪ax=axes[1,0])
axes[1,0].set( title = 'DailyRate against DistanceFromHome')

sns.kdeplot(attrition['DailyRate'], attrition['JobSatisfaction'], shade=True,␣
 ↪ax=axes[1,1])
axes[1,1].set( title = 'DailyRate against JobSatisfaction')

sns.kdeplot(attrition['YearsAtCompany'], attrition['JobSatisfaction'],␣
 ↪shade=True, ax=axes[1,2])
axes[1,2].set( title = 'YearsAtCompany against JobSatisfaction')

sns.kdeplot(attrition['YearsAtCompany'], attrition['DailyRate'], shade=True,␣
 ↪ax=axes[2,0])
axes[2,0].set( title = 'YearsAtCompany against DailyRate')
```

```
sns.kdeplot(attrition['RelationshipSatisfaction'],␣
 ↪attrition['YearsWithCurrManager'], shade=True, ax=axes[2,1])
axes[2,1].set( title = 'RelationshipSatisfaction against YearsWithCurrManager')

sns.kdeplot(attrition['WorkLifeBalance'], attrition['JobSatisfaction'],␣
 ↪shade=True, ax=axes[2,2])
axes[2,2].set( title = 'WorkLifeBalance against JobSatisfaction')
```

[33]:
```
[Text(0.5, 1.0, 'WorkLifeBalance against JobSatisfaction')]
```

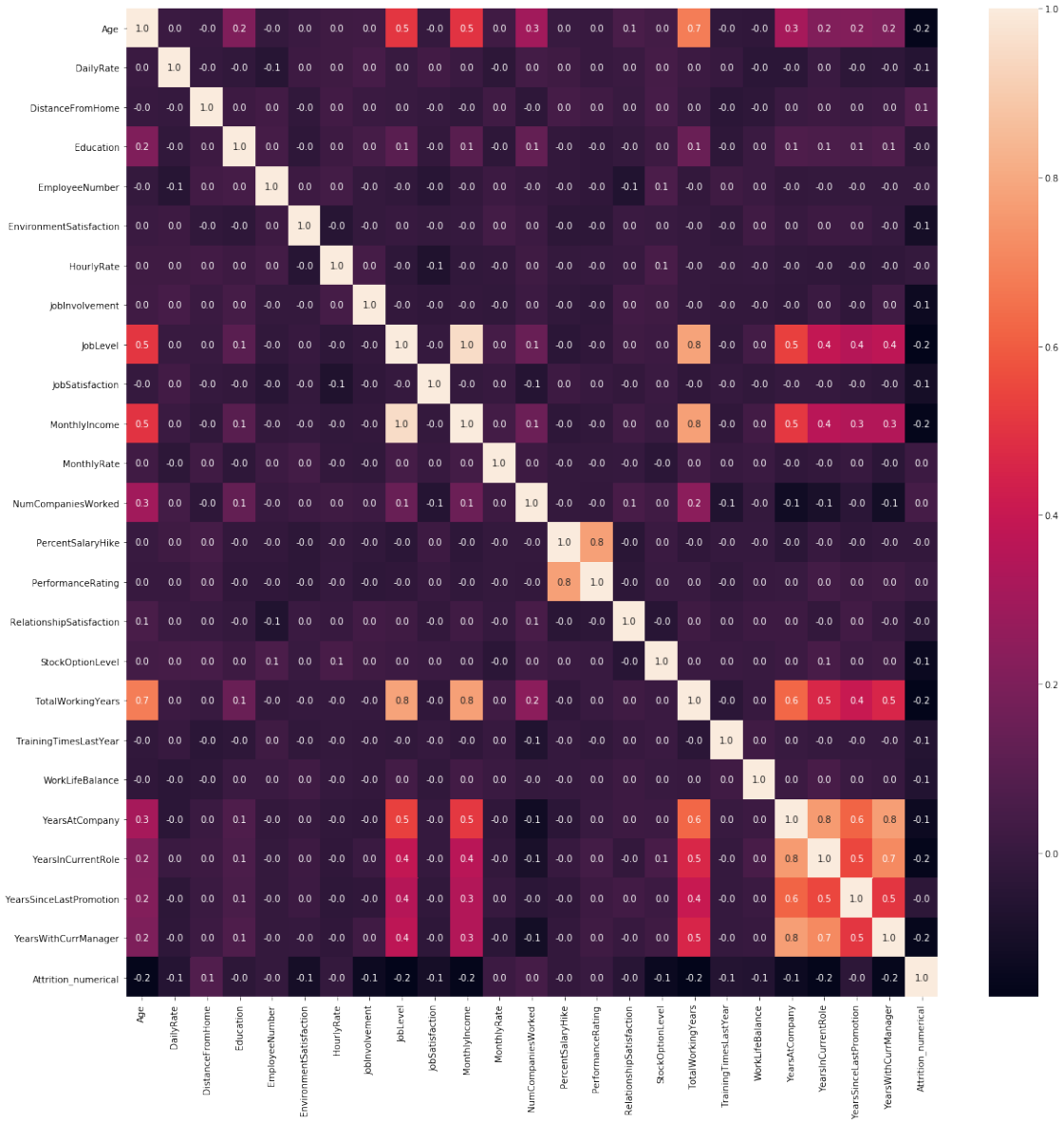


[34]:
```
attrition["Attrition_numerical"]=attrition['Attrition'].map({'Yes':1, 'No':0})
```

[35]:
```
numerical = [col for col in attrition.columns if attrition[col].dtype ==␣
 ↪'int64']
```

[8]:
```
num_df = attrition[numerical]
num_df=num_df.drop(['EmployeeCount', 'StandardHours'], axis = 1)
plt.figure(figsize = (20, 20))
sns.heatmap(num_df.corr(), annot = True, fmt= '0.1f')
```

[8]: <matplotlib.axes._subplots.AxesSubplot at 0x1a240a62b0>

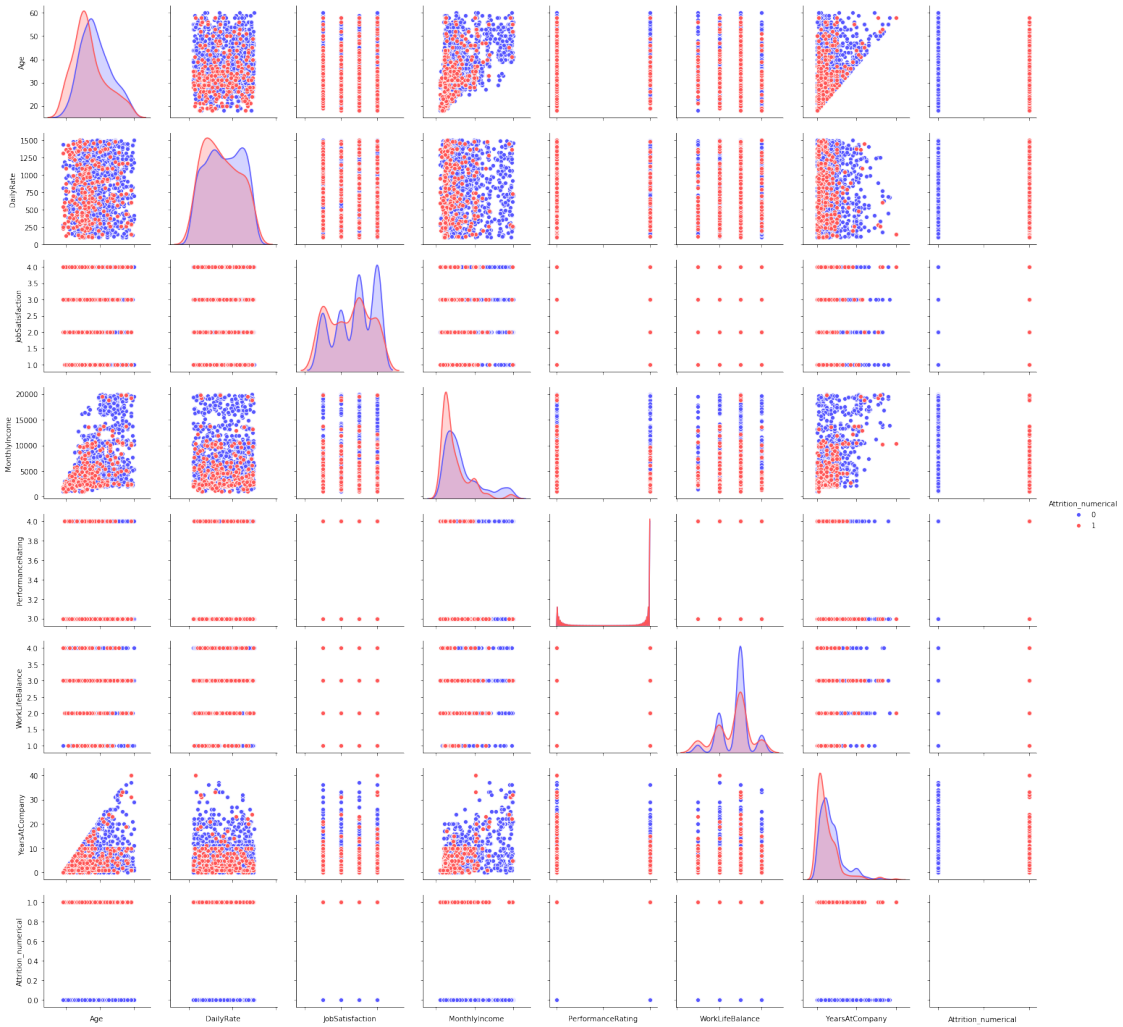


[9]:
```
numerical = [u'Age', u'DailyRate',  u'JobSatisfaction',
       u'MonthlyIncome', u'PerformanceRating',
        u'WorkLifeBalance', u'YearsAtCompany', u'Attrition_numerical']
```

