Data Exploration

In [264]:

```
import numpy as np
```

In [265]:

```
import pandas as pd
```

In [266]:

```
df = pd.read_csv("https://raw.githubusercontent.com/anshupandey/Machine_Learning_Train
df.shape
```

Out[266]:

```
(1000, 7)
```

In [267]:

```
df.head()
```

Out[267]:

	lifetime	broken	pressureInd	moistureInd	temperatureInd	team	provider
0	56	0	92.178854	104.230204	96.517159	TeamA	Provider4
1	81	1	72.075938	183.065701	87.271062	TeamC	Provider4
2	60	0	96.272254	77.801376	112.196170	TeamA	Provider1
3	86	1	94.406461	178.493608	72.025374	TeamC	Provider2
4	34	0	97.752899	99.413492	103.756271	TeamB	Provider1

In [268]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 7 columns):
lifetime          1000 non-null int64
broken            1000 non-null int64
pressureInd       996 non-null float64
moistureInd       1000 non-null float64
temperatureInd    997 non-null float64
team              1000 non-null object
provider          1000 non-null object
dtypes: float64(3), int64(2), object(2)
memory usage: 54.8+ KB
```

In [269]:

```
df.describe()

```

Out[269]:

	lifetime	broken	pressureInd	moistureInd	temperatureInd
count	1000.000000	1000.000000	996.000000	1000.000000	997.000000
mean	55.195000	0.397000	98.681100	111.088723	100.553499
std	26.472737	0.489521	19.879703	41.839005	19.592059
min	1.000000	0.000000	33.481917	70.928815	42.279598
25%	34.000000	0.000000	85.562282	94.532547	87.672094
50%	60.000000	0.000000	97.311091	102.844084	100.528015
75%	80.000000	1.000000	112.253190	113.532970	113.522496
max	93.000000	1.000000	173.282541	1156.493254	172.544140

DATA CLEANING

In [270]:

```
df.duplicated().sum()
```

Out[270]:

```
0
```

In [271]:

```
df.isnull().sum()
```

Out[271]:

```
lifetime          0
broken            0
pressureInd       4
moistureInd       0
temperatureInd    3
team              0
provider          0
dtype: int64
```

In [272]:

```
df.dropna(thresh=3,inplace=True)
```

In [273]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1000 entries, 0 to 999
Data columns (total 7 columns):
lifetime          1000 non-null int64
broken            1000 non-null int64
pressureInd       996 non-null float64
moistureInd       1000 non-null float64
temperatureInd    997 non-null float64
team              1000 non-null object
provider          1000 non-null object
dtypes: float64(3), int64(2), object(2)
memory usage: 62.5+ KB
```

In [274]:

```
df.team.unique()
```

Out[274]:

```
array(['TeamA', 'TeamC', 'TeamB'], dtype=object)
```

In [275]:

```
df.provider.unique()
```

Out[275]:

```
array(['Provider4', 'Provider1', 'Provider2', 'Provider3'], dtype=object)
```

In [276]:

```
df.isnull().sum()
```

Out[276]:

```
lifetime          0
broken            0
pressureInd       4
moistureInd       0
temperatureInd    3
team              0
provider          0
dtype: int64
```

In [277]:

```
df.skew()
```

Out[277]:

```
lifetime          -0.407597
broken             0.421663
pressureInd        0.117541
moistureInd       15.982324
temperatureInd    -0.070839
dtype: float64
```

In [278]:

```
median = df['temperatureInd'].median()
median
```

Out[278]:

```
100.52801459999999
```

In [279]:

```
df['temperatureInd'].fillna(median, inplace=True)
```

In [280]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1000 entries, 0 to 999
Data columns (total 7 columns):
lifetime          1000 non-null int64
broken            1000 non-null int64
pressureInd       996 non-null float64
moistureInd       1000 non-null float64
temperatureInd    1000 non-null float64
team              1000 non-null object
provider          1000 non-null object
dtypes: float64(3), int64(2), object(2)
memory usage: 62.5+ KB
```

In [281]:

```
mean = df['pressureInd'].mean()
mean
```

Out[281]:

```
98.68109962325292
```

In [282]:

```
df['pressureInd'].fillna(mean, inplace=True)
```

In [283]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1000 entries, 0 to 999
Data columns (total 7 columns):
lifetime          1000 non-null int64
broken            1000 non-null int64
pressureInd       1000 non-null float64
moistureInd       1000 non-null float64
temperatureInd    1000 non-null float64
team              1000 non-null object
provider          1000 non-null object
dtypes: float64(3), int64(2), object(2)
memory usage: 62.5+ KB
```

In [284]:

```
df.isnull().sum()
```

Out[284]:

```
lifetime          0
broken            0
pressureInd       0
moistureInd       0
temperatureInd    0
team              0
provider          0
dtype: int64
```

In [285]:

```
df.skew()
```

Out[285]:

```
lifetime          -0.407597
broken             0.421663
pressureInd        0.117776
moistureInd       15.982324
temperatureInd    -0.070933
dtype: float64
```

In [286]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1000 entries, 0 to 999
Data columns (total 7 columns):
lifetime          1000 non-null int64
broken            1000 non-null int64
pressureInd       1000 non-null float64
moistureInd       1000 non-null float64
temperatureInd    1000 non-null float64
team              1000 non-null object
provider          1000 non-null object
dtypes: float64(3), int64(2), object(2)
memory usage: 62.5+ KB
```

In [ ]:

```


```

In [287]:

```
df[(df.moistureInd >250)]
```

Out[287]:

	lifetime	broken	pressureInd	moistureInd	temperatureInd	team	provider
604	80	1	96.105244	1156.493254	97.143188	TeamB	Provider1

In [288]:

```
mean1 = df['moistureInd'].mean()
mean1
```

Out[288]:

```
111.08872284591999
```

In [289]:

```
df.loc[(df.moistureInd >250),'moistureInd']=mean1
```

In [290]:

```
df[(df.moistureInd >250)]
```

Out[290]:

	lifetime	broken	pressureInd	moistureInd	temperatureInd	team	provider

In [291]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1000 entries, 0 to 999
Data columns (total 7 columns):
lifetime          1000 non-null int64
broken            1000 non-null int64
pressureInd       1000 non-null float64
moistureInd       1000 non-null float64
temperatureInd    1000 non-null float64
team              1000 non-null object
provider          1000 non-null object
dtypes: float64(3), int64(2), object(2)
memory usage: 62.5+ KB
```

DATA ANALYTICS

In [292]:

```
df.columns
```

Out[292]:

```
Index(['lifetime', 'broken', 'pressureInd', 'moistureInd', 'temperatureInd',
       'team', 'provider'],
      dtype='object')
```

In [293]:

```
num = ['lifetime', 'pressureInd', 'moistureInd', 'temperatureInd']
cats = ['broken','team', 'provider']
```

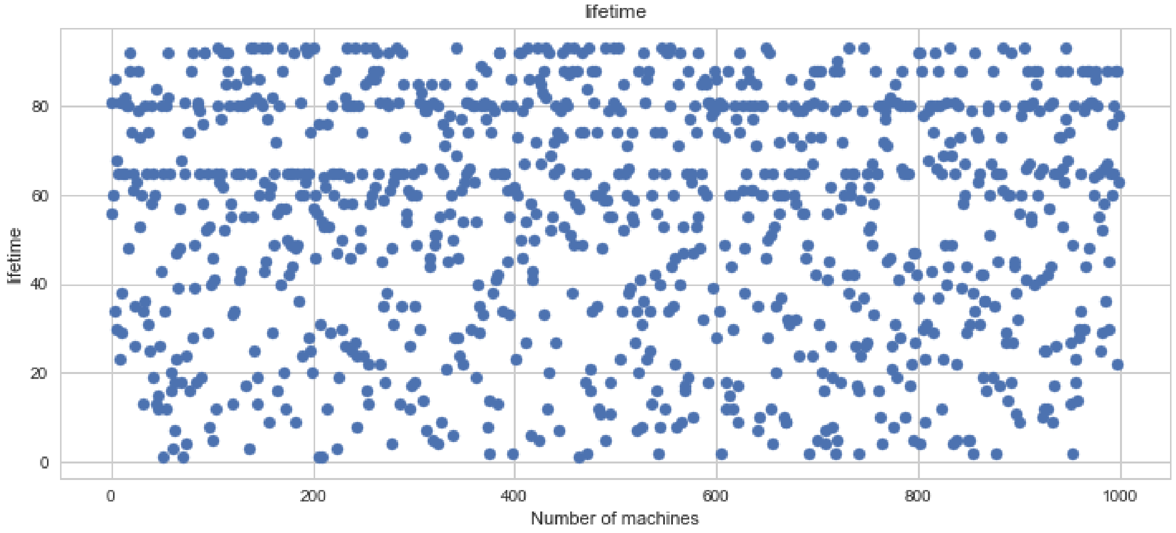
Univarient Analytics

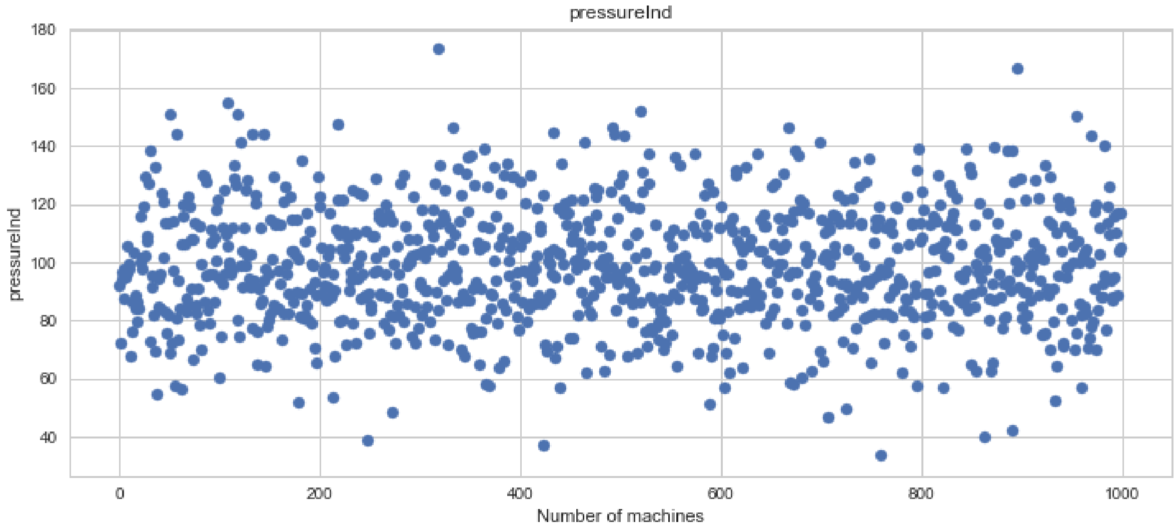
In [294]:

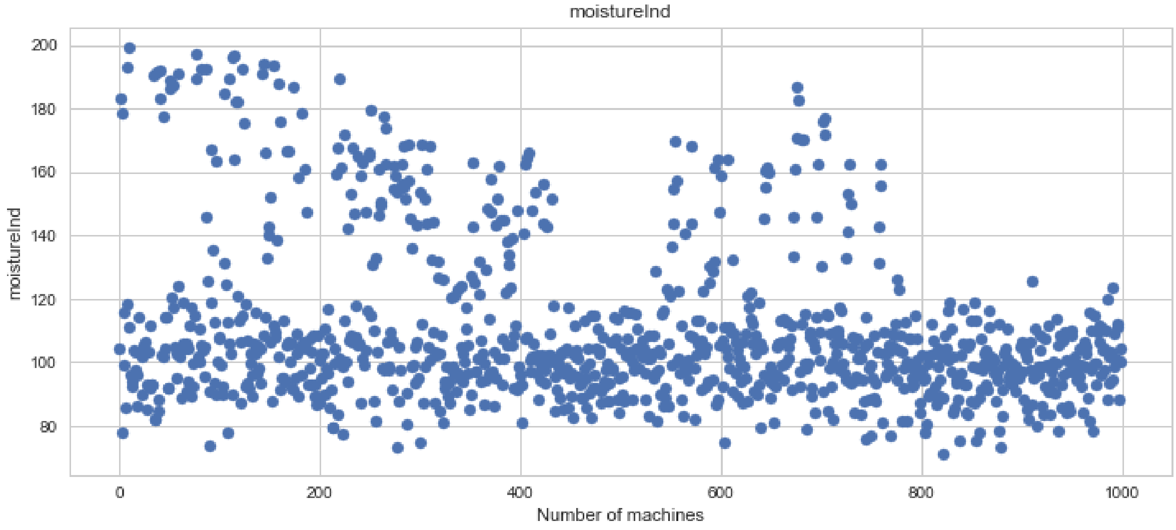
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

