```
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import load_wine
```

```
y = load_wine()
```

```
from sklearn.model_selection import train_test_split
```

```
df = pd.DataFrame(y.data , columns = y.feature_names)
```

- 각 column간의 상관계수 분석

```
df.corr()
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nc
alcohol	1.000000	0.094397	0.211545	-0.310235	0.270798	0.289101	0.236815	
malic_acid	0.094397	1.000000	0.164045	0.288500	-0.054575	-0.335167	-0.411007	
ash	0.211545	0.164045	1.000000	0.443367	0.286587	0.128980	0.115077	
alcalinity_of_ash	-0.310235	0.288500	0.443367	1.000000	-0.083333	-0.321113	-0.351370	
magnesium	0.270798	-0.054575	0.286587	-0.083333	1.000000	0.214401	0.195784	
total_phenols	0.289101	-0.335167	0.128980	-0.321113	0.214401	1.000000	0.864564	
flavanoids	0.236815	-0.411007	0.115077	-0.351370	0.195784	0.864564	1.000000	
nonflavanoid_phenols	-0.155929	0.292977	0.186230	0.361922	-0.256294	-0.449935	-0.537900	
proanthocyanins	0.136698	-0.220746	0.009652	-0.197327	0.236441	0.612413	0.652692	
color_intensity	0.546364	0.248985	0.258887	0.018732	0.199950	-0.055136	-0.172379	
hue	-0.071747	-0.561296	-0.074667	-0.273955	0.055398	0.433681	0.543479	
od280/od315_of_diluted_wines	0.072343	-0.368710	0.003911	-0.276769	0.066004	0.699949	0.787194	
proline	0.643720	-0.192011	0.223626	-0.440597	0.393351	0.498115	0.494193	

분석 결과 , [2,3,5,6,11]열의 상관계수가 가장 높은 것을 알 수 있으므로, 이 열만 가지고 KNN을 해보기

```
from sklearn.model_selection import train_test_split
```

```
X_train , X_test , y_train , y_test = train_test_split(y.data[:,[2,3,5,6,11]],y.target , test_size = 0.3 , random_state = 2)
```

```
from sklearn.neighbors import KNeighborsClassifier
```

```
for i in range(1,100,2):
  knn = KNeighborsClassifier(n_neighbors = i)
  knn.fit(X_train , y_train)
  score = knn.score(X_train , y_train)
  print(score)
```

```
1.0
0.9435483870967742
0.9032258064516129
0.8951612903225806
0.8870967741935484
0.8548387096774194
0.8790322580645161
0.8387096774193549
0.8548387096774194
0.8467741935483871
0.8467741935483871
0.8467741935483871
0.8387096774193549
0.8387096774193549
0.8306451612903226
0.8064516129032258
0.8225806451612904
0.8064516129032258
0.8145161290322581
```

```
0.7903225806451613
0.7983870967741935
0.782258064516129
0.7258064516129032
0.7580645161290323
0.75
0.7096774193548387
0.7016129032258065
0.6693548387096774
0.6290322580645161
0.6048387096774194
0.6209677419354839
0.6209677419354839
0.6209677419354839
0.6209677419354839
0.6129032258064516
0.5887096774193549
0.5806451612903226
0.532258064516129
0.5403225806451613
0.532258064516129
0.532258064516129
0.5483870967741935
0.5483870967741935
0.5483870967741935
0.5645161290322581
0.5483870967741935
0.5483870967741935
0.5161290322580645
0.43548387096774194
0.3951612903225806
```

```
kn = KNeighborsClassifier(n_neighbors = 3)
```

```
df
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proantho
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28	
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26	
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	0.30	
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	0.24	
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	0.39	
...	...	...	...	...	...	...	...	...	
173	13.71	5.65	2.45	20.5	95.0	1.68	0.61	0.52	
174	13.40	3.91	2.48	23.0	102.0	1.80	0.75	0.43	
175	13.27	4.28	2.26	20.0	120.0	1.59	0.69	0.43	
176	13.17	2.59	2.37	20.0	120.0	1.65	0.68	0.53	
177	14.13	4.10	2.74	24.5	96.0	2.05	0.76	0.56	

