# Twitter Sentiment Analysis - Logistic Regression

## Python Code

import pandas as pd   
from sklearn.model\_selection import train\_test\_split  
from sklearn.feature\_extraction.text import TfidfVectorizer  
from sklearn.linear\_model import LogisticRegression  
from sklearn.metrics import accuracy\_score,classification\_report  
  
# Read dataset  
df = pd.read\_csv('twitter\_sentiment.csv',encoding='latin1')  
print(df)  
  
# Check null values  
print(df.isnull().sum())  
  
# Convert categorical to numerical  
df['sentiment'] = df['sentiment'].replace({'positive':1, 'negative':0})  
print(df)  
  
# Feature separation  
x = df['text']  
y = df['sentiment']  
print(x)  
print(y)  
  
# Train-test split  
x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.25, random\_state=42)  
print("x training data:", x\_train)  
print("x testing data", x\_test)  
print("y training data", y\_train)  
print("y testing data", y\_test)  
  
# Tf-idf vectorizer  
tfv = TfidfVectorizer()  
x\_train\_tfidf = tfv.fit\_transform(x\_train)  
x\_test\_tfidf = tfv.transform(x\_test)  
print("training data", x\_train\_tfidf)  
print("testing data", x\_test\_tfidf)  
  
# Logistic Regression  
logr = LogisticRegression()  
logr.fit(x\_train\_tfidf, y\_train)  
  
# Prediction  
y\_predict = logr.predict(x\_test\_tfidf)  
print(y\_predict)  
  
# Accuracy score  
accuracy = accuracy\_score(y\_test, y\_predict)  
print("Accuracy score:", accuracy)  
print("Classification Report:", classification\_report(y\_test, y\_predict))