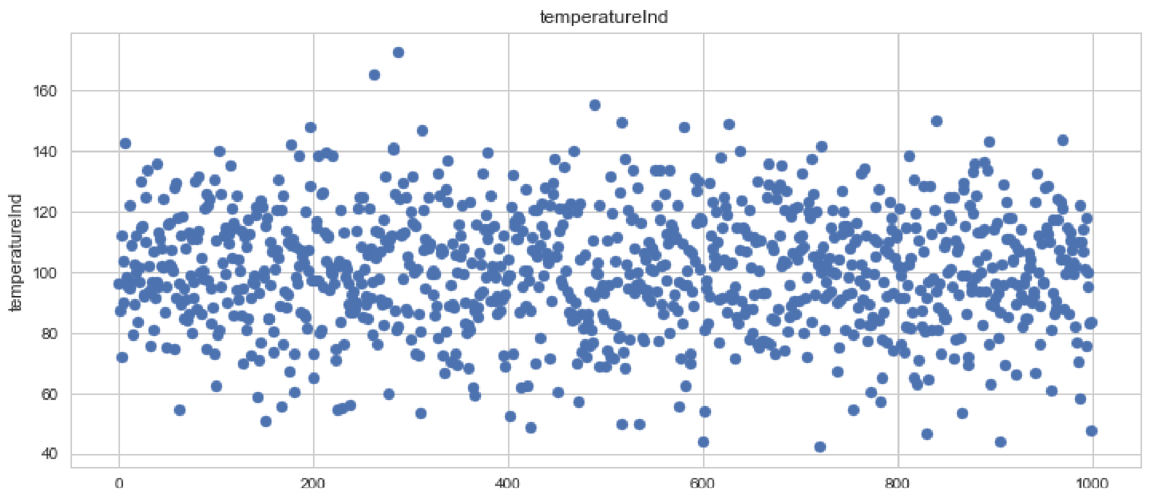
In [295]:

```
for col in num:
    plt.figure(figsize=(12,5))
    plt.scatter(np.arange(1000),df[col])
    plt.xlabel("Number of machines")
    plt.ylabel(col)
    plt.title(col)
    plt.show()
```







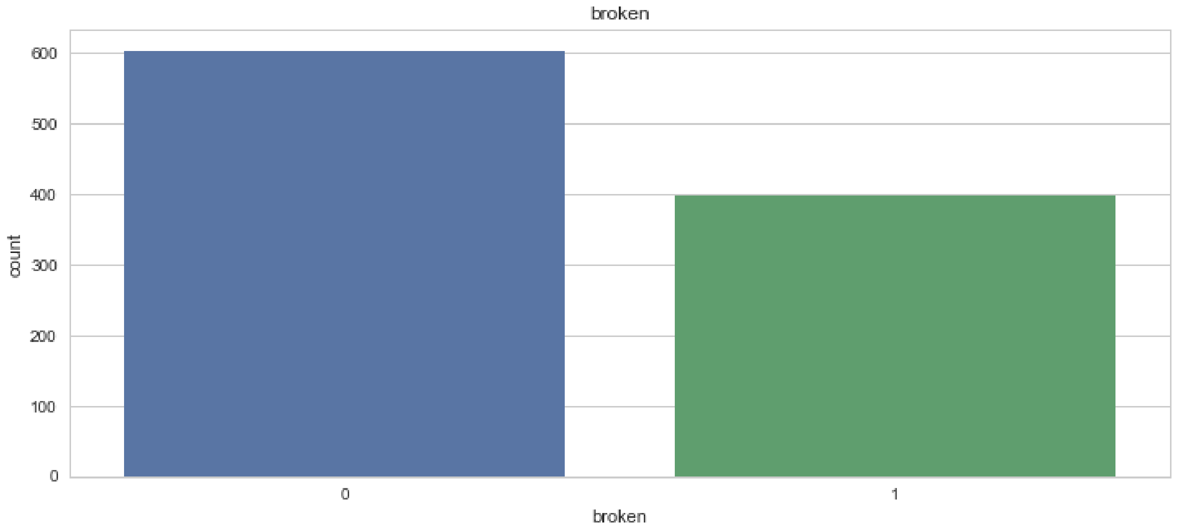


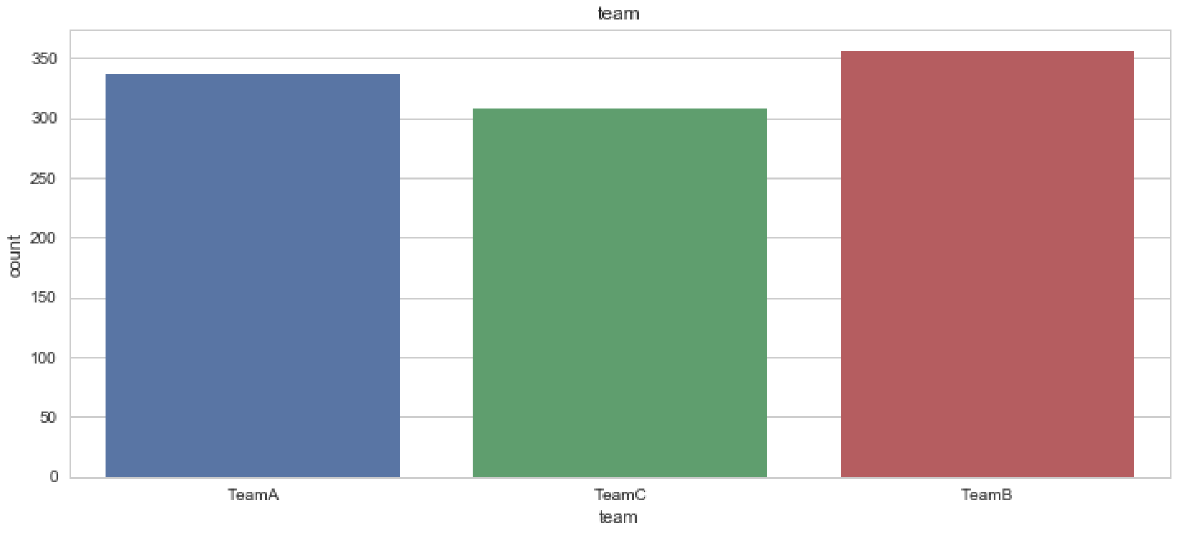
#Note : ##From the univarient analysis, we can say that the lifetime of machines lies anywhere between 10-90 years and spread uniformly.

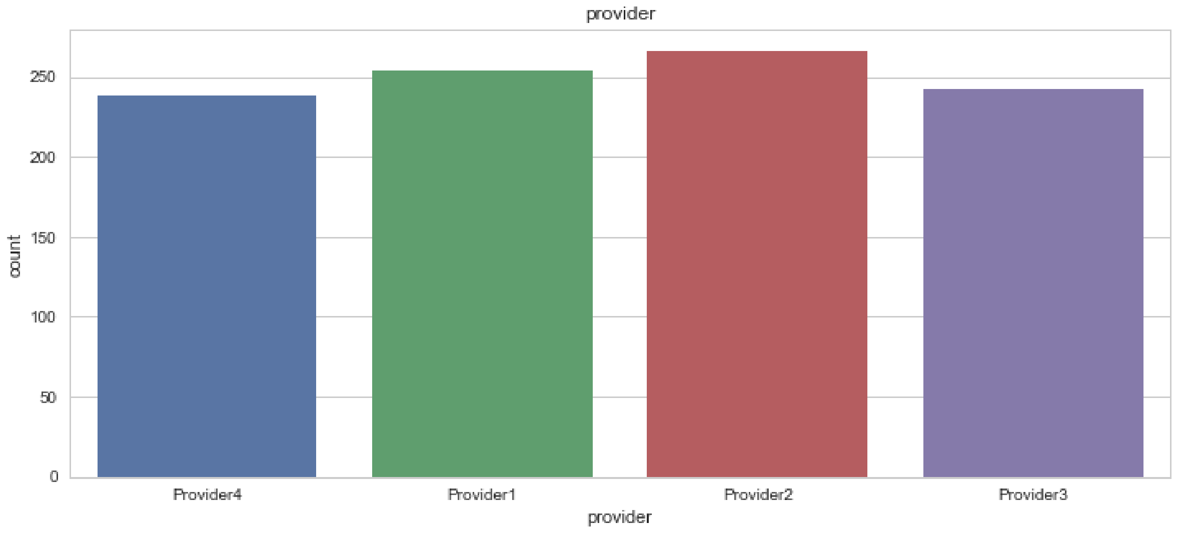
```
##Most of the pressure lies between 60 to 140.
##Excpet for one machine pressure is below 200.
## Most of the temperature lies between 60-140.
```

In [296]:

```
for col in cats:
    plt.figure(figsize=(12,5))
    sns.countplot(df[col])
    #plt.xlabel("No of customers")
    #plt.ylabel(col)
    plt.title(col)
    plt.show()
```







In [ ]:

```

```

In [ ]:

```

```

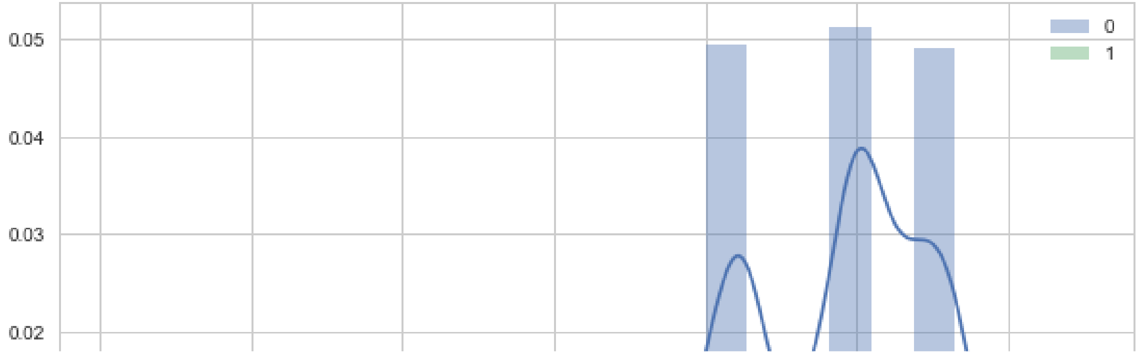
Bi-varient Analysis

In [297]:

```
#lifetime vs getting damaged
#numerical vs categorical

plt.figure(figsize=(10,5))
sns.distplot(df.lifetime[df.broken==1])
sns.distplot(df.lifetime[df.broken==0])
plt.legend(['0','1'])
plt.show()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462:
UserWarning: The 'normed' kwarg is deprecated, and has been replaced by th
e 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462:
UserWarning: The 'normed' kwarg is deprecated, and has been replaced by th
e 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```



Note : Lifetime is an affecting factor as we can see machines with lifetime in range 60-100 are getting damaged

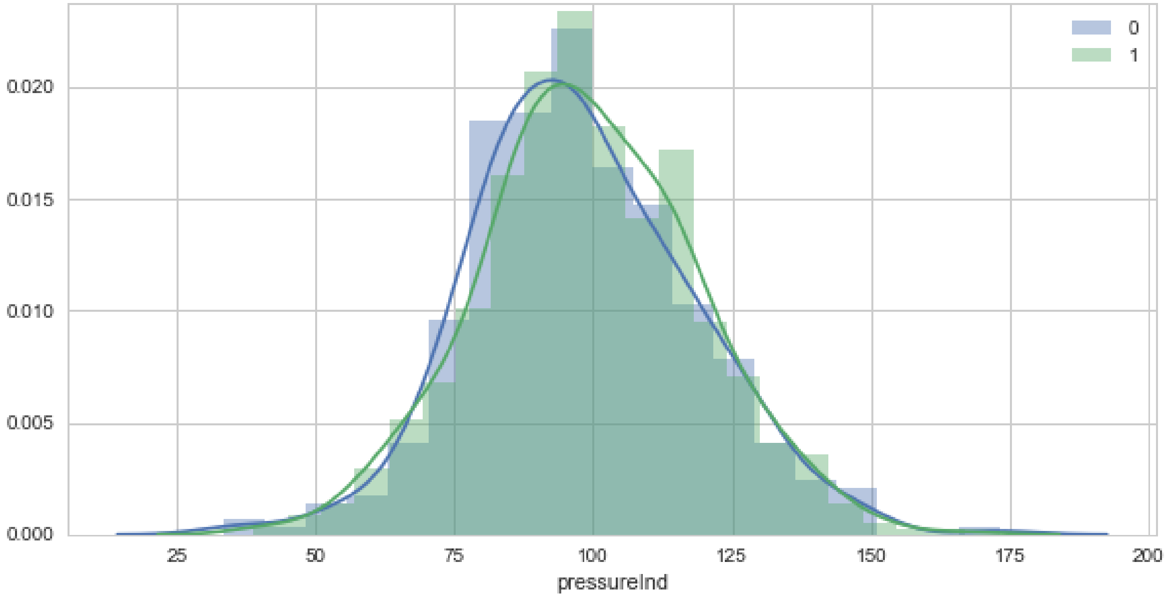
In [298]:

```

#numerical vs categorical

plt.figure(figsize=(10,5))
sns.distplot(df.pressureInd[df.broken==1])
sns.distplot(df.pressureInd[df.broken==0])
plt.legend(['0','1'])
plt.show()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: Us
erWarning: The 'normed' kwarg is deprecated, and has been replaced by the 'd
ensity' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: Us
erWarning: The 'normed' kwarg is deprecated, and has been replaced by the 'd
ensity' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```



Note : pressure is not an affecting factor.

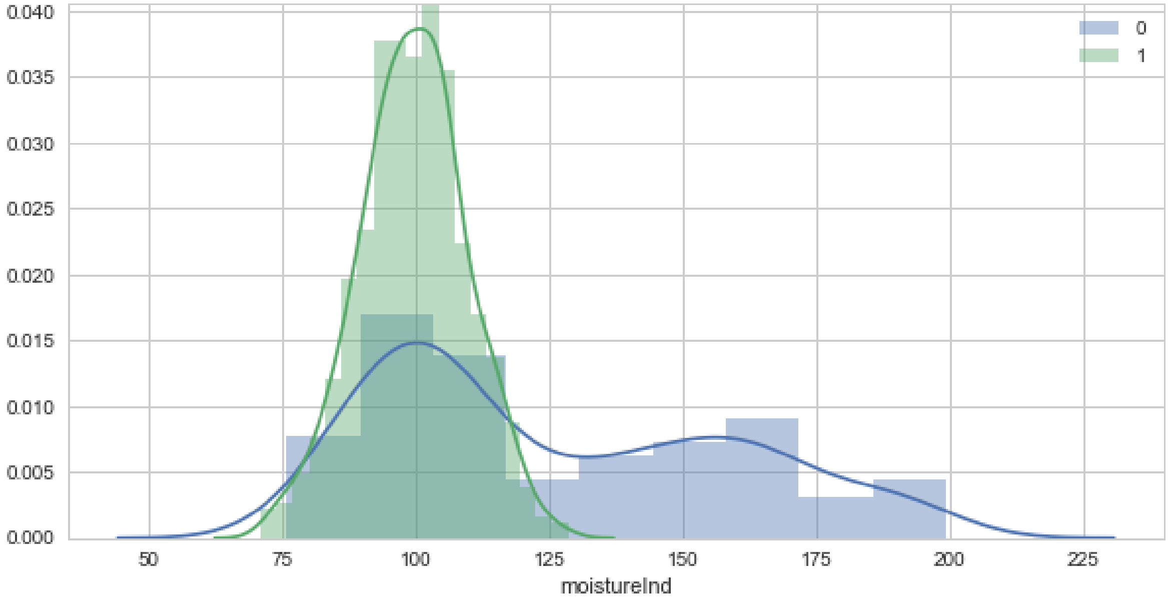
In [299]:

```

#numerical vs categorical

plt.figure(figsize=(10,5))
sns.distplot(df.moistureInd[df.broken==1])
sns.distplot(df.moistureInd[df.broken==0])
plt.legend(['0','1'])
plt.show()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: Us
erWarning: The 'normed' kwarg is deprecated, and has been replaced by the 'd
ensity' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: Us
erWarning: The 'normed' kwarg is deprecated, and has been replaced by the 'd
ensity' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```



Note: Moisture is an affecting factor.

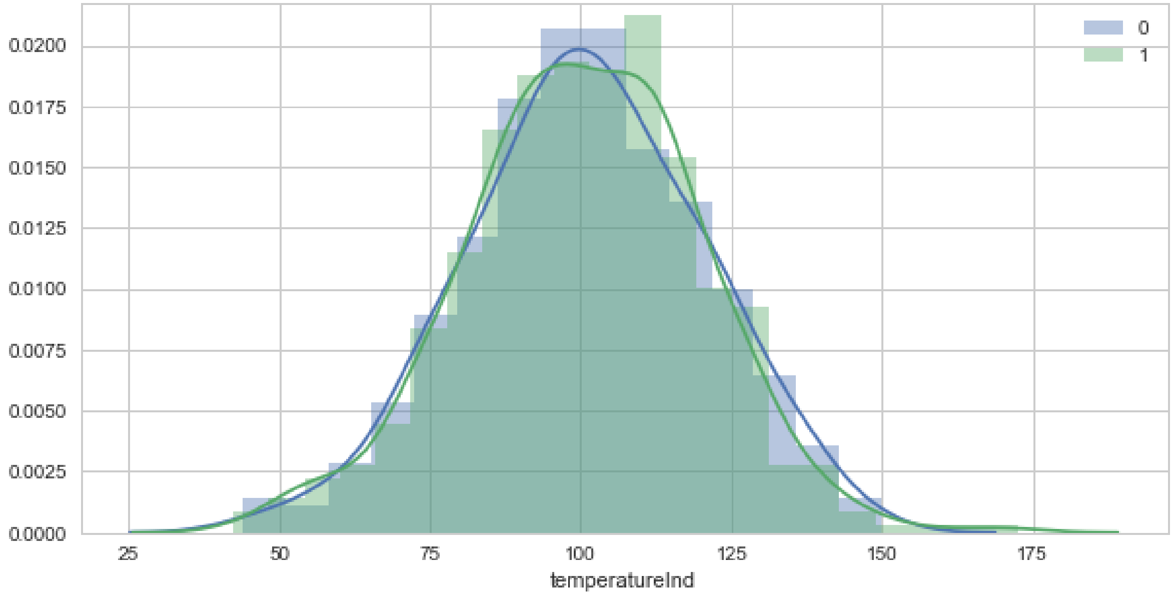
In [300]:

```


#numerical vs categorical

plt.figure(figsize=(10,5))
sns.distplot(df.temperatureInd[df.broken==1])
sns.distplot(df.temperatureInd[df.broken==0])
plt.legend(['0','1'])
plt.show()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: Us
erWarning: The 'normed' kwarg is deprecated, and has been replaced by the 'd
ensity' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: Us
erWarning: The 'normed' kwarg is deprecated, and has been replaced by the 'd
ensity' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```



Note: Temperature is not an affecting factor.

