In [104]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings("ignore")
import seaborn as sns
```

In [105]:
```
from sklearn.datasets import load_wine
```

In [106]:
```
data=load_wine()
```

In [107]:
```
df=pd.DataFrame(data.data,columns=data.feature_names)
```

In [108]:
```
df["target"]=data.target
```

In [109]:
```
df.head()
```

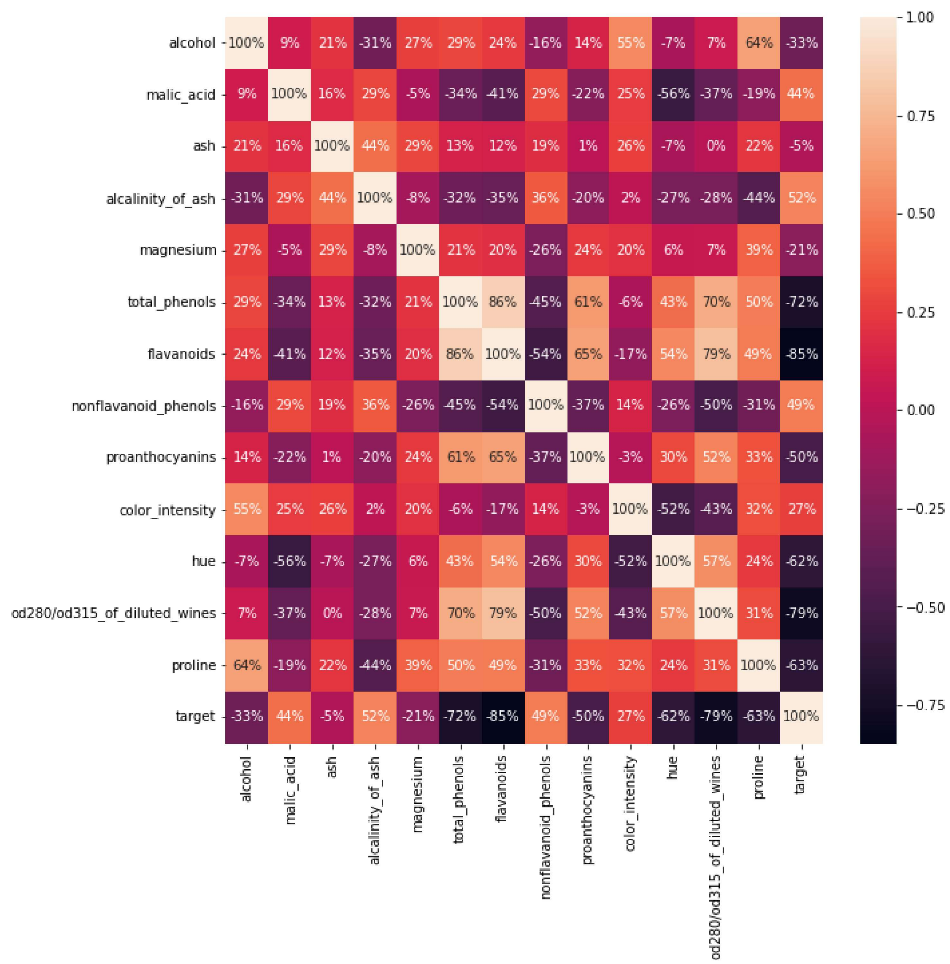
Out[109]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	od280/od315
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28	2.29	5.64	1.04	
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26	1.28	4.38	1.05	
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	0.30	2.81	5.68	1.03	
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	0.24	2.18	7.80	0.86	
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	0.39	1.82	4.32	1.04	

In [ ]:

In [112]:
```
plt.figure(figsize=(10,10))
sns.heatmap(df.corr(),annot=True,fmt=".0%")
```

Out[112]: <AxesSubplot:>



SCALING

In [56]:
```
X=df.drop(["target"],axis=1)
```

In [57]:
```
y=df["target"]
```

In [58]:
```
from sklearn.preprocessing import StandardScaler
```

In [59]:
```
sc=StandardScaler()
```

In [60]:
```
for i in X:
    X[i]=sc.fit_transform(X[[i]])
```

In [ ]:

MODEL BUILDING

In [61]:
```
from sklearn.decomposition import PCA
```

In [73]:
```
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from xgboost import XGBClassifier
models = []

models.append(("Logistic Regression:",LogisticRegression()))
models.append(("K-Nearest Neighbour:",KNeighborsClassifier(n_neighbors=3)))
models.append(("Support Vector Machine-linear:",SVC(kernel="linear")))
models.append(("Support Vector Machine-rbf:",SVC(kernel="rbf")))
models.append(("eXtreme Gradient Boost:",XGBClassifier()))
```

In [102]:
```
pca=PCA(n_components=7)
x_transformed=pca.fit_transform(X)
```

In [103]:
```
results = []
names = []
for name,model in models:
    kfold = KFold(n_splits=10)
    cv_result = cross_val_score(model,x_transformed,y.values.ravel(), cv = kfold,scoring = "accuracy")
    names.append(name)
    results.append(cv_result)
for i in range(len(names)):
    print(names[i],results[i].mean()*100)
```

```
Logistic Regression: 96.1111111111111
K-Nearest Neighbour: 92.77777777777779
Support Vector Machine-linear: 94.9673202614379
Support Vector Machine-rbf: 96.04575163398692
eXtreme Gradient Boost: 94.41176470588235
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: