Date Page

4) Logistic Regression : classify emails as spam or not using the spam dataset.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from sklearn.preprocessing import StandardScaler

df = pd.read_csv('spam.csv')
print("Dataset preview:")
print(df.head())
x = df.drop('spam', axis=1)
y = df['spam']
x_train, x_test, y_train = train_test_split(X, y, test_size=0.3, random_state=42)
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
model = LogisticRegression()
model.fit(X_train_scaled, y_train)
y_pred = model.predict(X_test_scaled)
print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred))
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
print("\nAccuracy Score:")
print(accuracy_score(y_test, y_pred))
```

Output : Dataset preview :-

	word_freq_make	word_freq_address	word_freq_all	
0	0.00	0.64	0.64	0.0
1	0.21	0.28	0.50	0.0
2	0.06	0.00	0.71	0.0
3	0.00	0.00	0.00	0.0
4	0.00	0.00	0.00	0.0

	word_freq_3d	word_freq_our	word_freq_over	char_freq_;	char_freq
0	0.00	0.00	0.00	0.000	
1	0.00	0.94	0.00	0.132	
2	0.64	0.25	0.01	0.143	
3	0.31	0.63	0.00	0.137	
4	0.31	0.63	0.00	0.135	

	word_freq_addresses	word_freq_free	word_freq_business	word_freq_email
0	0.32	0.00	0.00	0.00
1	0.14	0.28	0.21	0.07
2	1.23	0.19	0.19	0.12

	capital_run_length_average	capital_run_length_longest	capital_run_length_total	spam
0	3.756	61	278	1
1	5.114	101	1028	1
2	9.821	485	2259	1
3	3.537	40	191	1
4	3.537	40	191	1

Confusion Matrix :

[[769 35]
[71 506]]

Classification Report :

	precision	recall	f1-score	support
0	0.92	0.96	0.94	804
1	0.94	0.88	0.91	577
accuracy			0.92	1381
macro avg	0.93	0.92	0.92	1381
weighted avg	0.92	0.92	0.92	1381

Accuracy Score :
0.9232442060680667