Import Necessary Libraries

In [2]:
```
import pandas as pd
import numpy as np
from sklearn import preprocessing
from sklearn import metrics
import seaborn as sns
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from matplotlib import pyplot as plt
from sklearn.decomposition import PCA
from mlxtend.plotting import plot_decision_regions
```

Business Problem

Classify the Size Category using SVM

Data collection

In [5]:
```
forest=pd.read_csv('forestfires (1).csv',sep=',')
forest
```

Out[5]:

	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	...	monthfeb	monthjan	monthj
0	mar	fri	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0	...	0	0	
1	oct	tue	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	...	0	0	
2	oct	sat	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	...	0	0	
3	mar	fri	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	...	0	0	
4	mar	sun	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	...	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
512	aug	sun	81.6	56.7	665.6	1.9	27.8	32	2.7	0.0	...	0	0	
513	aug	sun	81.6	56.7	665.6	1.9	21.9	71	5.8	0.0	...	0	0	
514	aug	sun	81.6	56.7	665.6	1.9	21.2	70	6.7	0.0	...	0	0	
515	aug	sat	94.4	146.0	614.7	11.3	25.6	42	4.0	0.0	...	0	0	
516	nov	tue	79.5	3.0	106.7	1.1	11.8	31	4.5	0.0	...	0	0	

517 rows × 31 columns

In [6]:
```
forest.head()
```

Out[6]:

	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	...	monthfeb	monthjan	monthjul
0	mar	fri	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0	...	0	0	0
1	oct	tue	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	...	0	0	0
2	oct	sat	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	...	0	0	0
3	mar	fri	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	...	0	0	0
4	mar	sun	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	...	0	0	0

5 rows × 31 columns

In [8]:
```
forest.shape
```

Out[8]:
```
(517, 31)
```

In [10]:
```
forest[forest.columns[0:11]].describe().T
```

Out[10]:

	count	mean	std	min	25%	50%	75%	max
FFMC	517.0	90.644681	5.520111	18.7	90.2	91.60	92.90	96.20
DMC	517.0	110.872340	64.046482	1.1	68.6	108.30	142.40	291.30
DC	517.0	547.940039	248.066192	7.9	437.7	664.20	713.90	860.60
ISI	517.0	9.021663	4.559477	0.0	6.5	8.40	10.80	56.10
temp	517.0	18.889168	5.806625	2.2	15.5	19.30	22.80	33.30
RH	517.0	44.288201	16.317469	15.0	33.0	42.00	53.00	100.00
wind	517.0	4.017602	1.791653	0.4	2.7	4.00	4.90	9.40
rain	517.0	0.021663	0.295959	0.0	0.0	0.00	0.00	6.40
area	517.0	12.847292	63.655818	0.0	0.0	0.52	6.57	1090.84

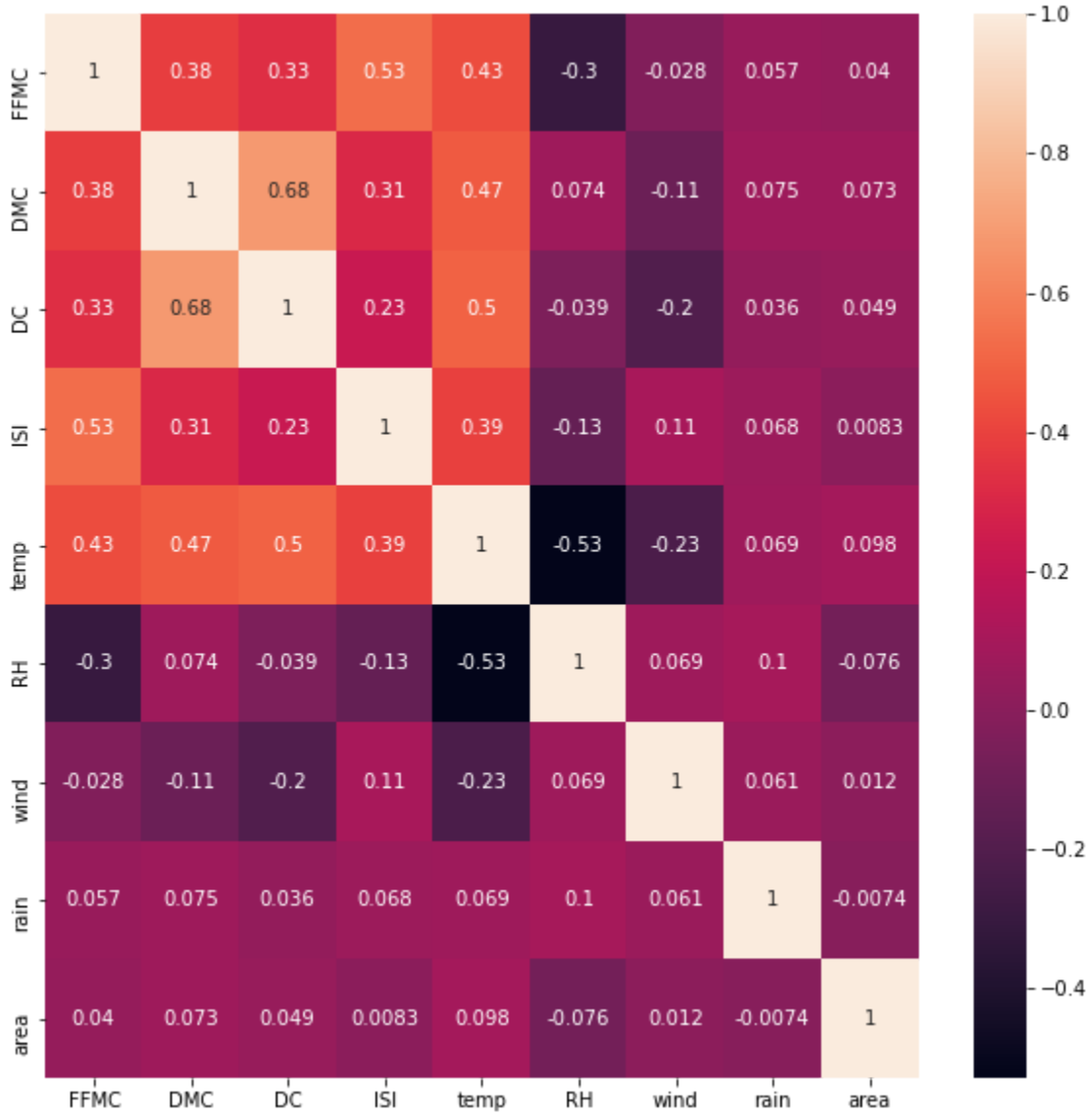
In [11]:
```
forest[forest.columns[0:11]].isnull().sum()
```

Out[11]:
```
month    0
day      0
FFMC     0
DMC      0
DC       0
ISI      0
temp     0
RH       0
wind     0
rain     0
area     0
dtype: int64
```

In [12]:
```
corr = forest[forest.columns[0:11]].corr()
```

In [13]:
```
plt.figure(figsize=(10,10))
sns.heatmap(corr,annot=True)
```

Out[13]:
```
<AxesSubplot:>
```



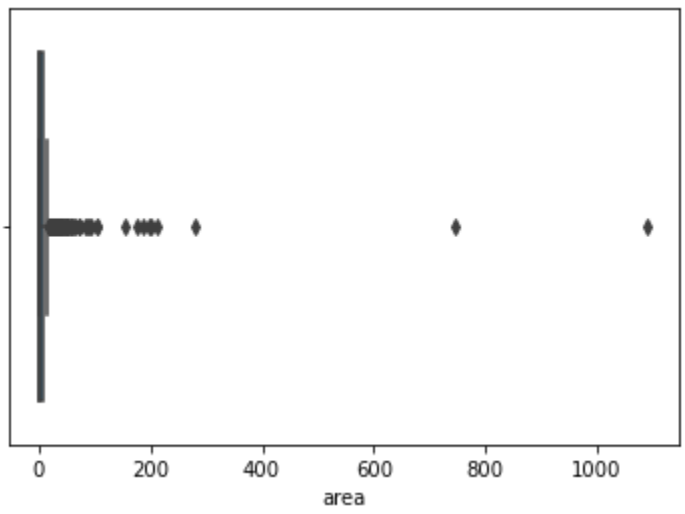
In [15]:

```
import warnings
warnings.filterwarnings('ignore')
```