```
g = sns.pairplot(attrition[numerical], hue='Attrition_numerical',␣
 ↪palette='seismic', diag_kind = 'kde',diag_kws=dict(shade=True))
g.set(xticklabels=[])
```

```
/Users/shijiecai/anaconda3/lib/python3.7/site-
packages/statsmodels/nonparametric/kde.py:487: RuntimeWarning: invalid value
encountered in true_divide
  binned = fast_linbin(X, a, b, gridsize) / (delta * nobs)
/Users/shijiecai/anaconda3/lib/python3.7/site-
packages/statsmodels/nonparametric/kdetools.py:34: RuntimeWarning: invalid value
encountered in double_scalars
  FAC1 = 2*(np.pi*bw/RANGE)**2
```

[9]: `<seaborn.axisgrid.PairGrid at 0x10f25b160>`



[10]:
```
a = pd.DataFrame(np.random.randn(4,3),columns=['col1','col2','col3'])
a
```

[10]:
```
       col1      col2      col3
0  2.493088  0.218716  1.472357
1 -0.259747  0.142459  0.397852
2  0.639417  0.555888 -0.133025
3 -1.610374  0.984699  0.205528
```

[36]:
```
target = attrition['Attrition'].map({'Yes':1, 'No':0})
attrition1 = attrition.drop(['Attrition','Attrition_numerical'], axis = 1)
attrition_final = pd.get_dummies(data = attrition1,  columns = [col for col in␣
 ↪attrition1.columns if attrition1[col].dtype == 'object'])
```

[37]:
```
attrition_final.head()
```

[37]:
```
   Age  DailyRate  DistanceFromHome  Education  EmployeeCount  EmployeeNumber  \
0   41       1102                 1          2              1               1
1   49        279                 8          1              1               2
2   37       1373                 2          2              1               4
3   33       1392                 3          4              1               5
4   27        591                 2          1              1               7

   EnvironmentSatisfaction  HourlyRate  JobInvolvement  JobLevel  \
0                        2          94               3         2
1                        3          61               2         2
2                        4          92               2         1
3                        4          56               3         1
4                        1          40               3         1

   JobSatisfaction  MonthlyIncome  MonthlyRate  NumCompaniesWorked  \
0                4           5993        19479                   8
1                2           5130        24907                   1
2                3           2090         2396                   6
3                3           2909        23159                   1
4                2           3468        16632                   9

   PercentSalaryHike  PerformanceRating  RelationshipSatisfaction  \
0                 11                  3                         1
1                 23                  4                         4
2                 15                  3                         2
3                 11                  3                         3
4                 12                  3                         4

