In [1]:

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

In [2]:

```
df = pd.read_csv('https://raw.githubusercontent.com/anshupandey/Machine_Learning_Training/m
```

In [3]:

```
df
```

Out[3]:

	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
5	30	0	87.678801	115.712262	89.792105	TeamA	Provider1
6	68	0	94.614174	85.702236	142.827001	TeamB	Provider2
7	65	1	96.483303	193.046797	98.316190	TeamB	Provider3
8	23	0	105.486158	118.291997	96.028822	TeamB	Provider2
9	81	1	99.178235	199.138717	95.492965	TeamC	Provider4
10	38	0	97.817844	111.074168	94.942443	TeamB	Provider4

In [4]:

```
df.head()
```

Out[4]:

	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

data selection

In [5]:

```
df.describe()
```

Out[5]:

	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

In [6]:

```
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 [7]:

```
df.shape
```

Out[7]:

```
(1000, 7)
```

In [8]:

```
df.columns
```

Out[8]:

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

In [9]:

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

Out[9]:

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

In [10]:

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

Out[10]:

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

Data Cleaning

In [11]:

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

Out[11]:

```
0
```

In [12]:

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

Out[12]:

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

In [13]:

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

In [14]:

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

In [15]:

```
df.skew()
```

Out[15]:

```
lifetime          -0.407597
broken             0.421663
pressureInd        0.117776
moistureInd       15.982324
temperatureInd    -0.070945
dtype: float64
```

Data Visualization

In [16]:

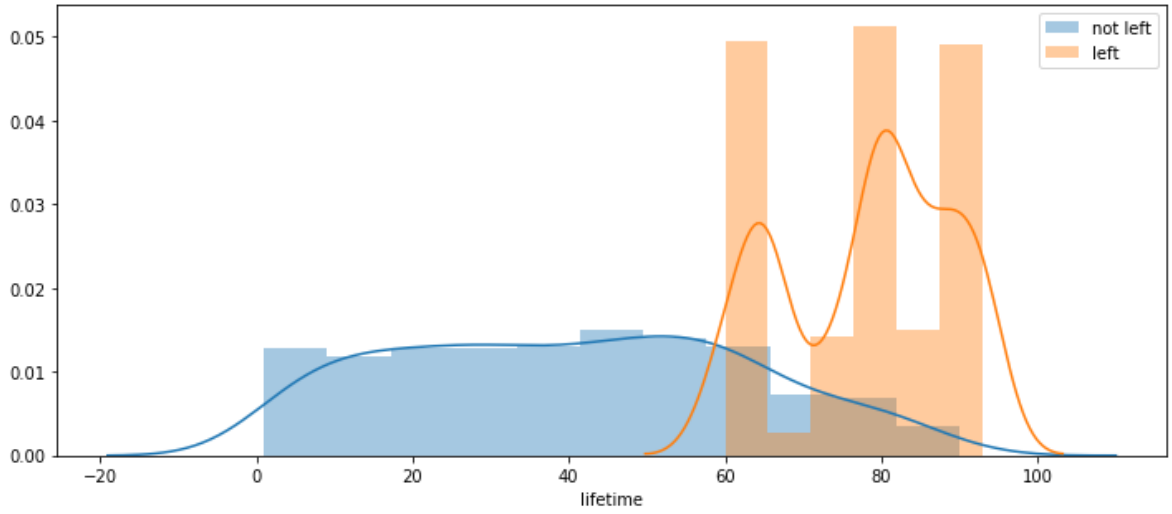
```
df.columns
```

Out[16]:

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

In [17]:

```
   #numerical v/s catogorical
   #lifetime v/s broken
plt.figure(figsize=(12,5))
sns.distplot(df.lifetime[df.broken==0])
sns.distplot(df.lifetime[df.broken==1])
plt.legend(['not left','left'])
plt.show()
```

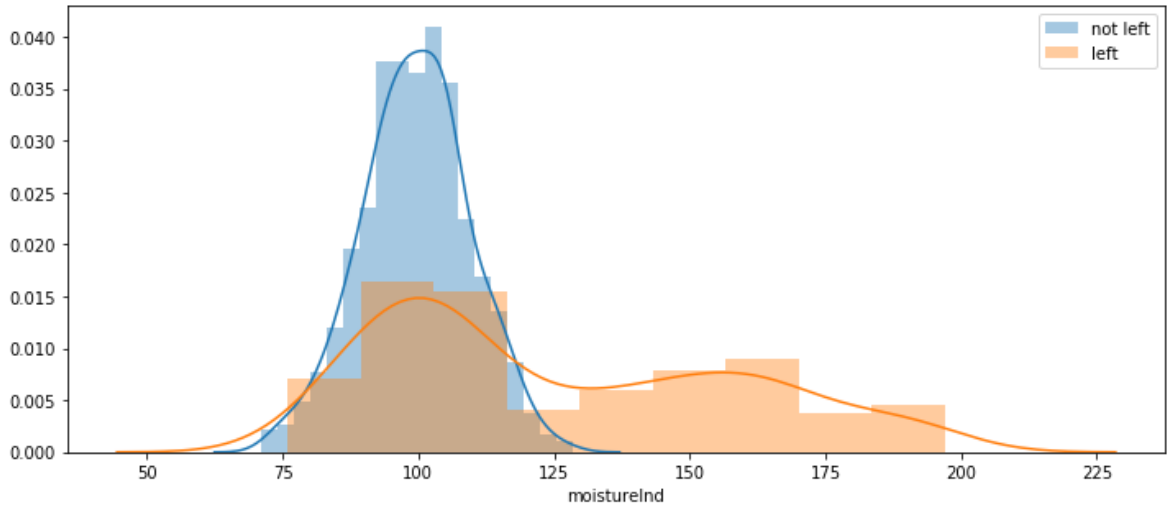


In [18]:

```
#machine get broken after the 60 months lifetime more before that it woks good
```

In [34]:

```
   #numerical v/s catogorical
   #moistureInd v/s broken
plt.figure(figsize=(12,5))
sns.distplot(df.moistureInd[df.broken==0])
sns.distplot(df.moistureInd[df.broken==1])
plt.legend(['not left','left'])
plt.show()
```

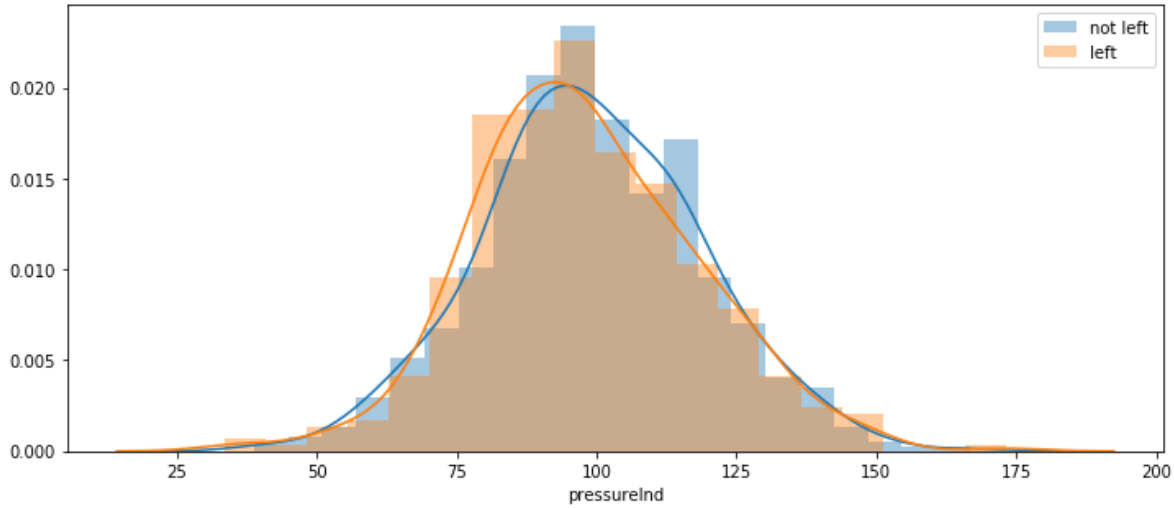


In [20]:

```
#when the moisture is more then around 112 the machine starts damaging before that machine
```

In [21]:

```
   #numerical v/s catogorical
   #pressureInd v/s broken
plt.figure(figsize=(12,5))
sns.distplot(df.pressureInd[df.broken==0])
sns.distplot(df.pressureInd[df.broken==1])
plt.legend(['not left','left'])
plt.show()
```



In [22]:

```
#can't be said while compairing the pressureInd and broken
```

In [32]:

```
df.moistureInd.quantile(0.99)
```

Out[32]:

192.31769522800002

In [33]:

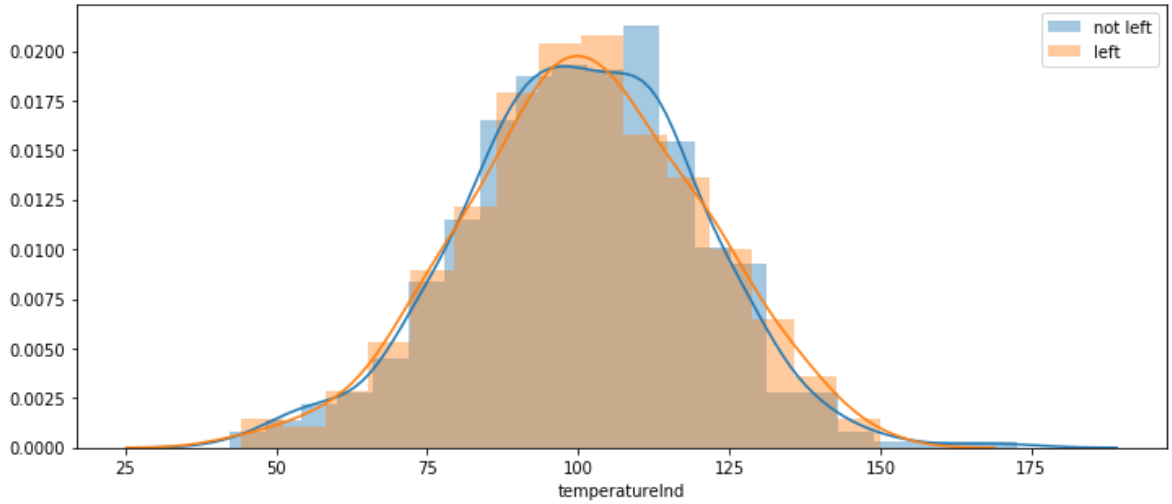
```
df=df[df['moistureInd']<=df['moistureInd'].quantile(0.999)]
df.shape
```

Out[33]:

(998, 7)