Outlier check

In [16]:

```
ax = sns.boxplot(forest['area'])
```

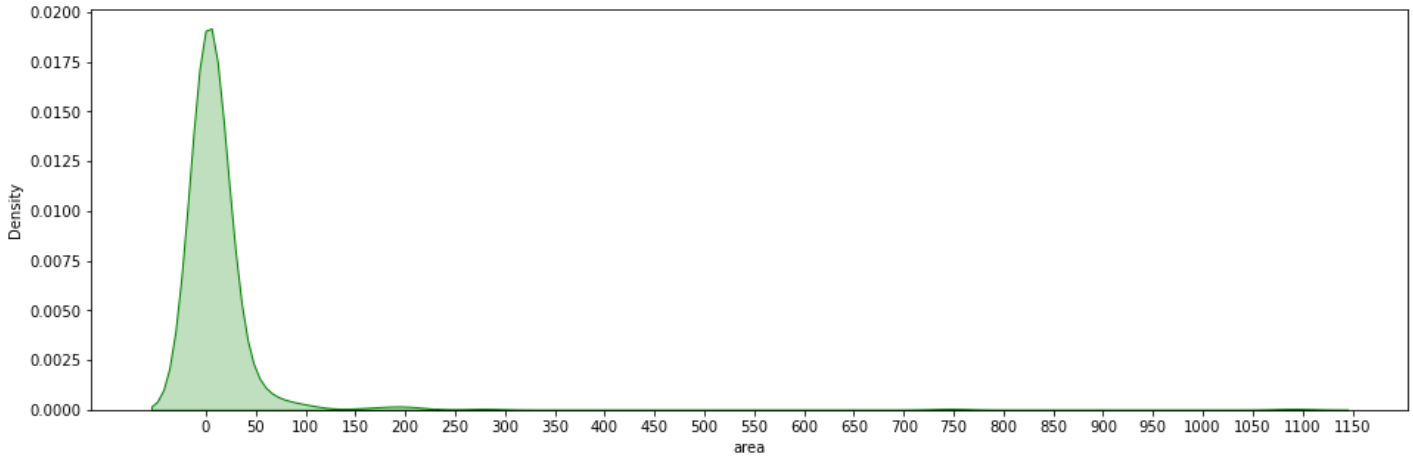


There are 3 outliers in data

In [17]:
```
plt.rcParams["figure.figsize"] = 9,5
```

In [18]:
```
plt.figure(figsize=(16,5))
print("Skew: {}".format(forest['area'].skew()))
print("Kurtosis: {}".format(forest['area'].kurtosis()))
ax = sns.kdeplot(forest['area'],shade=True,color='g')
plt.xticks([i for i in range(0,1200,50)])
plt.show()
```