   StandardHours  StockOptionLevel  TotalWorkingYears  TrainingTimesLastYear  \
0             80                 0                  8                      0
1             80                 1                 10                      3
2             80                 0                  7                      3
3             80                 0                  8                      3
```

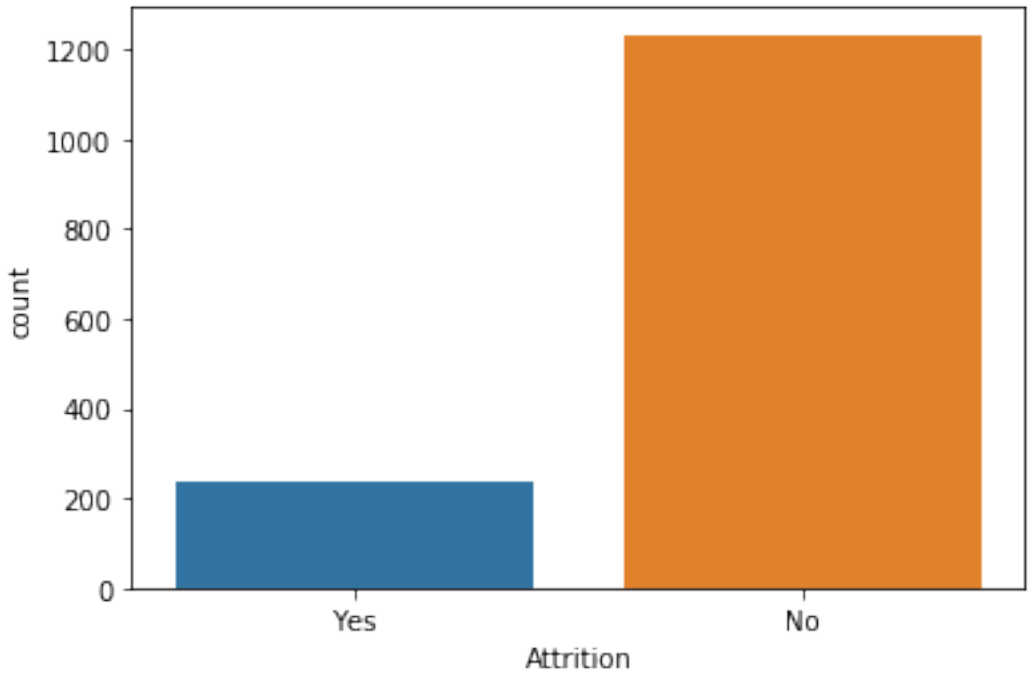
```
4             80                  1                   6                         3

   WorkLifeBalance  YearsAtCompany  YearsInCurrentRole  \
0                1               6                   4
1                3              10                   7
2                3               0                   0
3                3               8                   7
4                3               2                   2

   YearsSinceLastPromotion  YearsWithCurrManager  BusinessTravel_Non-Travel  \
0                        0                     5                          0
1                        1                     7                          0
2                        0                     0                          0
3                        3                     0                          0
4                        2                     2                          0

   BusinessTravel_Travel_Frequently  BusinessTravel_Travel_Rarely  \
0                                 0                             1
1                                 1                             0
2                                 0                             1
3                                 1                             0
4                                 0                             1

   Department_Human Resources  Department_Research & Development  \
0                           0                                   0
1                           0                                   1
2                           0                                   1
3                           0                                   1
4                           0                                   1

   Department_Sales  EducationField_Human Resources  \
0                 1                               0
1                 0                               0
2                 0                               0
3                 0                               0
4                 0                               0

   EducationField_Life Sciences  EducationField_Marketing  \
0                             1                         0
1                             1                         0
2                             0                         0
3                             1                         0
4                             0                         0

   EducationField_Medical  EducationField_Other  \
0                       0                     0
1                       0                     0
```

```
2                       0                    1
3                       0                    0
4                       1                    0

   EducationField_Technical Degree  Gender_Female  Gender_Male  \
0                                0              1            0
1                                0              0            1
2                                0              0            1
3                                0              1            0
4                                0              0            1

   JobRole_Healthcare Representative  JobRole_Human Resources  \
0                                  0                        0
1                                  0                        0
2                                  0                        0
3                                  0                        0
4                                  0                        0

