```
import pandas as pd                    PCA (7)
dataset = pd.read_csv(r"/Users/Machine Learning/Dataset/wine.csv")
 x = dataset. iloc[:,0,13]
 y = dataset. iloc[:13]
from sklearn.model selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.2, random_state=0)

from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
xtrain = sc.fit_transform(xtrain)
xtest = sc.transform(xtest)

 from sklearn.decomposition import PCA
pca = PCA(n_components=2)
xtrain = pca.fit_transform(xtrain)
xtest = pca.transform(xtest)

 from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model. fit(xtrain, ytrain)
ypred = model.predict(xtest)

from sklearn.metrics import confusion_matrix, classification_report
print("Confusion matrix: \n", confusion_matrix(ytest, ypred))
print("Classifiaction report \n", classification_report(ytest, ypred))
```

```
import seaborn as sns          Text Analysis(6)
import matplotlib.pyplot as plt
from sklearn.datasets import fetch_2@newsgroups
from sklearn.pipeline import make_pipeline
from sklearn.naive_bayes import MultinomialNB
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import confusion_matrix, accuracy_score

data = fetch_20newsgroups()
model = make_pipeline(TfidfVectorizer(), MultinomialNB())

for i in range(0, len(data.target_names), 5):
cat = data.target_names [i:i+5]
train = fetch_20newsgroups(subset="'train', categories=cat)
test = fetch_20newsgroups(subset='test', categories=cat)
model. fit(train.data, train.target)
label = model.predict(test.data)
print(f"\nAC: {i} to {i+5} ", accuracy_score(test.target, label))
cm = confusion_matrix(test.target, label)
sns.heatmap(cm.T, square=True, annot = True, fmt = 'd',
 xticklabels=train.target_names, yticklabels=train.target
plt.title(f"sample no. {i}")
plt.xlabel("True Label")
plt.ylabel( ‘Predicted Label’)
plt.show()
```

```
import pandas as pd           LDA(8)
dataset = pd.read_csv(r"/Users//Dataset/wine.csv")
x = dataset. iloc[:,0:13]
y = dataset. iloc[:,13]

from sklearn.model selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.2, random_state=0)

from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
xtrain = sc.fit_transform(xtrain)
xtest = sc.transform(xtest)

from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
lda = LinearDiscriminantAnalysis(n_components=2)
xtrain = lda.fit_transform(xtrain, ytrain)
xtest = lda.transform(xtest)

from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(max_depth=2, random_state=0)
model. fit(xtrain, ytrain)
ypred = model.predict(xtest)

from sklearn.metrics import confusion_matrix, classification_report
print("Confusion matrix: \n", confusion_matrix(ytest, ypred))
print("Classifiaction report \n", classification_report(ytest, ypred))
```

```
import pandas as pd      MultiLayered Perceptron(9)
dataset = pd.read_csv(r"/Users/shreygupta/Dataset/wine.csv")

x = dataset.iloc[:,0:13]
y = dataset.iloc[:,13]

from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x,y,test_size=0.2, random_state=0)

from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
xtrain = sc.fit_transform(xtrain)
xtest = sc.transform(xtest)

from sklearn.neural_network import MLPClassifier
model = MLPClassifier(hidden_layer_sizes=(10,3,10), max_iter=10000)
model. fit(xtrain, ytrain)

ypred = model.predict(xtest)

from sklearn.metrics import accuracy_score, confusion_matrix
print("Confusion: \n", confusion_matrix(ytest, ypred))
print(“Accuracy: ", accuracy_score(ytest, ypred))
```