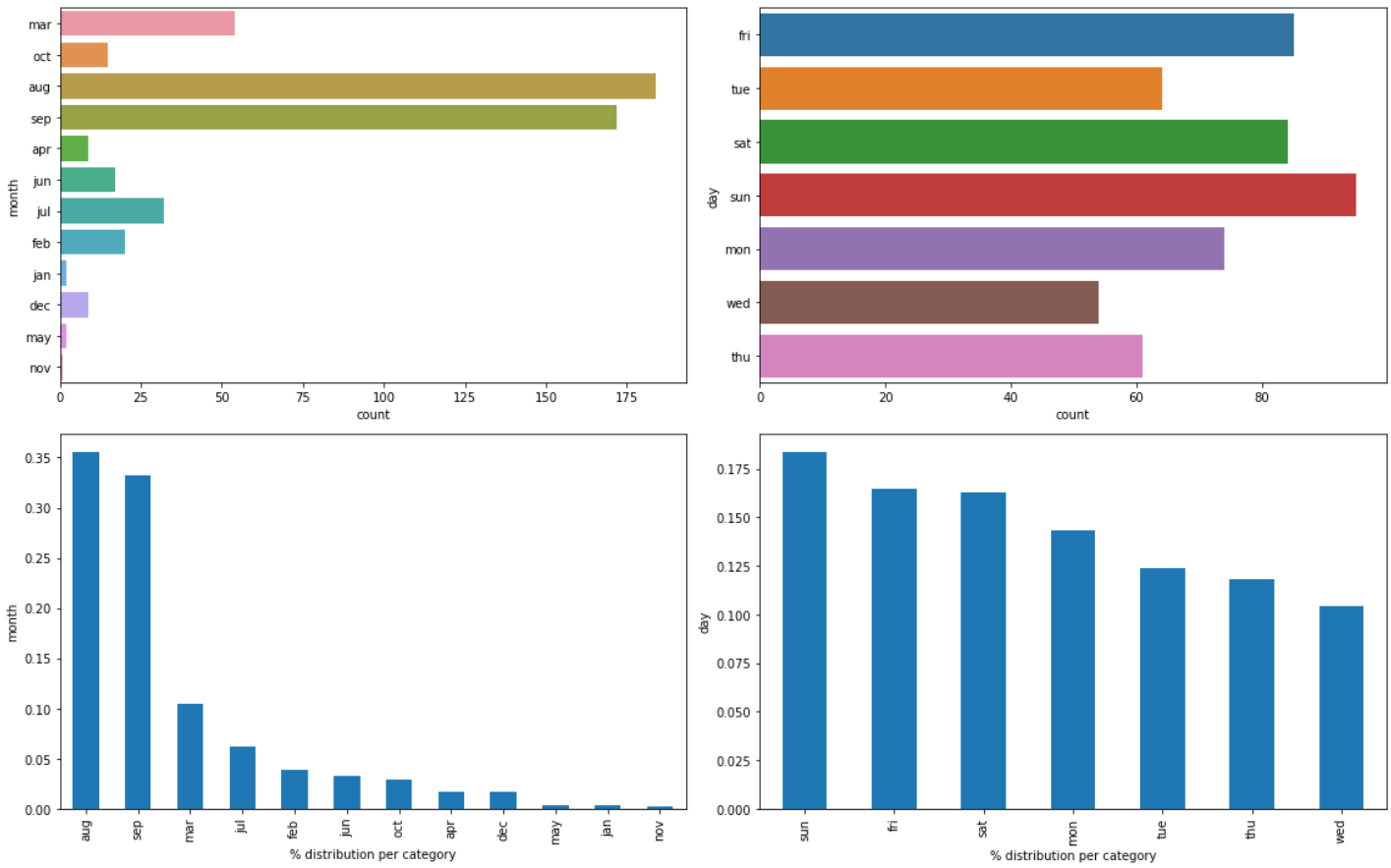
```
Skew: 12.846933533934868
Kurtosis: 194.1407210942299
```



The Data is highly Skewed and has large kurtosis value

In [19]:
```
dfa = forest[forest.columns[0:10]]
month_colum = dfa.select_dtypes(include='object').columns.tolist()
```

In [20]:
```
plt.figure(figsize=(16,10))
for i,col in enumerate(month_colum,1):
    plt.subplot(2,2,i)
    sns.countplot(data=dfa,y=col)
    plt.subplot(2,2,i+2)
    forest[col].value_counts(normalize=True).plot.bar()
    plt.ylabel(col)
    plt.xlabel('% distribution per category')
plt.tight_layout()
plt.show()
```