   JobRole_Laboratory Technician  JobRole_Manager  \
0                              0                0
1                              0                0
2                              1                0
3                              0                0
4                              1                0

   JobRole_Manufacturing Director  JobRole_Research Director  \
0                               0                         0
1                               0                         0
2                               0                         0
3                               0                         0
4                               0                         0

   JobRole_Research Scientist  JobRole_Sales Executive  \
0                           0                        1
1                           1                        0
2                           0                        0
3                           1                        0
4                           0                        0

   JobRole_Sales Representative  MaritalStatus_Divorced  \
0                             0                       0
1                             0                       0
2                             0                       0
3                             0                       0
4                             0                       0

   MaritalStatus_Married  MaritalStatus_Single  Over18_Y  OverTime_No  \
```

```
0                    0                    1          1          0
1                    1                    0          1          1
2                    0                    1          1          0
3                    1                    0          1          0
4                    1                    0          1          1

   OverTime_Yes
0             1
1             0
2             1
3             1
4             0
```

[38]:
```
sns.countplot(attrition['Attrition'])
```

[38]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1c4536ba20>
```



[39]:
```
from sklearn.model_selection import train_test_split
from sklearn.model_selection import StratifiedShuffleSplit

train, test, target_train, target_val = train_test_split(attrition_final,
                                                         target,
                                                         train_size= 0.80,
                                                         random_state=0);
```

[40]:
```
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import (accuracy_score, log_loss, classification_report)
```

[41]:
```
import scipy
from imblearn.over_sampling import SMOTE
```

[42]:
```
import xgboost
```

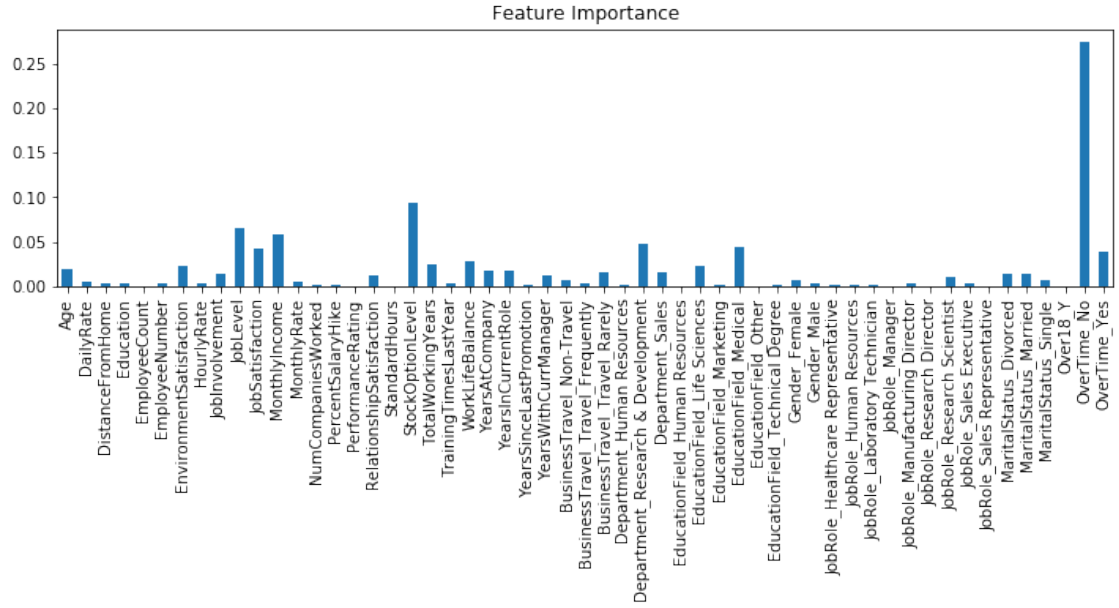
[43]:
```
oversampler=SMOTE(random_state=0)
smote_train, smote_target = oversampler.fit_sample(train,target_train)
```

[44]:
```
#seed = 0
rf_params = {
    'n_estimators': 1000,
    'max_features': 0.3,
    'max_depth': 4,
    'random_state' : 0,
}

rf = RandomForestClassifier(**rf_params)

rf.fit(smote_train, smote_target)

rf_predictions = rf.predict(test)

print("Accuracy score: {}".format(accuracy_score(target_val, rf_predictions)))

print(classification_report(target_val, rf_predictions))
```

```
Accuracy score: 0.8367346938775511
              precision    recall  f1-score   support

           0       0.89      0.92      0.90       245
           1       0.51      0.43      0.47        49

    accuracy                           0.84       294
   macro avg       0.70      0.67      0.69       294
weighted avg       0.83      0.84      0.83       294
```

[49]:
```
plt.figure(figsize=(12,3))
features = smote_train.columns.values.tolist()
importance = rf.feature_importances_.tolist()
feature_series = pd.Series(data=importance,index=features)
feature_series.plot.bar()

plt.title('Feature Importance')
```

[49]: `Text(0.5, 1.0, 'Feature Importance')`



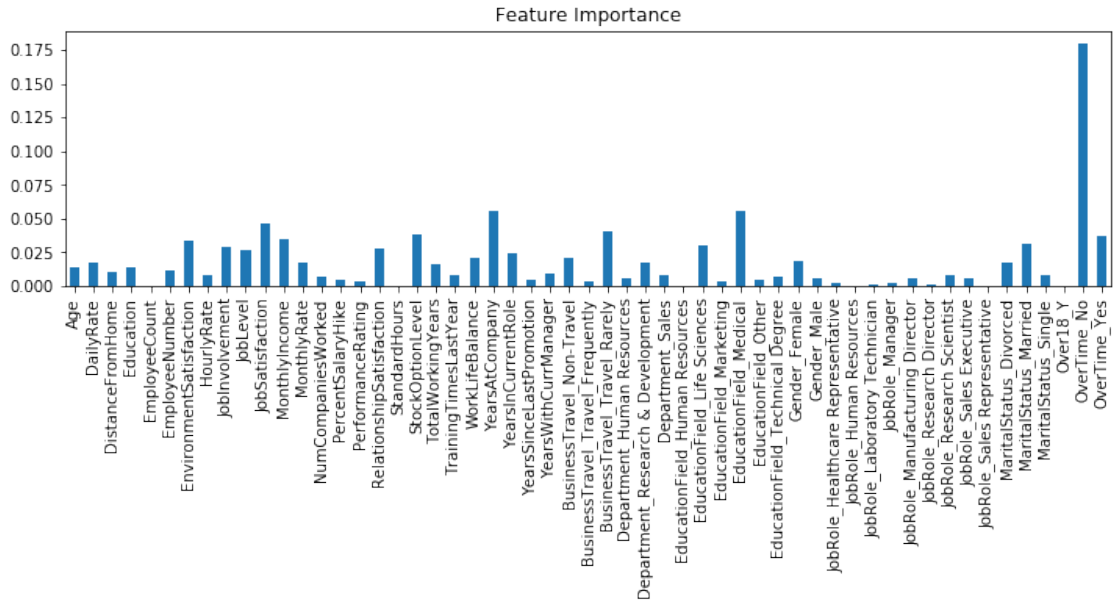
[50]:
```
gb_params ={
    'n_estimators': 1500,
    'max_features': 0.9,
    'learning_rate' : 0.25,
    'max_depth': 4,
    'min_samples_leaf': 2,
    'subsample': 1,
    'max_features' : 'sqrt',
    'random_state' : seed,
    'verbose': 0
}

gb = GradientBoostingClassifier(**gb_params)

gb.fit(smote_train, smote_target)

gb_predictions = gb.predict(test)

print(accuracy_score(target_val, gb_predictions))
print(classification_report(target_val, gb_predictions))
```

```
0.8673469387755102
              precision    recall  f1-score   support

           0       0.88      0.97      0.92       245
```

```
           1       0.69      0.37      0.48        49

    accuracy                           0.87       294
   macro avg       0.79      0.67      0.70       294
weighted avg       0.85      0.87      0.85       294
```

[51]:
```
plt.figure(figsize=(12,3))
features = smote_train.columns.values.tolist()
importance = gb.feature_importances_.tolist()
feature_series = pd.Series(data=importance,index=features)
feature_series.plot.bar()

plt.title('Feature Importance')
```

[51]:
```
Text(0.5, 1.0, 'Feature Importance')
```



[ ]: