



ARTIFICIAL INTELLIGENCE LAB (CSL5402)

Name: Lakhan Kumawat

Roll: 1906055

Program: B.Tech CSE
(5th Sem JUL-DEC 2021)

Assignment - 9

1. Read iris dataset (shared with the named "iris" in the file section inside the dataset folder) using K-Means calculate the accuracy and F1 score.

```
#importing libraries
import numpy as np
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.metrics import f1_score, accuracy_score

df = pd.read_csv('iris.csv')
X = df.iloc[:, :-1].values
y = df.iloc[:, -1].values
kmeans = KMeans(n_clusters = 3, init = 'k-means++')
y_pred = kmeans.fit_predict(X)

print('accuracy score: ', accuracy_score(y_pred, y))
print('f1 score(weighted):', f1_score(y, y_pred, average =
'weighted'))
print('f1 score(micro):', f1_score(y, y_pred, average =
'micro'))
print('f1 score(macro):', f1_score(y, y_pred, average =
'macro'))
print('f1 score(None):', f1_score(y, y_pred, average = None))
```

OUTPUT:

```
accuracy score: 0.8933333333333333
f1 score(weighted): 0.8917748917748918
f1 score(micro): 0.8933333333333333
f1 score(macro): 0.8917748917748917
f1 score(None): [1.          0.85714286 0.81818182]
```

2. Implement spam and non-spam classification task (shared with the named "SMSspamcollection" in the file section inside the dataset folder) using Naïve Bayes calculate the accuracy.

```
#importing important libraries
import numpy as np
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix, accuracy_score

[nltk_data] Downloading package stopwords to C:\Users\LAKHN
[nltk_data] KUMAWAT\AppData\Roaming\nltk_data...
[nltk_data] Package stopwords is already up-to-date!

df = pd.read_csv('spam.csv', encoding='latin-1')
df.head()
```

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN

```
corpus = []
for i in range(0, len(df)):
    review = re.sub('[^a-zA-Z]', ' ', df['v2'][i])
    review = review.lower()
    review = review.split()
    ps = PorterStemmer()
    all_stopwords = stopwords.words('english')
    all_stopwords.remove('not')
    review = [ps.stem(word) for word in review if not word in
set(all_stopwords)]
    review = ' '.join(review)
    corpus.append(review)
cv = CountVectorizer(max_features = 1500)
X = cv.fit_transform(corpus).toarray()
y = df.iloc[:, 0].values
le = LabelEncoder()
y = le.fit_transform(y)

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size = 0.2)
classifier = GaussianNB()
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
cm = confusion_matrix(y_test, y_pred)
print(cm)
print("Accuracy score:", accuracy_score(y_test, y_pred))
```

OUTPUT:

```
[[753 194]
 [ 11 157]]
Accuracy score: 0.8161434977578476
```

3. Read iris dataset (shared with the named "iris" in the file section inside the dataset folder) using Logistic Regression calculate the accuracy and the classification score (precision recall f1-score).

```
#importing libraries
```

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import
accuracy_score, classification_report
from sklearn.preprocessing import StandardScaler
```

```
df = pd.read_csv('iris.csv')
df.head()
```

	sepal_length	sepal_width	petal_length	petal_width	plant
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

```
#checking null values
```

```
df.isnull().sum()
```

```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
plant           0
dtype: int64
```

```
x = df.iloc[:, :-1].values
y = df.iloc[:, -1].values
X_train, X_test, y_train, y_test = train_test_split(x,
y, stratify=df['plant'], test_size =0.2)
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
classifier = LogisticRegression()
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)

print("Accuracy score: ", accuracy_score(y_test, y_pred))
print('classification report:\n')
print(classification_report(y_test, y_pred))
```

OUTPUT:

Accuracy score: 0.9666666666666667

classification report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	10
1	0.91	1.00	0.95	10
2	1.00	0.90	0.95	10
accuracy			0.97	30
macro avg	0.97	0.97	0.97	30
weighted avg	0.97	0.97	0.97	30

End Of Assignment