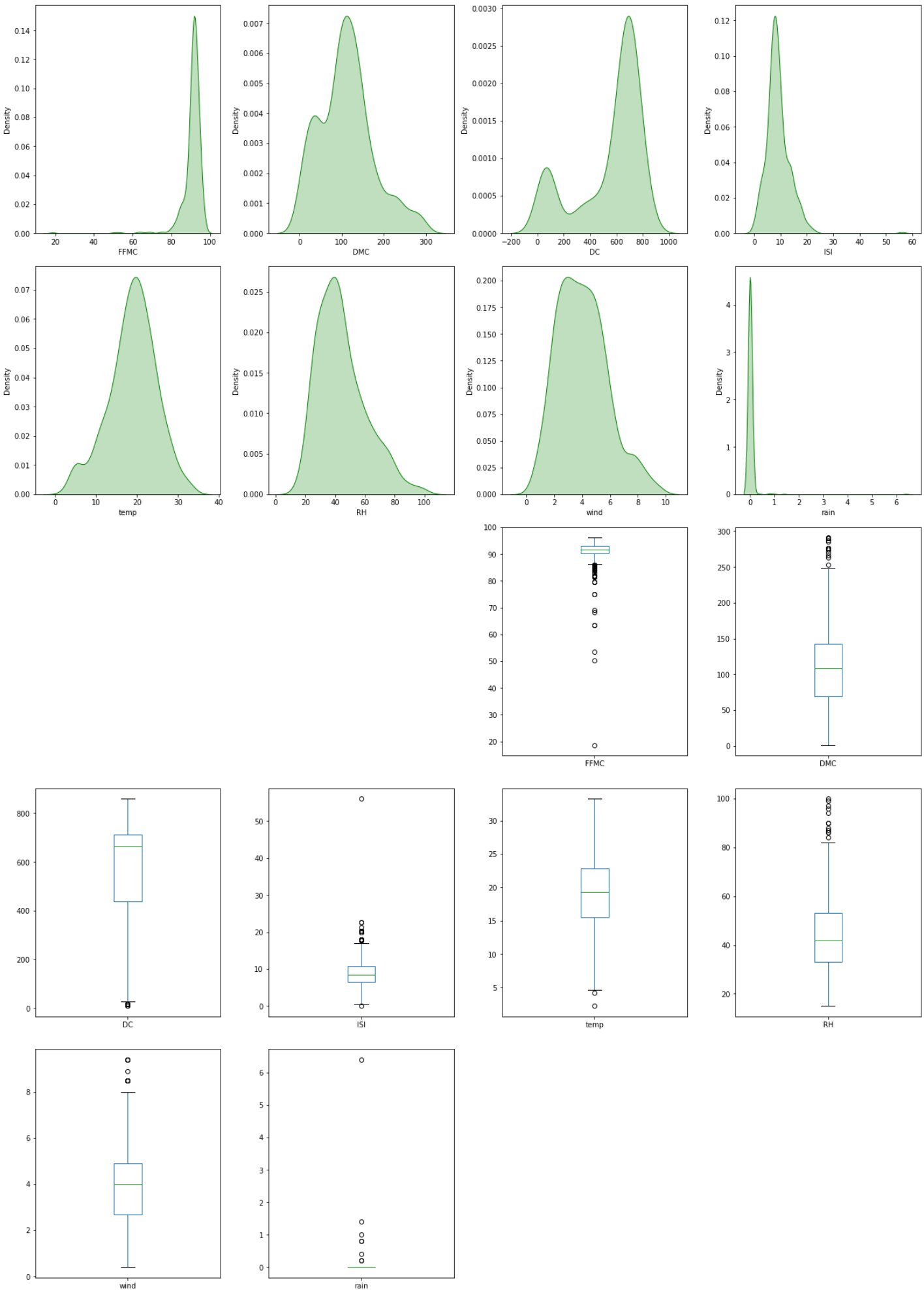


In [21]:

```
num_columns = dfa.select_dtypes(exclude='object').columns.tolist()
```

In [23]:

```
plt.figure(figsize=(18,40))
for i,col in enumerate(num_columns,1):
    plt.subplot(8,4,i)
    sns.kdeplot(forest[col],color='g',shade=True)
    plt.subplot(8,4,i+10)
    forest[col].plot.box()
plt.tight_layout()
plt.show()
num_data = forest[num_columns]
pd.DataFrame(data=[num_data.skew(),num_data.kurtosis()],index=['skewness','kurtosis'])
```



Out[23]:

	FFMC	DMC	DC	ISI	temp	RH	wind	rain
	6.575606	0.547498	-1.100445	2.536325	-0.331172	0.862904	0.571001	19.816344

	FFMC	DMC	DC	ISI	temp	RH	wind	rain
kurtosis	67.066041	0.204822	-0.245244	21.458037	0.136166	0.438183	0.054324	421.295964