178 rows × 13 columns

```
x_new = np.array([[2.43 , 15 , 2.79 , 3.05 , 3.91]])
```

```
kn.fit(X_train , y_train)
```

```
▾        KNeighborsClassifier
KNeighborsClassifier(n_neighbors=3)
```

```
yhat = kn.predict(x_new)
```

```
y.target_names[yhat]
```

```
array(['class_0'], dtype='<U7')
```

```
kn.score(X_train , y_train)
```

```
0.9435483870967742
```

```
kn.score(X_test , y_test)
```

```
0.8148148148148148
```

- 앞서 했던 KNN에선 , 점수가 0.8을 넘기지 못했는데 , 상관관계가 높은 것들만 뽑아서 해보니 0.8보다 커졌다. 변수가 많을 땐 상관관계를 고려해서 하는것이 옳았다.

```
from sklearn.metrics import accuracy_score
```

```
yhat = kn.predict(X_test)
```

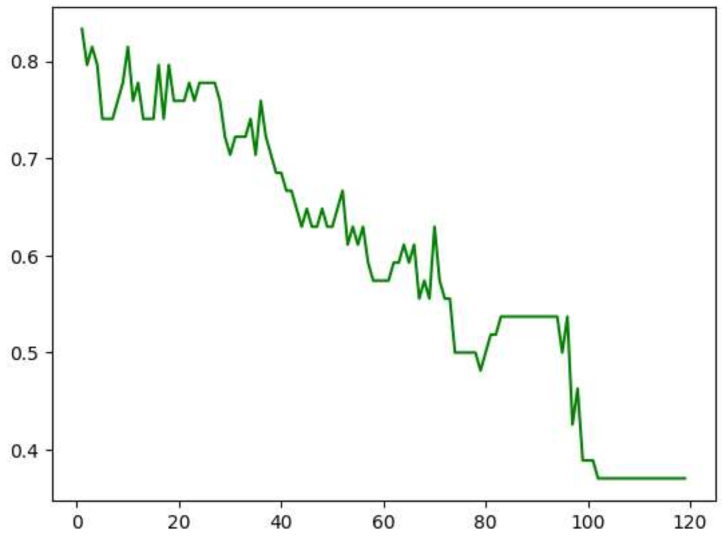
```
n = 120
acc = np.zeros([n-1])
for i in range(1,n):
  clf = KNeighborsClassifier(n_neighbors = i).fit(X_train , y_train)
  yhat = clf.predict(X_test)
  acc[i-1] = accuracy_score(y_test , yhat)

print(acc)
```

```
[0.83333333 0.7962963  0.81481481 0.7962963  0.74074074 0.74074074
 0.74074074 0.75925926 0.77777778 0.81481481 0.75925926 0.77777778
 0.74074074 0.74074074 0.74074074 0.7962963  0.74074074 0.7962963
 0.75925926 0.75925926 0.75925926 0.77777778 0.75925926 0.77777778
 0.77777778 0.77777778 0.77777778 0.75925926 0.72222222 0.7037037
 0.72222222 0.72222222 0.72222222 0.74074074 0.7037037  0.75925926
 0.72222222 0.7037037  0.68518519 0.68518519 0.66666667 0.66666667
 0.64814815 0.62962963 0.64814815 0.62962963 0.62962963 0.64814815
 0.62962963 0.62962963 0.64814815 0.66666667 0.61111111 0.62962963
 0.61111111 0.62962963 0.59259259 0.57407407 0.57407407 0.57407407
 0.57407407 0.59259259 0.59259259 0.61111111 0.59259259 0.61111111
 0.55555556 0.57407407 0.55555556 0.62962963 0.57407407 0.55555556
 0.55555556 0.5        0.5        0.5        0.5        0.5
 0.48148148 0.5        0.51851852 0.51851852 0.53703704 0.53703704
 0.53703704 0.53703704 0.53703704 0.53703704 0.53703704 0.53703704
 0.53703704 0.53703704 0.53703704 0.53703704 0.5        0.53703704
 0.42592593 0.46296296 0.38888889 0.38888889 0.38888889 0.37037037
 0.37037037 0.37037037 0.37037037 0.37037037 0.37037037 0.37037037
 0.37037037 0.37037037 0.37037037 0.37037037 0.37037037 0.37037037
 0.37037037 0.37037037 0.37037037 0.37037037 0.37037037]
```

```
acc.max()
```

```
0.8333333333333334
```

```
plt.plot(range(1,n) , acc , color = 'g')
plt.show()
```



- 이웃 수가 1일 땐 Train 점수가 1이기 때문에 제외 , 그 외 가장 높은 K = 3을 선택