Name: Shalin Patel

Id: 38

Ans 1:

Difference between LDA and LR

**LDA** works by Least squares estimation; where as LR works on maximum likelihood

Matrices should be identical in population within-group covariance in LDA

For **LDA** similar nn required for groups, not for LR

LDA is sensitive to outliers where LR is not

LDA is younger then LR

LDA is more efficient then LR

Code:

from sklearn import datasets  
from sklearn.discriminant\_analysis import LinearDiscriminantAnalysis as LDA  
iris = datasets.load\_iris()  
xVal = iris.data[:, :2]  
print("--------------X Value--------------------------- \n",xVal)  
yVal = iris.target  
print("\n ----------------Target Value--------------------- \n",yVal)  
myClassifier = LDA()  
myClassifier.fit(xVal, yVal)  
LDA(n\_components=None, priors=None, shrinkage=None, solver='svd',  
 store\_covariance=False, tol=0.0001)  
print("\n-----Prediction------------------------\n")  
print(myClassifier.predict([[-0.6, -2]]))

**Explanation:**

Used Iris dataset

Taken first two arguments of Iris and target as Y

Calculated LDA

Output:A screenshot of a computer

Description generated with very high confidenceA screenshot of a computer

Description generated with very high confidence

Ans 2:

import numpy as np  
from sklearn import datasets  
from sklearn import svm  
from sklearn.model\_selection import train\_test\_split  
from sklearn import metrics  
  
irisData = datasets.load\_iris()  
x = irisData.data[:,:2]  
y = irisData.target  
  
c=0.2  
minX, maxX = x[:, 0].min() - 1, x[:, 0].max() + 1  
minY, maxY = x[:, 1].min() - 1, x[:, 1].max() + 1  
x1, y1 = np.meshgrid(np.arange(minX, maxX, c),  
 np.arange(minY, maxY, c))  
  
x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.15)  
model = svm.SVC()  
  
predictions\_linear = model.set\_params(kernel='linear').fit(x\_train, y\_train).predict(x\_test)  
  
predictions\_rbf = model.set\_params(kernel='rbf').fit(x\_train, y\_train).predict(x\_test)  
  
accuracy\_linear = metrics.accuracy\_score(y\_test,predictions\_linear)  
accuracy\_rbf = metrics.accuracy\_score(y\_test,predictions\_rbf)  
  
print("Accuracy with kernel=linear:",accuracy\_linear)

print("Accuracy with kernel=rbf:",accuracy\_rbf)

**Output:**

A screenshot of a computer screen

Description generated with very high confidence

Ans 3:

Code:

import nltk  
from nltk import word\_tokenize  
from nltk.util import ngrams  
from collections import Counter  
  
import re  
import urllib.request  
  
  
with open("input.txt",'r') as f:  
 content = f.read()  
  
#Tokenization  
  
from nltk.stem import WordNetLemmatizer  
print("\n\n")  
print("LEMMATIZATION")  
lemmatizer=WordNetLemmatizer()  
print(lemmatizer.lemmatize(content))  
  
  
frequencies = Counter([])  
token = word\_tokenize(content)  
bigrams = ngrams(token, 2)  
frequencies += Counter(bigrams)  
  
topFiveBG=list()  
for i in range (0,5) :  
 topFiveBG.append(" ".join(re.findall("[a-zA-Z]+",str(frequencies).split(":")[i])))  
print(frequencies)  
  
print("top five words")  
print(topFiveBG)  
  
lines={}  
for line in content.split("."):  
 for word in topFiveBG:  
 if word in line:  
 if line in lines:  
 pass  
 else:  
 lines[line]=""  
  
result=list()  
for line in lines:  
 result.append(line+".")  
  
print("Final Result")  
print(result)

Output:

A screenshot of a computer screen

Description generated with very high confidenceA screenshot of a computer screen

Description generated with very high confidence

Ans 4:

The value of K can affect in following aspects:

In KNN, K is the parameter that can be used to get best fit which controls the shpe of the boundry decisions

For small K result in restraining the region of a given prediction and forcing our classifier to be “more blind” to the overall distribution. A small value for K provides the most flexible fit, which will have low bias but high variance. Our boundary of decision will be more jagged graphically.

If we choose high K, it results in resilient to outliers which provides increased bias but smoother decision boundaries and lower variance