Support Vestor Machine

In [25]:
```
X = forest.iloc[:,2:30]
y = forest.iloc[:,30]
```

In [26]:
```
mapping = {'small': 1, 'large': 2}
```

In [27]:
```
y = y.replace(mapping)
```

In [28]:
```
x_train,x_test,y_train,y_test = train_test_split(X,y,test_size =  0.20, stratify = y)
```

Linear

In [29]:
```
model_linear = SVC(kernel = "linear")
model_linear.fit(x_train,y_train)
pred_test_linear = model_linear.predict(x_test)
print("Accuracy:",metrics.accuracy_score(y_test, pred_test_linear))
```

```
Accuracy: 0.9903846153846154
```

Poly

In [30]:
```
model_poly = SVC(kernel = "poly")
model_poly.fit(x_train,y_train)
pred_test_poly = model_poly.predict(x_test)
print("Accuracy:",metrics.accuracy_score(y_test, pred_test_poly))
```

```
Accuracy: 0.7788461538461539
```

RBF

In [31]:
```
model_rbf = SVC(kernel = "rbf")
model_rbf.fit(x_train,y_train)
pred_test_rbf = model_rbf.predict(x_test)
print("Accuracy:",metrics.accuracy_score(y_test, pred_test_rbf))
```

```
Accuracy: 0.7596153846153846
```

Sigmoid

In [32]:
```
model_sigmoid = SVC(kernel = "sigmoid")
model_sigmoid.fit(x_train,y_train)
pred_test_sigmoid = model_sigmoid.predict(x_test)
print("Accuracy:",metrics.accuracy_score(y_test, pred_test_sigmoid))
```

```
Accuracy: 0.6826923076923077
```

In [36]:
```
ytt = y_train.to_numpy()
```

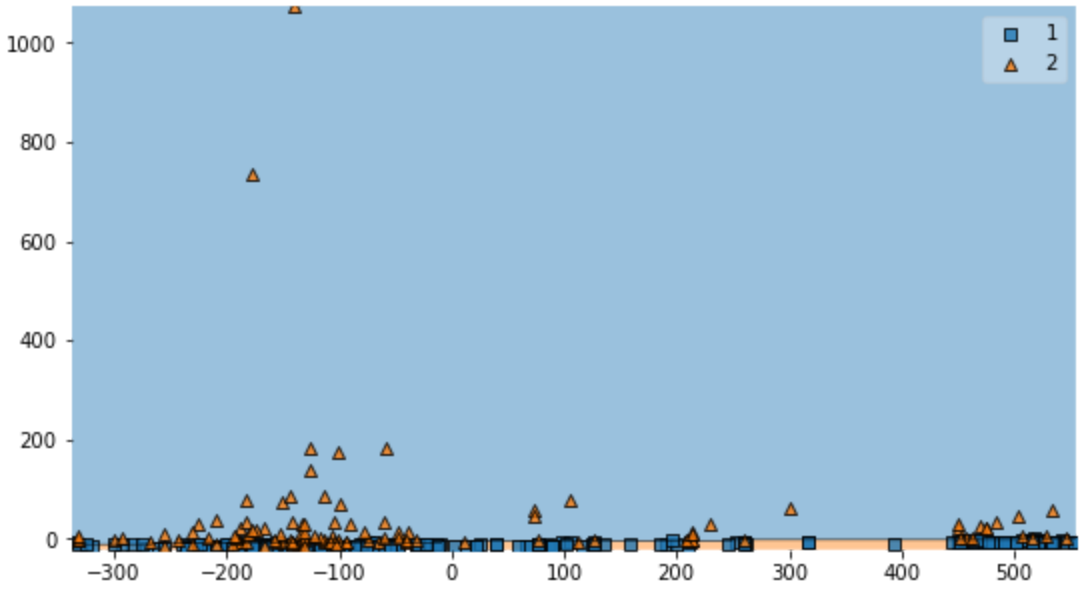
In [37]:
```
pca = PCA(n_components = 2)
```

In [38]:
```
x_train2 = pca.fit_transform(x_train)
```

In [40]:
```
model_linear.fit(x_train2,ytt)
```

Out[40]:
```
SVC(kernel='linear')
```

In [41]:
```
plot_decision_regions(x_train2,ytt, clf=model_linear)
plt.show()
```



In [ ]:
```

```