DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

CHANDIGARH UNIVERSITY
UNIVERSITY INSTITUTE OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Submitted By: Lipakshi	**Submitted To:** Navneet Kaur
Subject Name	Machine Learning Lab
Subject Code	20CSP-317
Branch	Computer Science
Semester	5th

UNIVERSITY INSTITUTE OF ENGINEERING

Department of Computer Science & Engineering

CU
CHANDIGARH UNIVERSITY

DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Subject Name: Machine Learning Lab

Subject Code: 20CSP-317

Submitted to:

Faculty name: Navneet Kaur

Submitted by:

Name: Lipakshi

UID: 20BCS5082

Section: 607

Group: B

Ex. No	**List of Experiments**	**Date**	**Conduct (MM: 12)**	**Viva (MM: 10)**	**Record (MM: 8)**	**Total (MM: 30)**	**Remarks/Signature**
1.1	Implement Exploratory Data Analysis on any data set.						
1.2	Implement Data Visualization.	23-08-2022					
1.3	Data analysis of any data set via graphs using linear regression.						
1.4	Implement support Vector machine on any data set and analyse the accuracy with logistic regression.	10-10-2022					
2.2	Implement Naive Bayes on any Data Set.	10-10-2022					

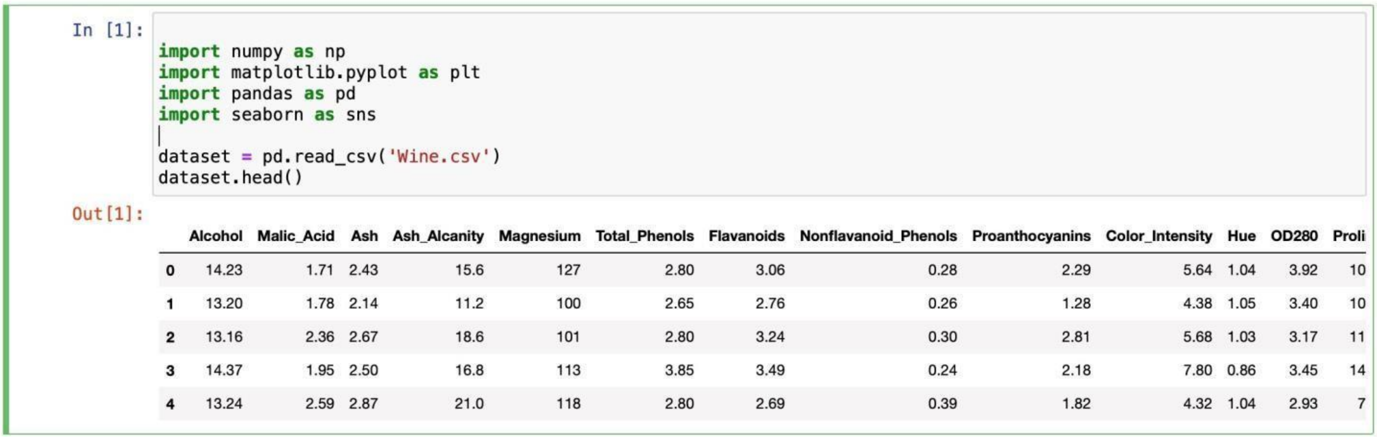
2.3							
2.4							
3.1	Implement K-Means	07-11-2022					
3.2	Implement PCA	07-11-2022					
3.3							

Experiment 9

Q1. Task to be done/ Which logistics used: Implement PCA.

# Code:

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

dataset = pd.read_csv('Wine.csv')
dataset.head()
```



In [1]:

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

dataset = pd.read_csv('Wine.csv')
dataset.head()
```

Out[1]:

	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflavanoid_Phenols	Proanthocyanins	Color_Intensity	Hue	OD280	Proli
0	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	5.64	1.04	3.92	10
1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05	3.40	10
2	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	5.68	1.03	3.17	11
3	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	7.80	0.86	3.45	14
4	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	4.32	1.04	2.93	7

```
X = dataset.iloc[:, 0:13].values y
= dataset.iloc[:, 13].values


from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)


from sklearn.preprocessing import StandardScaler sc
= StandardScaler()

X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

from sklearn.decomposition import PCA

pca = PCA(n_components = 2)

X_train = pca.fit_transform(X_train)
X_test = pca.transform(X_test)

explained_variance = pca.explained_variance_ratio_

from sklearn.linear_model import LogisticRegression
```

classifier = LogisticRegression(random_state = 0)

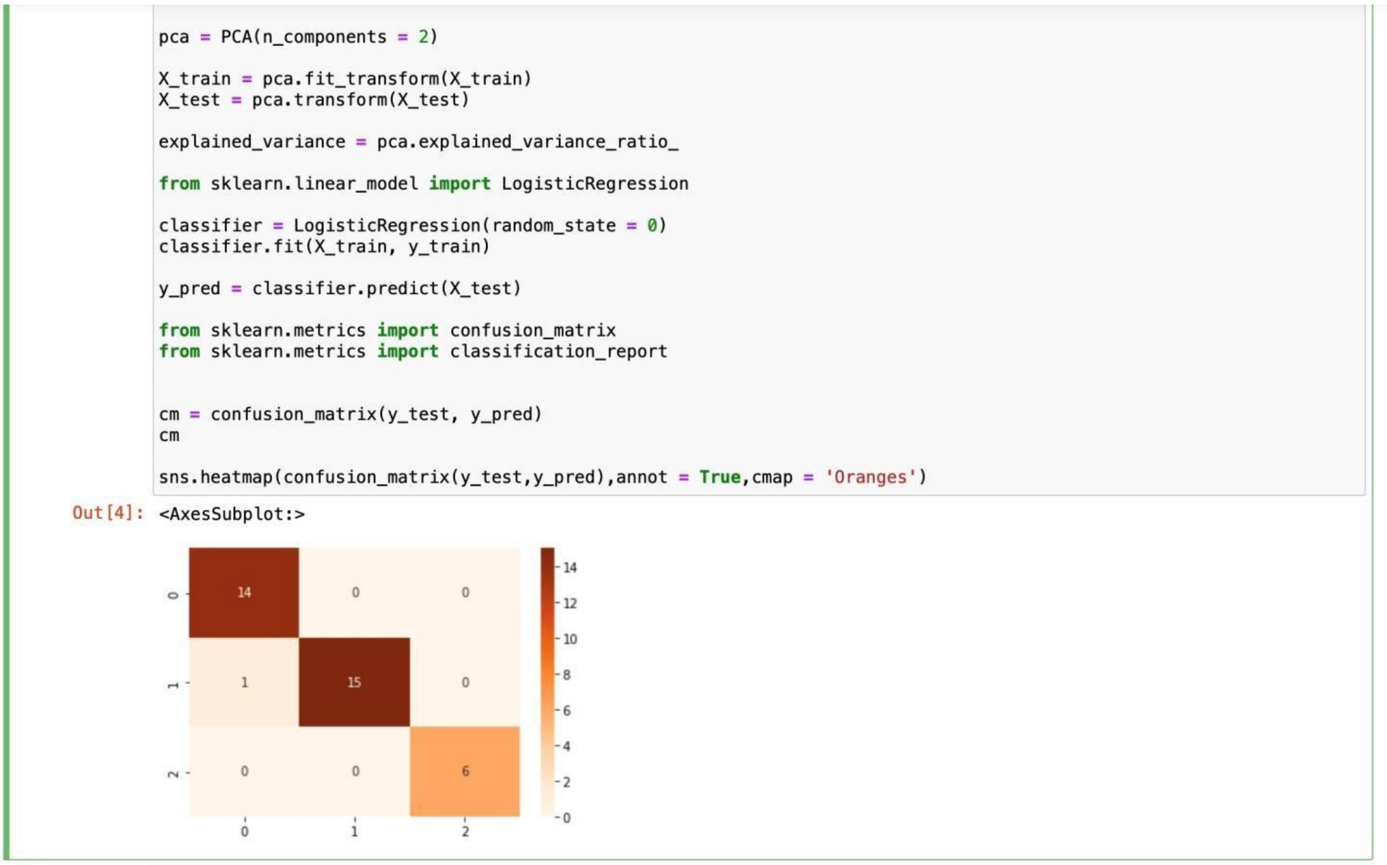
classifier.fit(X_train, y_train) y_pred =

classifier.predict(X_test)

from sklearn.metrics import confusion_matrix from
sklearn.metrics import classification_report

cm = confusion_matrix(y_test, y_pred) cm

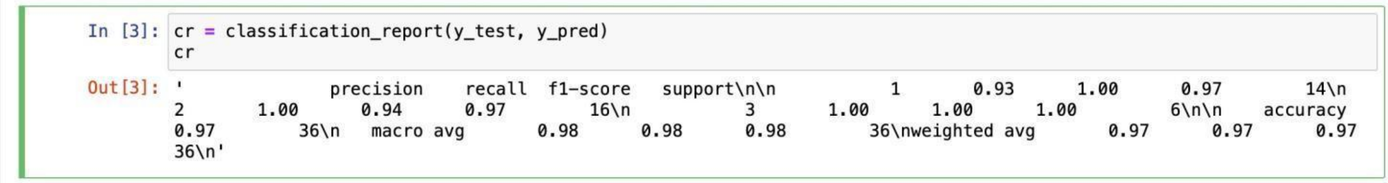
sns.heatmap(confusion_matrix(y_test,y_pred),annot = True,cmap = 'Oranges')



```
pca = PCA(n_components = 2)

X_train = pca.fit_transform(X_train)
X_test = pca.transform(X_test)

explained_variance = pca.explained_variance_ratio_

from sklearn.linear_model import LogisticRegression

classifier = LogisticRegression(random_state = 0)
classifier.fit(X_train, y_train)

y_pred = classifier.predict(X_test)

from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report


cm = confusion_matrix(y_test, y_pred)
cm

sns.heatmap(confusion_matrix(y_test,y_pred),annot = True,cmap = 'Oranges')
```

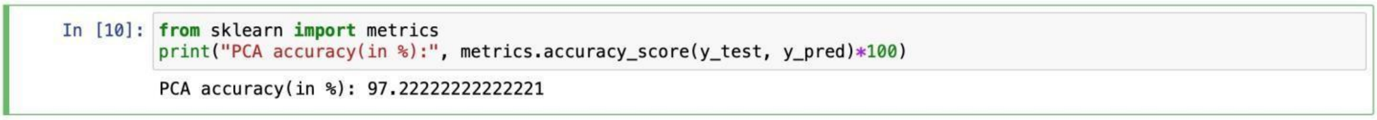
```
Out[4]: <AxesSubplot:>
```

cr = classification_report(y_test, y_pred) cr



```
In [3]: cr = classification_report(y_test, y_pred)
        cr
Out[3]: '              precision    recall  f1-score   support\n\n           1       0.93      1.00      0.97        14\n
        2       1.00      0.94      0.97        16\n           3       1.00      1.00      1.00         6\n\n    accuracy
        0.97        36\n   macro avg       0.98      0.98      0.98        36\nweighted avg       0.97      0.97      0.97
        36\n'
```

from sklearn import metrics
print("PCA accuracy(in %):", metrics.accuracy_score(y_test, y_pred)*100)



```
In [10]: from sklearn import metrics
         print("PCA accuracy(in %):", metrics.accuracy_score(y_test, y_pred)*100)

         PCA accuracy(in %): 97.22222222222221
```

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1			
2			
3			
4			