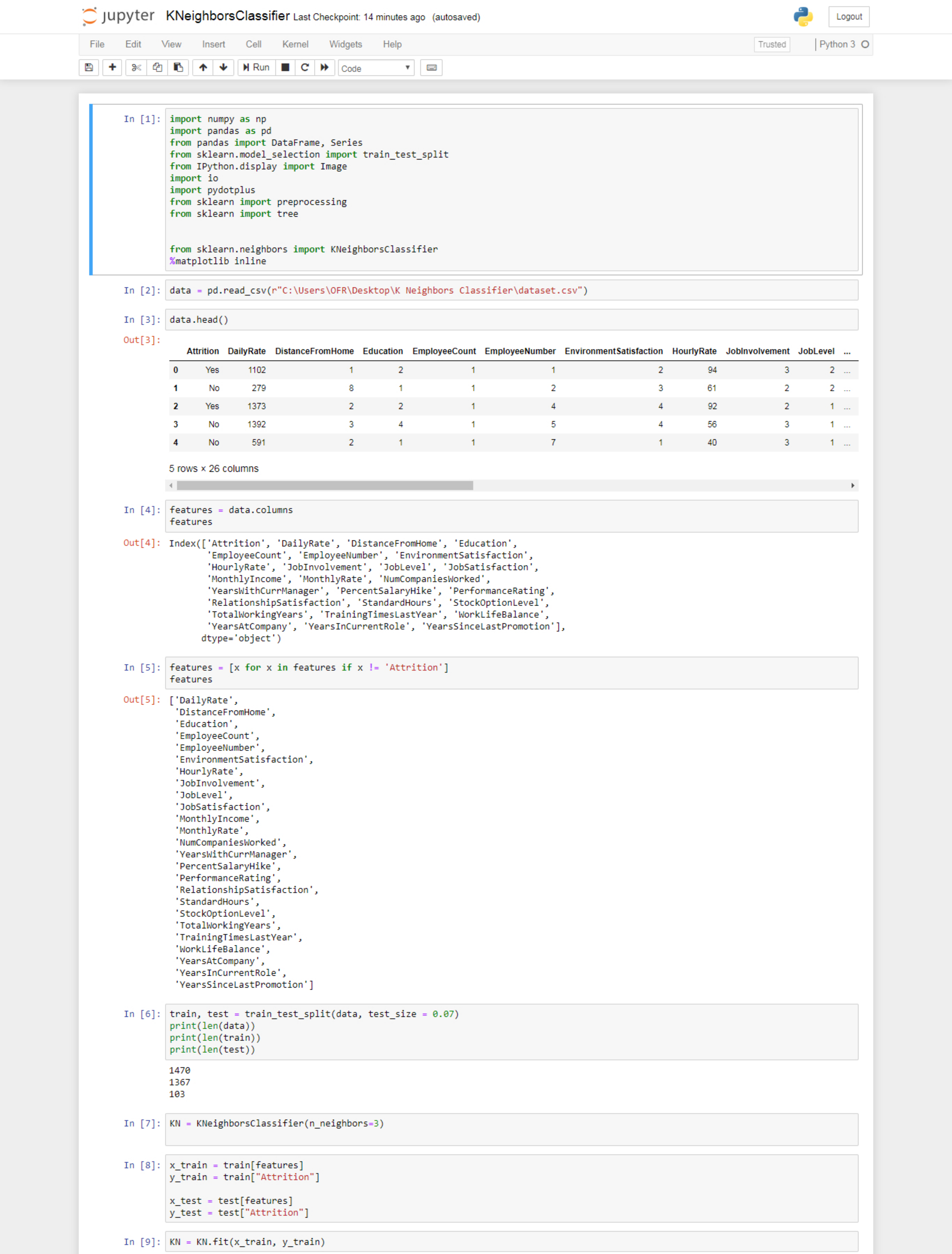


KNeighborsClassifier Last Checkpoint: 14 minutes ago (autosaved)

In [1]:
```
import numpy as np
import pandas as pd
from pandas import DataFrame, Series
from sklearn.model_selection import train_test_split
from IPython.display import Image
import io
import pydotplus
from sklearn import preprocessing
from sklearn import tree


from sklearn.neighbors import KNeighborsClassifier
%matplotlib inline
```

In [2]:
```
data = pd.read_csv(r"C:\Users\OFR\Desktop\K Neighbors Classifier\dataset.csv")
```

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

Out[3]:

	Attrition	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	JobInvolvement	JobLevel	...
0	Yes	1102	1	2	1	1	2	94	3	2	...
1	No	279	8	1	1	2	3	61	2	2	...
2	Yes	1373	2	2	1	4	4	92	2	1	...
3	No	1392	3	4	1	5	4	56	3	1	...
4	No	591	2	1	1	7	1	40	3	1	...

5 rows × 26 columns

In [4]:
```
features = data.columns
features
```

Out[4]:
```
Index(['Attrition', 'DailyRate', 'DistanceFromHome', 'Education',
       'EmployeeCount', 'EmployeeNumber', 'EnvironmentSatisfaction',
       'HourlyRate', 'JobInvolvement', 'JobLevel', 'JobSatisfaction',
       'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked',
       'YearsWithCurrManager', 'PercentSalaryHike', 'PerformanceRating',
       'RelationshipSatisfaction', 'StandardHours', 'StockOptionLevel',
       'TotalWorkingYears', 'TrainingTimesLastYear', 'WorkLifeBalance',
       'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion'],
      dtype='object')
```

In [5]:
```
features = [x for x in features if x != 'Attrition']
features
```

Out[5]:
```
['DailyRate',
 'DistanceFromHome',
 'Education',
 'EmployeeCount',
 'EmployeeNumber',
 'EnvironmentSatisfaction',
 'HourlyRate',
 'JobInvolvement',
 'JobLevel',
 'JobSatisfaction',
 'MonthlyIncome',
 'MonthlyRate',
 'NumCompaniesWorked',
 'YearsWithCurrManager',
 'PercentSalaryHike',
 'PerformanceRating',
 'RelationshipSatisfaction',
 'StandardHours',
 'StockOptionLevel',
 'TotalWorkingYears',
 'TrainingTimesLastYear',
 'WorkLifeBalance',
 'YearsAtCompany',
 'YearsInCurrentRole',
 'YearsSinceLastPromotion']
```

In [6]:
```
train, test = train_test_split(data, test_size = 0.07)
print(len(data))
print(len(train))
print(len(test))
```

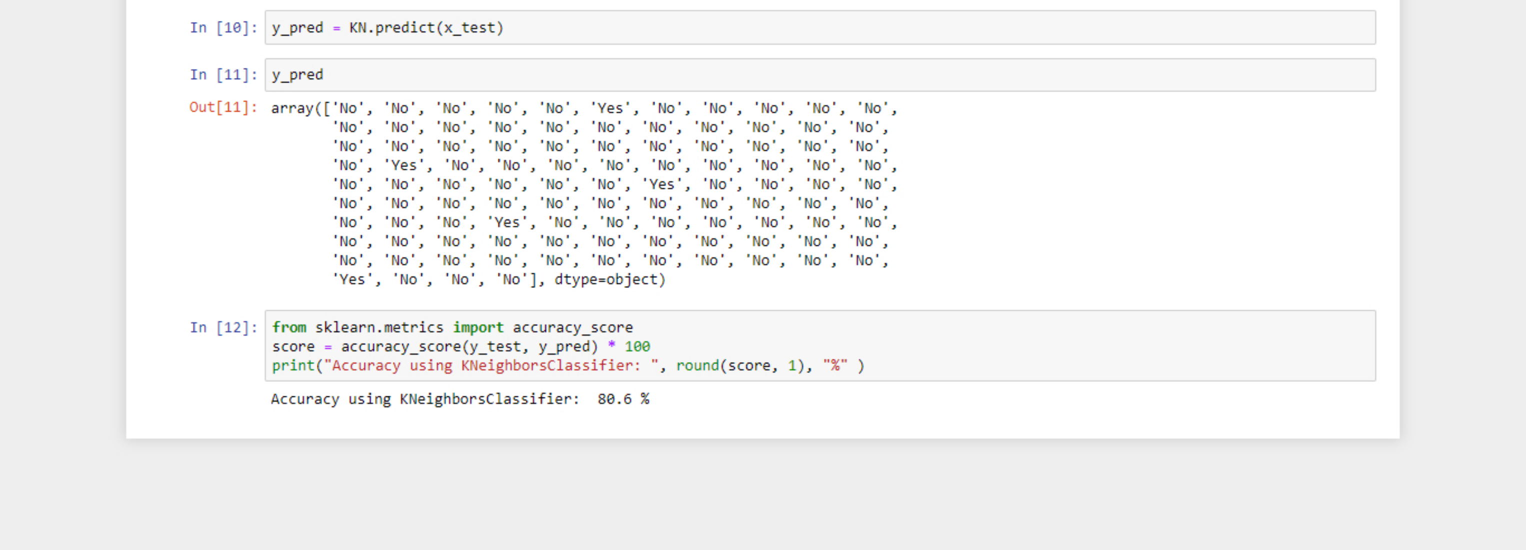
```
1470
1367
103
```

In [7]:
```
KN = KNeighborsClassifier(n_neighbors=3)
```

In [8]:
```
x_train = train[features]
y_train = train["Attrition"]

x_test = test[features]
y_test = test["Attrition"]
```

In [9]:
```
KN = KN.fit(x_train, y_train)
```



```
In [10]: y_pred = KN.predict(x_test)
```

```
In [11]: y_pred
```

```
Out[11]: array(['No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No',
                'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
                'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
                'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
                'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No',
                'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
                'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
                'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
                'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
                'Yes', 'No', 'No', 'No'], dtype=object)
```

```
In [12]: from sklearn.metrics import accuracy_score
         score = accuracy_score(y_test, y_pred) * 100
         print("Accuracy using KNeighborsClassifier: ", round(score, 1), "%" )
```

```
Accuracy using KNeighborsClassifier:  80.6 %
```