4/16/2021

ASSIGNMENT-3

Classification Algorithms

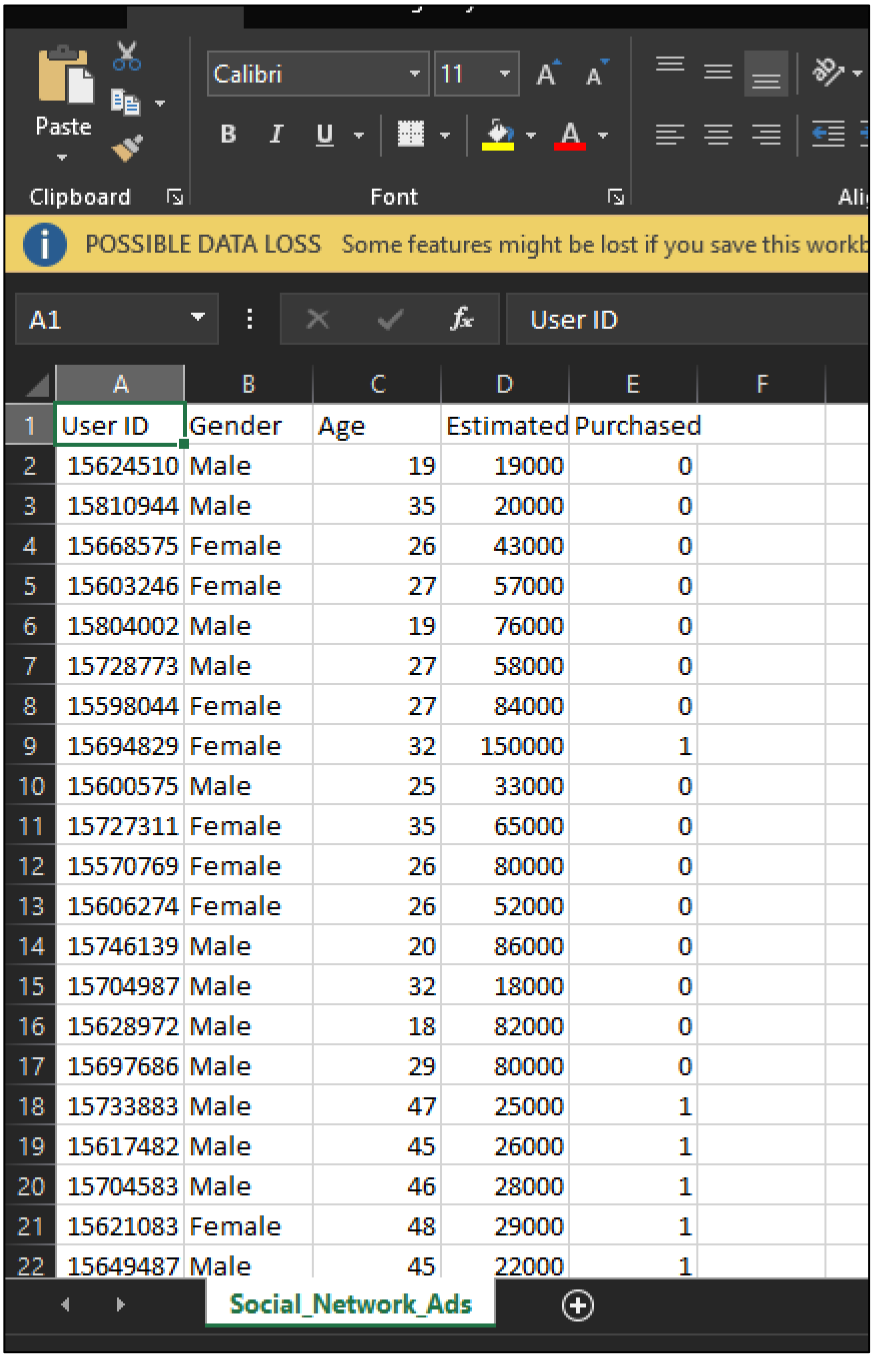
Name	Rishabh Sharma
Registration Number	20MAI0082

GitHub Link :- https://github.com/rishabh5197/Data-Mining/tree/main/Assignment-3

Dataset Link :- https://www.kaggle.com/rakeshrau/social-network-ads

Description About dataset

- There are 400 rows and 5 columns present in our dataset from which user_id is unique and the rest 3 columns are going to be independent data and the last one is going to be the dependent data which is purchased.