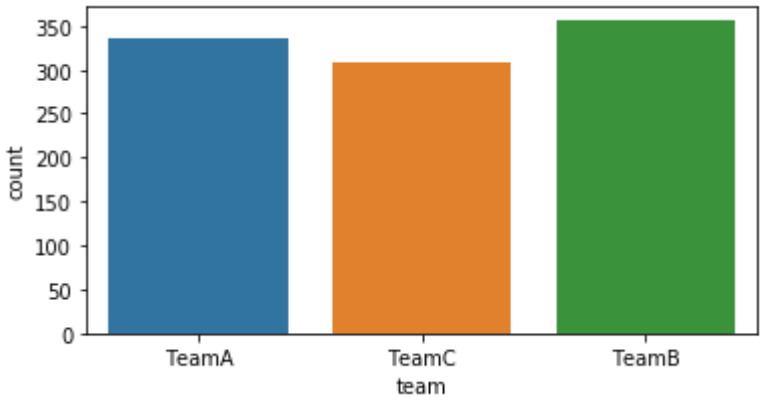
In [25]:

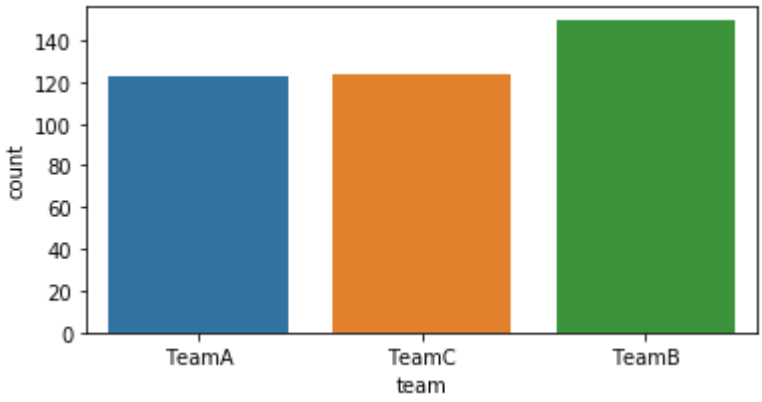
```
   #numerical v/s categorical
   #temperatureIn v/s broken
plt.figure(figsize=(12,5))
sns.distplot(df.temperatureInd[df.broken==0])
sns.distplot(df.temperatureInd[df.broken==1])
plt.legend(['not left','left'])
plt.show()
```



In [26]:

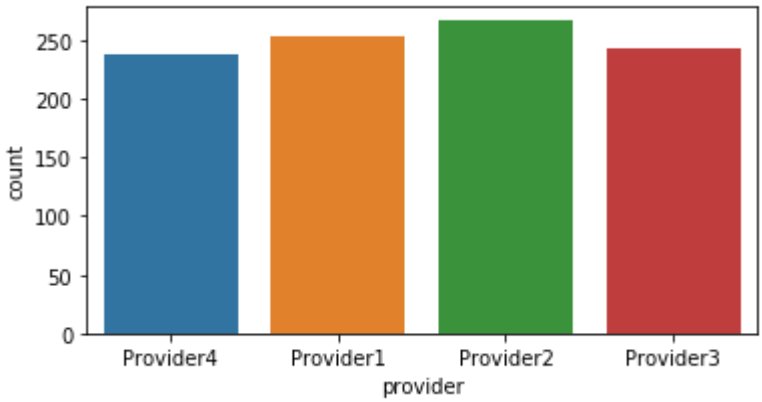
```
#Categorical v/s Categorical
# team       v/s  broken
plt.figure(figsize=(6,3))
sns.countplot(df.team,order=df.team.unique())
plt.show()
plt.figure(figsize=(6,3))
sns.countplot(df.team[df.broken==1],order=df.team.unique())
plt.show()
```





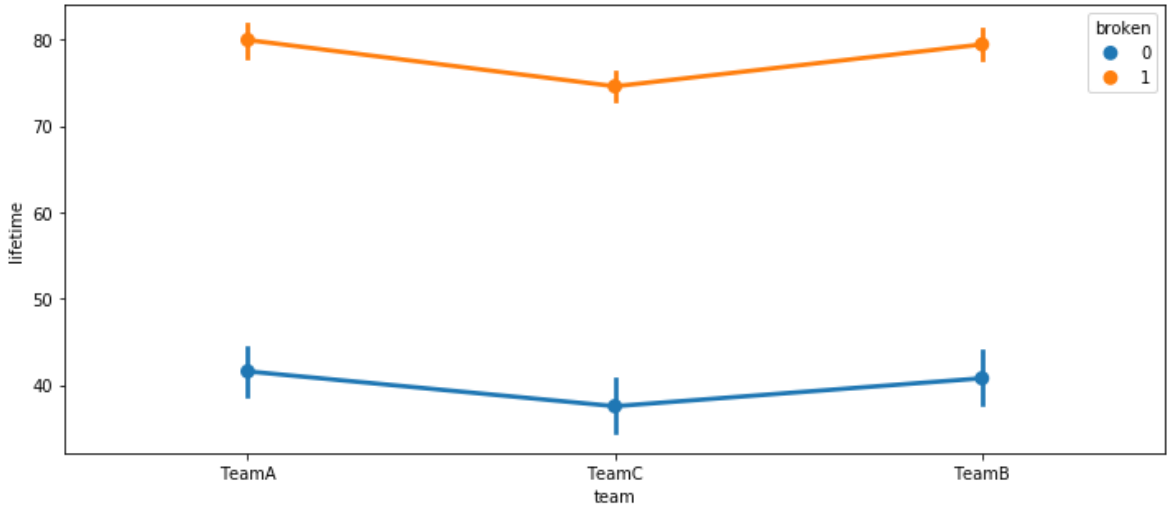
In [27]:

```
#Categorical v/s Categorical
# provider      v/s  broken
plt.figure(figsize=(6,3))
sns.countplot(df.provider,order=df.provider.unique())
plt.show()
plt.figure(figsize=(6,3))
sns.countplot(df.provider[df.broken==1],order=df.provider.unique())
plt.show()
```



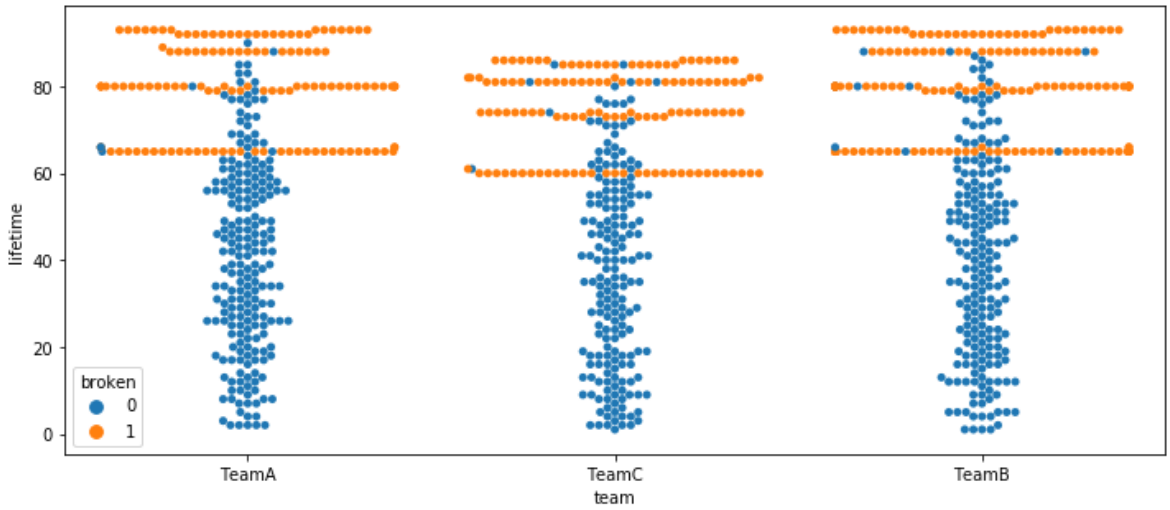
In [28]:

```
#point plot
#categorical  v/s numerical  v/s categorical
#x,y,hue>>x = categorical,y=Numeric ,hue =categorical
plt.figure(figsize=(12,5))
sns.pointplot(x='team',y='lifetime',hue='broken',data=df)
plt.show()
```



In [29]:

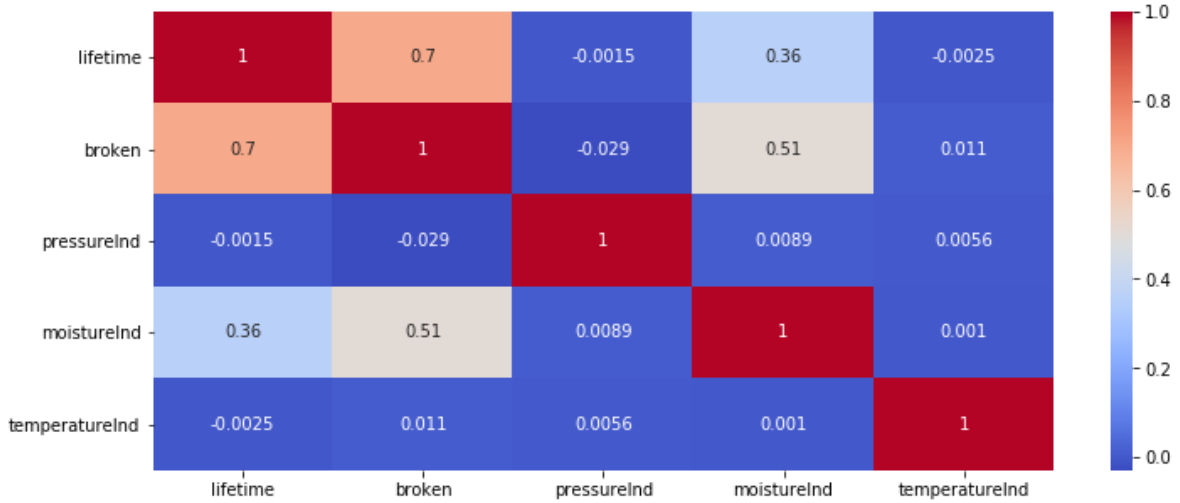
```
#swarm plot
#categorical  v/s numerical  v/s categorical
#x,y,hue>>x = categorical,y=Numeric ,hue =categorical
plt.figure(figsize=(12,5))
sns.swarmplot(x='team',y='lifetime',hue='broken',data=df)
plt.show()
```



Corralation

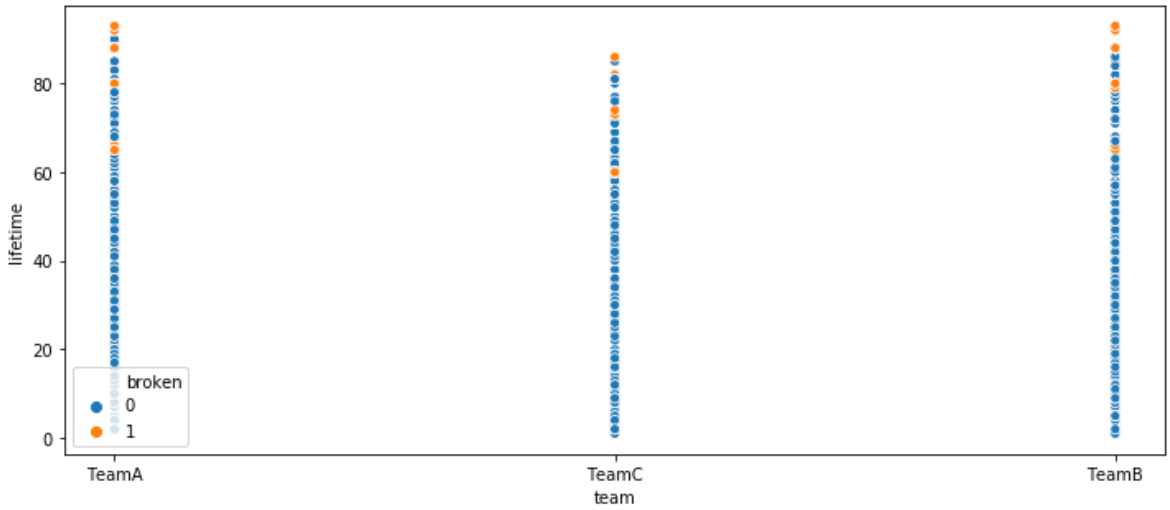
In [30]:

```
cor=df.corr()
#heatmap for visualization correlation  analysis
plt.figure(figsize=(12,5))
sns.heatmap(cor,annot=True,cmap='coolwarm')
plt.show()
```



In [31]:

```
#scatter polt
#x,y,hue>>x = categorical,y=Numeric ,hue =categorical
plt.figure(figsize=(12,5))
sns.scatterplot(x='team',y='lifetime',hue='broken',data=df)
plt.show()
```



Report :-

As the machine gets old it is likely to get broken, temperature pressure does not play important role but humidity plays important role in machines likely to get broken. machines supplied by provider 1 and 3 are getting more damage. team A or B or C machines are getting damaged when thier lifetime exceeds over 60. pressure id is highly corelated with broken moisture id is highly corelated with pressureINd temperature id is highly corelated with moistureind and pressureid. lifetime of mchines handled by team C is less ie methods followed by team C is less efficient

In [ ]:

```

```