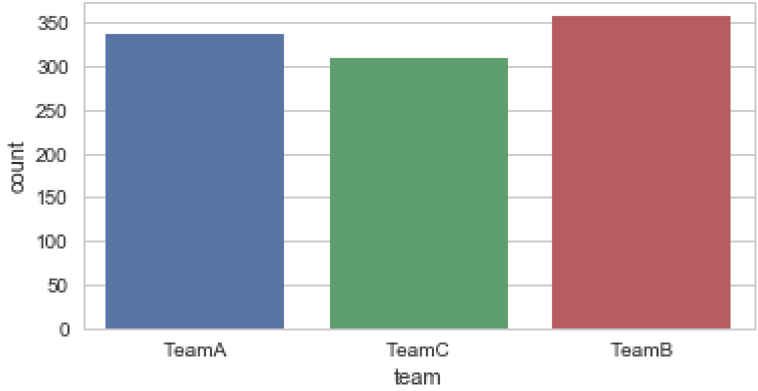
In [301]:

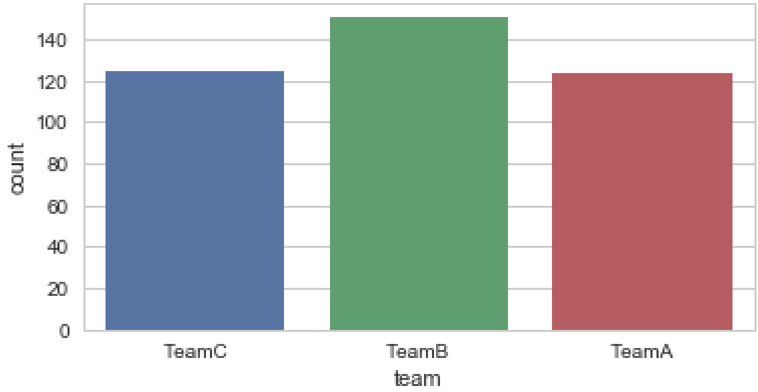
```
#categorical vs categorical

plt.figure(figsize=(6,3))
sns.countplot(df["team"])

plt.show()
plt.figure(figsize=(6,3))
sns.countplot(df["team"][df.broken==1])
plt.show()


```





Note : From the above comparison, it could be concluded the machine is broken irrespective of the team.

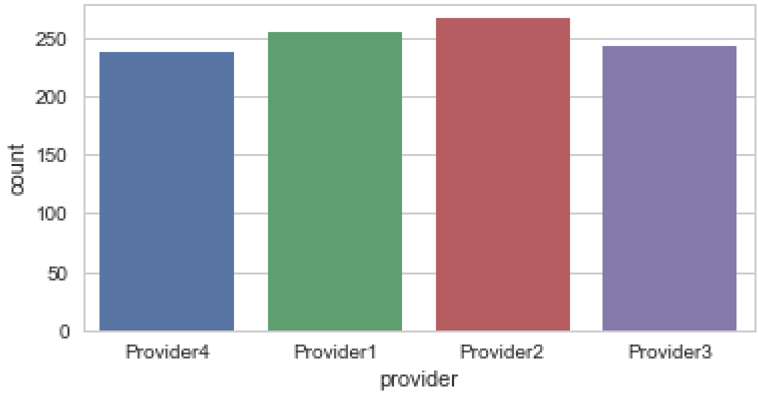
In [302]:

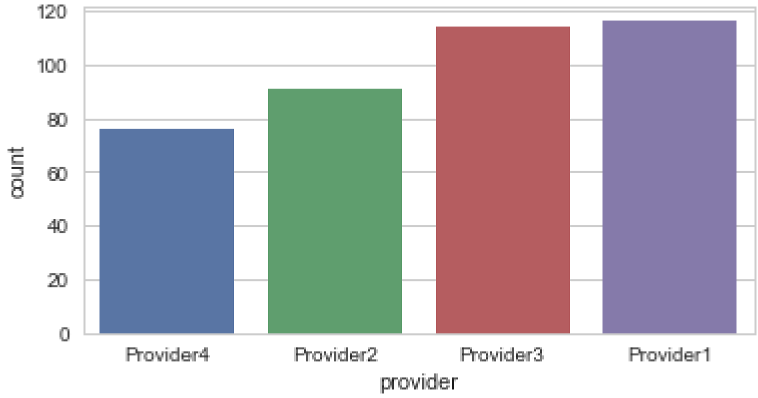
```
#categorical vs categorical

plt.figure(figsize=(6,3))
sns.countplot(df["provider"])

plt.show()
plt.figure(figsize=(6,3))
sns.countplot(df["provider"][df.broken==1])
plt.show()