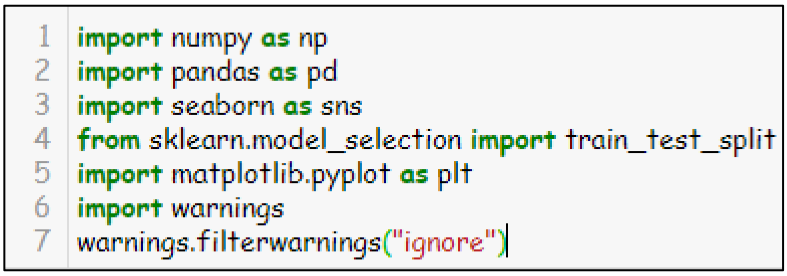
POSSIBLE DATA LOSS Some features might be lost if you save this workb

A1 User ID

	A	B	C	D	E	F
1	User ID	Gender	Age	Estimated	Purchased	
2	15624510	Male	19	19000	0	
3	15810944	Male	35	20000	0	
4	15668575	Female	26	43000	0	
5	15603246	Female	27	57000	0	
6	15804002	Male	19	76000	0	
7	15728773	Male	27	58000	0	
8	15598044	Female	27	84000	0	
9	15694829	Female	32	150000	1	
10	15600575	Male	25	33000	0	
11	15727311	Female	35	65000	0	
12	15570769	Female	26	80000	0	
13	15606274	Female	26	52000	0	
14	15746139	Male	20	86000	0	
15	15704987	Male	32	18000	0	
16	15628972	Male	18	82000	0	
17	15697686	Male	29	80000	0	
18	15733883	Male	47	25000	1	
19	15617482	Male	45	26000	1	
20	15704583	Male	46	28000	1	
21	15621083	Female	48	29000	1	
22	15649487	Male	45	22000	1	

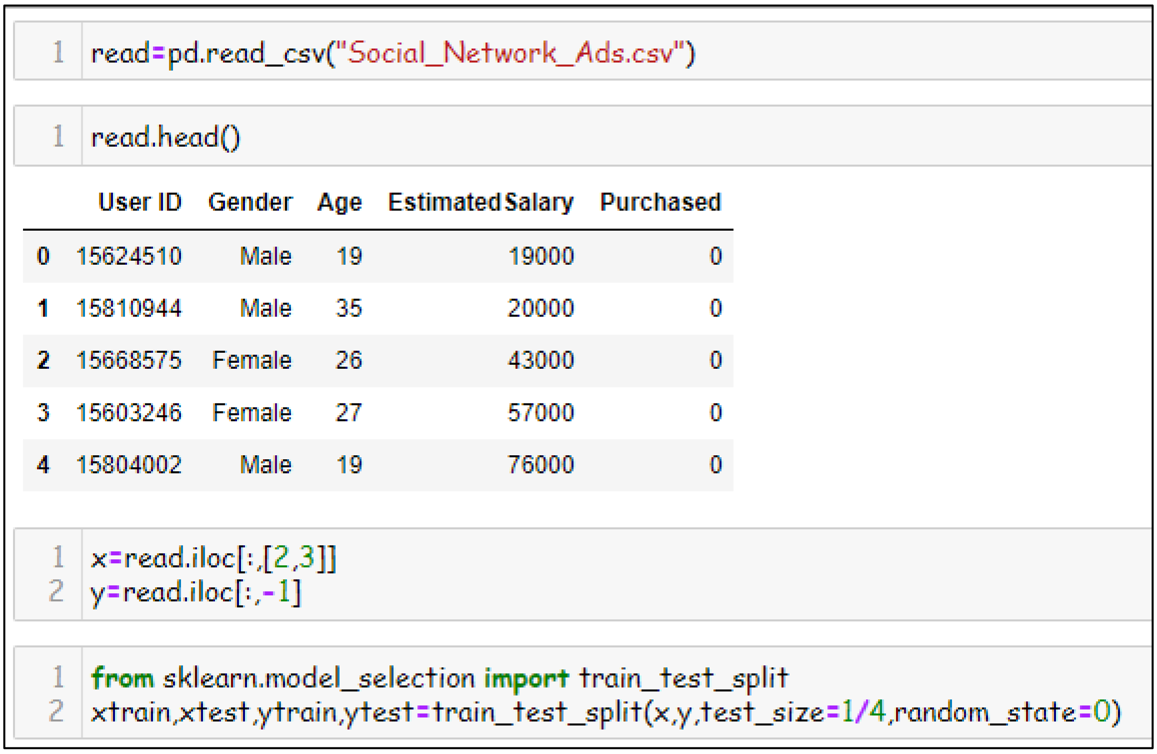
Python Program Implementation

- Importing necessary libraries



```python
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
```

- Reading dataset and selecting the features that are needed. Selecting Independent variable as x and dependent variable as y and dividing the dataset using train_test_split.



```python
read=pd.read_csv("Social_Network_Ads.csv")
```

```python
read.head()
```

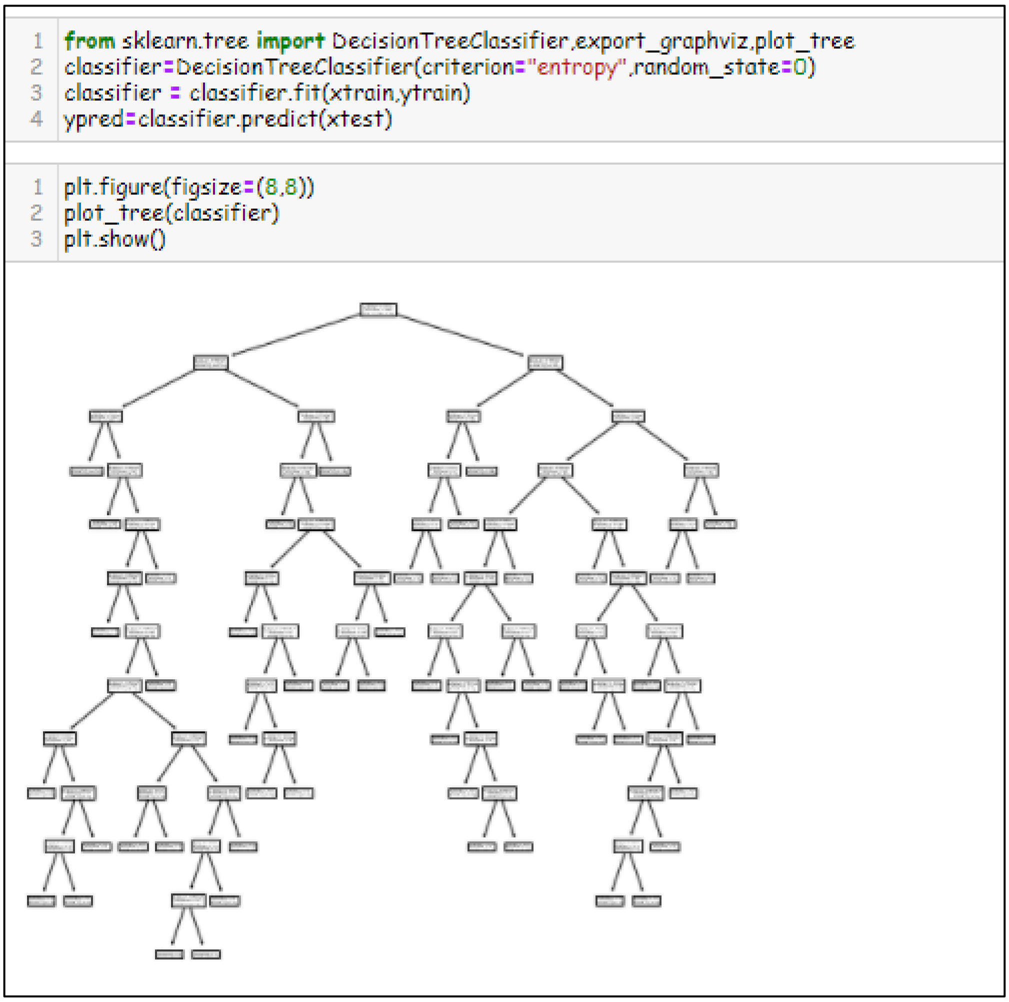
	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0

```python
x=read.iloc[:,[2,3]]
y=read.iloc[:,-1]
```

```python
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=1/4,random_state=0)
```

Firstly, implementing Decision Tree

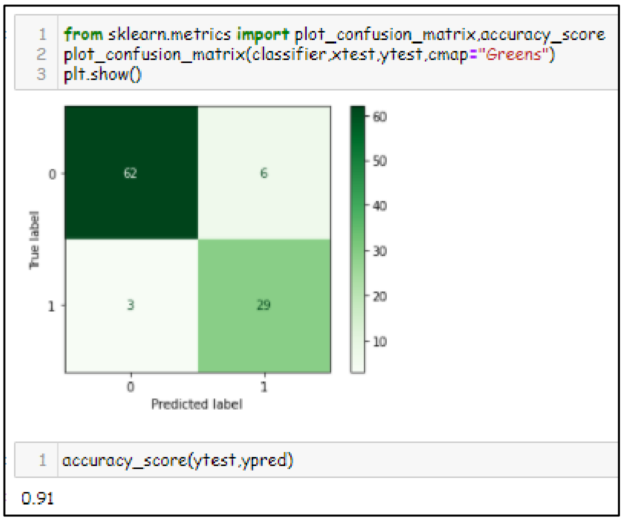
- Using Sklearn package to import decision tree and plotting it using plot_tree.



```
from sklearn.tree import DecisionTreeClassifier,export_graphviz,plot_tree
classifier=DecisionTreeClassifier(criterion="entropy",random_state=0)
classifier = classifier.fit(xtrain,ytrain)
ypred=classifier.predict(xtest)
```

```
plt.figure(figsize=(8,8))
plot_tree(classifier)
plt.show()
```

- Getting evaluation of the model using confusion matrix and accuracy score.



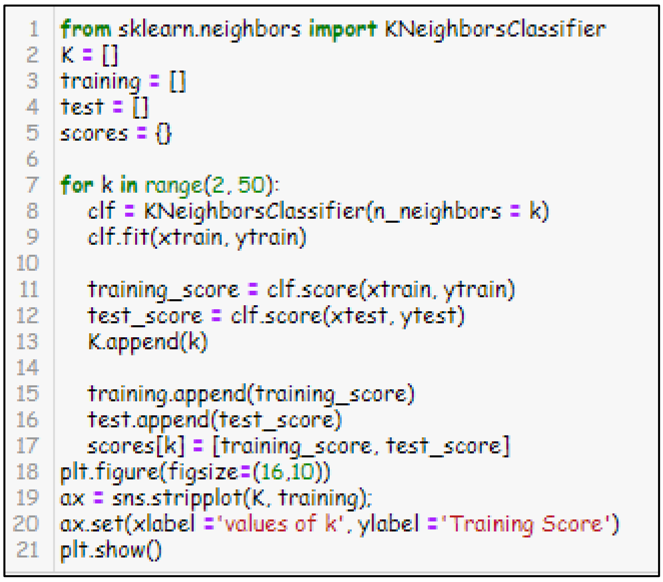
```
from sklearn.metrics import plot_confusion_matrix,accuracy_score
plot_confusion_matrix(classifier,xtest,ytest,cmap="Greens")
plt.show()
```

```
accuracy_score(ytest,ypred)
```