```

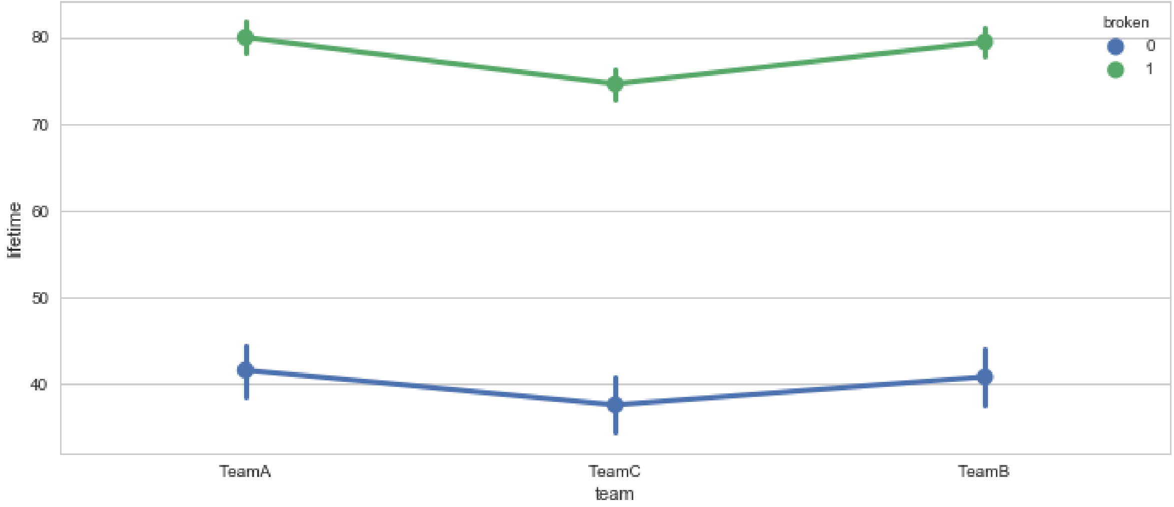




Note : Provider 4 has less number of damaged machines in contrast to other providers.

In [303]:

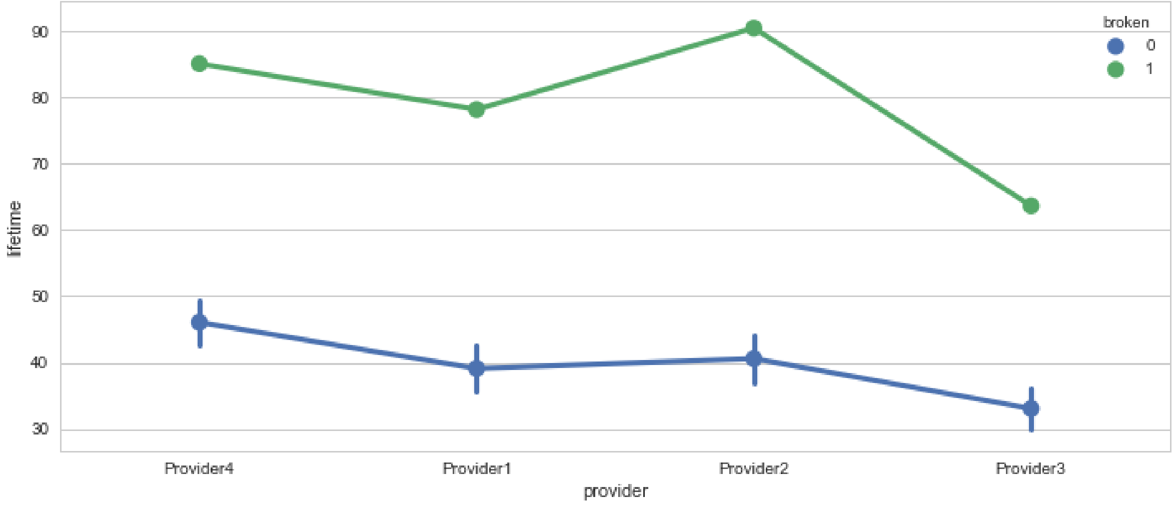
```
plt.figure(figsize=(12,5))
sns.pointplot(y='lifetime',x='team',hue='broken',data=df)
plt.show()
```



Note:

In [304]:

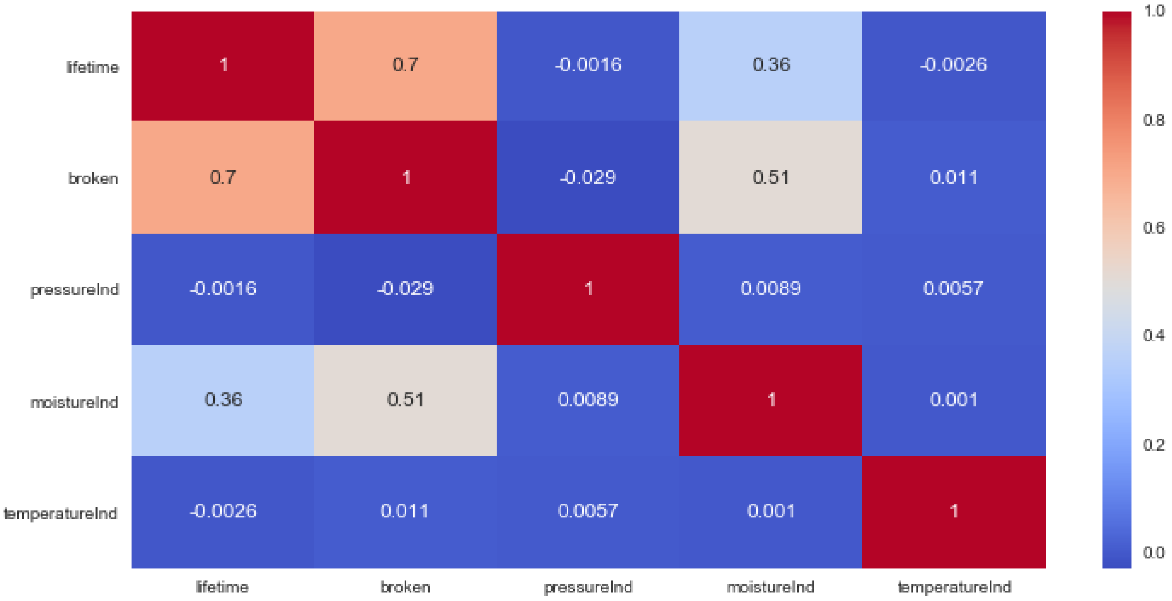
```
plt.figure(figsize=(12,5))
sns.pointplot(y='lifetime',x='provider',hue='broken',data=df)
plt.show()
```



Note:

In [305]:

```
cor = df.corr()
plt.figure(figsize=(12,6))
sns.heatmap(cor,annot=True,cmap='coolwarm')
plt.show()
```