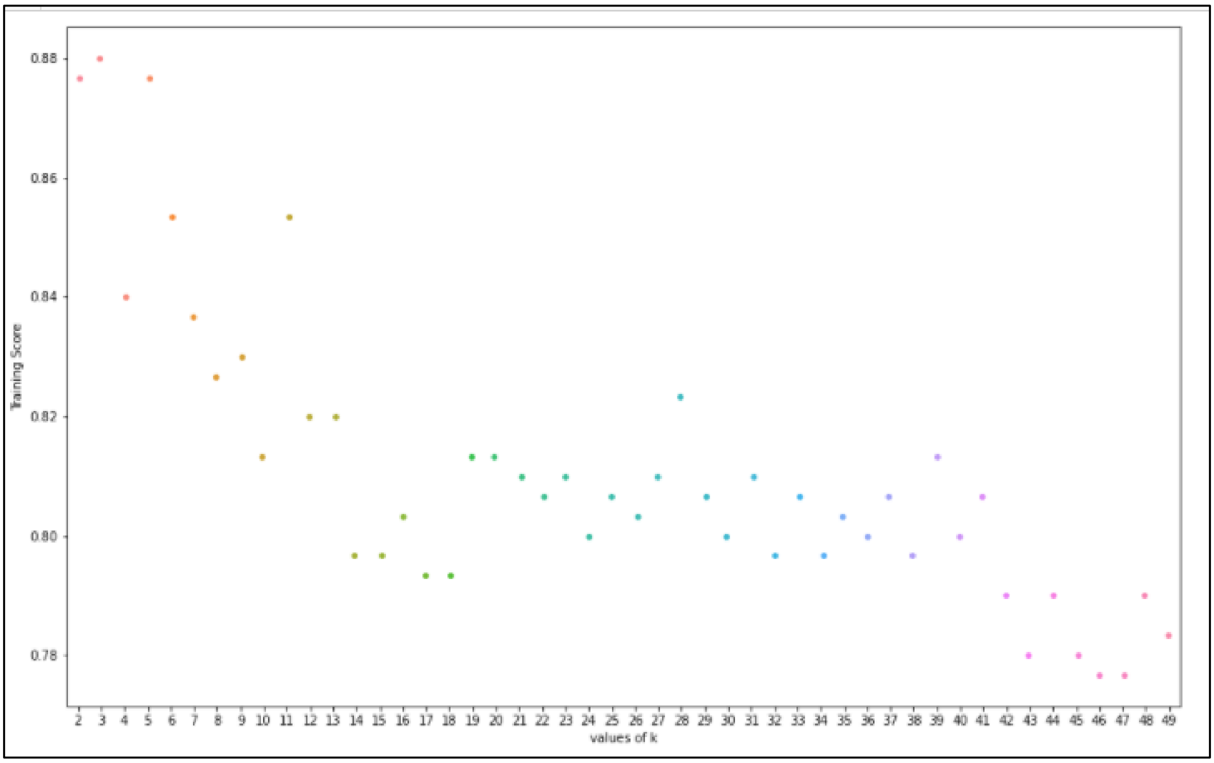
```
0.91
```

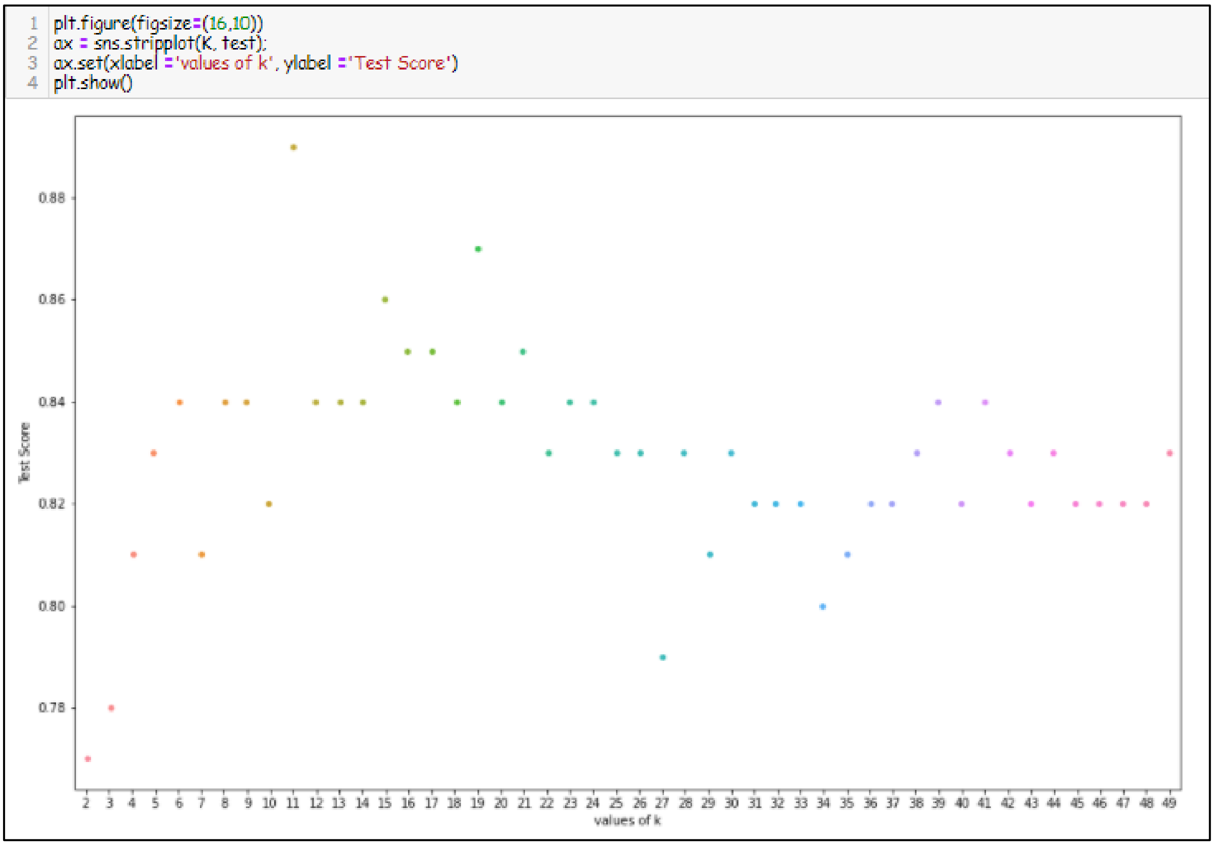
Implementing KNN

- Using sklearn package to import KNN classifier.



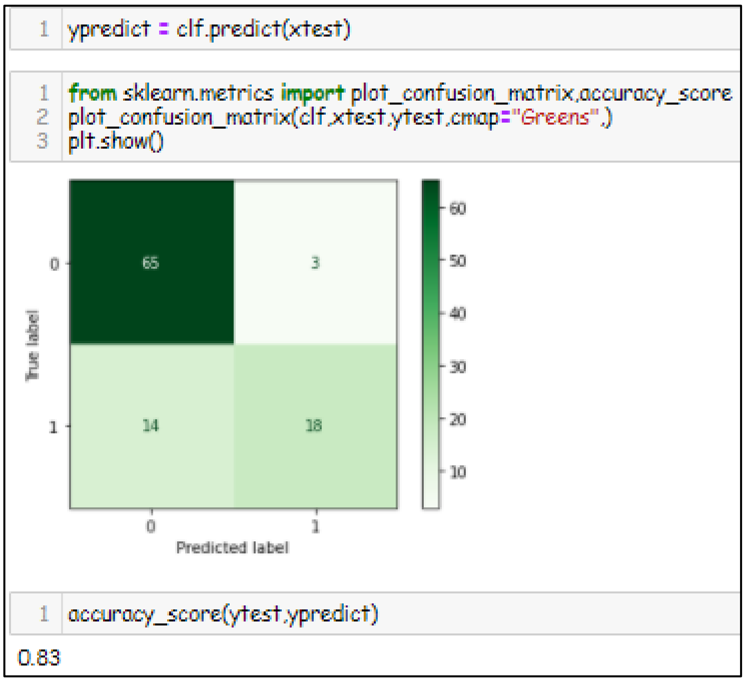
```python
from sklearn.neighbors import KNeighborsClassifier
K = []
training = []
test = []
scores = {}

for k in range(2, 50):
    clf = KNeighborsClassifier(n_neighbors = k)
    clf.fit(xtrain, ytrain)

    training_score = clf.score(xtrain, ytrain)
    test_score = clf.score(xtest, ytest)
    K.append(k)

    training.append(training_score)
    test.append(test_score)
    scores[k] = [training_score, test_score]
plt.figure(figsize=(16,10))
ax = sns.stripplot(K, training);
ax.set(xlabel ='values of k', ylabel ='Training Score')
plt.show()
```





```
plt.figure(figsize=(16,10))
ax = sns.stripplot(K, test);
ax.set(xlabel ='values of k', ylabel ='Test Score')
plt.show()
```

- Getting evaluation of the model using confusion matrix and accuracy score.



```
ypredict = clf.predict(xtest)
```

```
from sklearn.metrics import plot_confusion_matrix,accuracy_score
plot_confusion_matrix(clf,xtest,ytest,cmap="Greens",)
plt.show()
```

```
accuracy_score(ytest,ypredict)
```

```
0.83
```