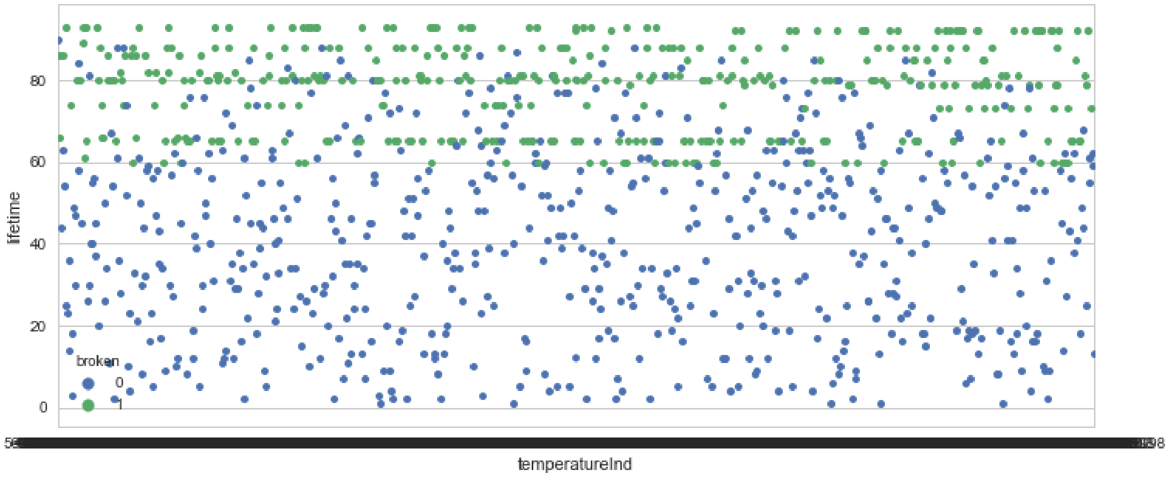


In [306]:

```
plt.figure(figsize=(12,5))
ax = sns.swarmplot(x='temperatureInd',y='lifetime',hue='broken',data=df)
ax
```

Out[306]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1efeff7db00>
```

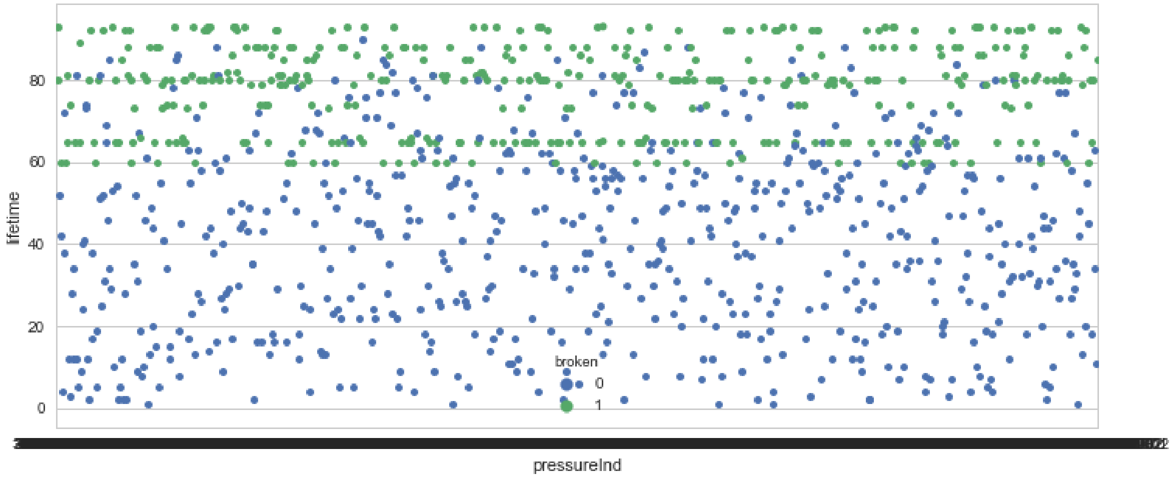


In [307]:

```python
plt.figure(figsize=(12,5))
ax = sns.swarmplot(x='pressureInd',y='lifetime',hue='broken',data=df)
ax
```

Out[307]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1efefffaf28>
```

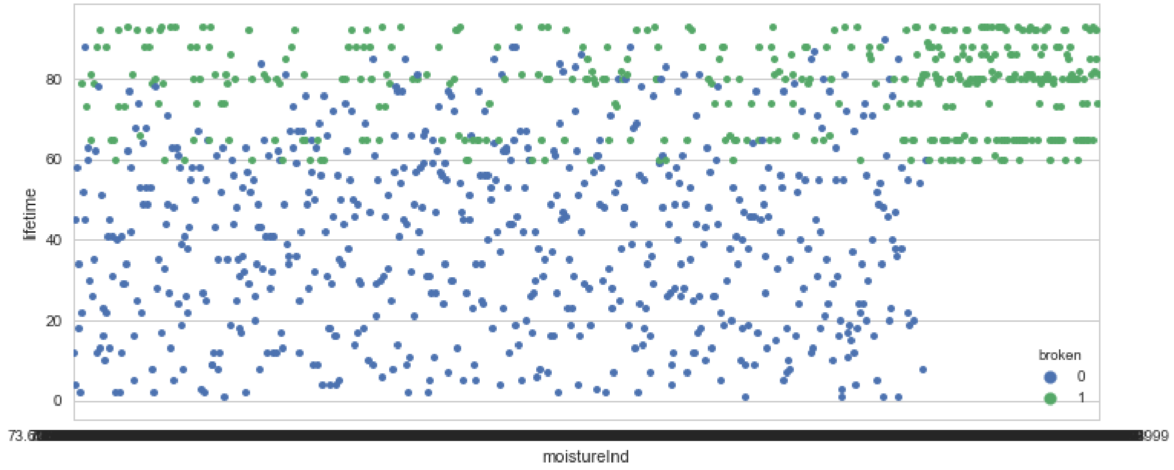


In [308]:

```python
plt.figure(figsize=(12,5))
ax = sns.swarmplot(x='moistureInd',y='lifetime',hue='broken',data=df)
ax
```

Out[308]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1efeed74400>
```



Note : From the above plots, it can be observed that lifetime plays a major role in damage of machines. Machine between age of 60-80 are more likely to get damaged.

Conclusion :

From all the above graphs and the various univarient, multivarient and bivarient analysis it can be concluded that the lifetime of machines is an afffecting factor why machines are getting damaged. It also seems plausible and logical for machines to have wear and tear, and damaged, as they are used for more duration. Machines between lifetime 60 and above are likely to get damaged/broken. It is requried to take good care of machines after the age of around 60 to prevent more loss.

In [ ]:

```

```