

ASSIGNMENT 03

NAME: BHOSALE SRUSHTI AMAR

CLASS: TY AIDS-A

ROLL NO.: 23107008

```
In [50]: import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.svm import LinearSVC
```

```
In [8]: df=pd.read_csv("Musical_instruments_reviews.csv")
```

```
In [9]: df
```

Out[9]:

	reviewerID	asin	reviewerName	helpful	reviewText	overall	
0	A2IBPI20UZIROU	1384719342	cassandra tu "Yeah, well, that's just like, u...	[0, 0]	Not much to write about here, but it does exac...	5.0	
1	A14VAT5EAX3D9S	1384719342	Jake	[13, 14]	The product does exactly as it should and is q...	5.0	
2	A195EZSQDW3E21	1384719342	Rick Bennette "Rick Bennette"	[1, 1]	The primary job of this device is to block the...	5.0	It
3	A2C00NNG1ZQQG2	1384719342	RustyBill "Sunday Rocker"	[0, 0]	Nice windscreen protects my MXL mic and preven...	5.0	WIN
4	A94QU4C90B1AX	1384719342	SEAN MASLANKA	[0, 0]	This pop filter is great. It looks and perform...	5.0	pc
...
10256	A14B2YH83ZXMPP	B00JBIVXGC	Lonnie M. Adams	[0, 0]	Great, just as expected. Thank to all.	5.0	
10257	A1RPTVW5VEOSI	B00JBIVXGC	Michael J. Edelman	[0, 0]	I've been thinking about trying the Nanoweb st...	5.0	Lon
10258	AWCJ12KBO5VII	B00JBIVXGC	Michael L. Knapp	[0, 0]	I have tried coated strings in the past (incl...	4.0	
10259	A2Z7S8B5U4PAKJ	B00JBIVXGC	Rick Langdon "Scriptor"	[0, 0]	Well, MADE by Elixir and DEVELOPED with Taylor...	4.0	Ta
10260	A2WA8TDCTGUADI	B00JBIVXGC	TheTerrorBeyond	[0, 0]	These strings are really quite good, but I wou...	4.0	The q b

10261 rows × 9 columns

In [10]: df = df[['reviewText', 'overall']]

In [11]: df = df.dropna()

In [39]: df = df[df['overall'] != 3]

```
df['sentiment'] = df['overall'].apply(lambda x: 1 if x >= 4 else 0)
```

```
In [40]: X = df['reviewText']
y = df['sentiment']
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

```
In [41]: vectorizer = TfidfVectorizer(stop_words='english')
X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)
```

LOGISTIC REGRESSION

```
In [42]: lr_model = LogisticRegression()
lr_model.fit(X_train_vec, y_train)
```

```
Out[42]: ▾ LogisticRegression ⓘ ?
```

```
LogisticRegression()
```

```
In [43]: lr_pred = lr_model.predict(X_test_vec)
```

```
In [45]: print("LOGISTIC REGRESSION RESULTS")
print("Accuracy:", accuracy_score(y_test, lr_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, lr_pred))
print("Classification Report:\n", classification_report(y_test, lr_pred))
```

LOGISTIC REGRESSION RESULTS

Accuracy: 0.9488666315234581

Confusion Matrix:

```
[[ 0  97]
 [ 0 1800]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	97
1	0.95	1.00	0.97	1800
accuracy			0.95	1897
macro avg	0.47	0.50	0.49	1897
weighted avg	0.90	0.95	0.92	1897

```
/home/admin1/anaconda3/lib/python3.12/site-packages/sklearn/metrics/_classification.p
y:1531: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in label
s with no predicted samples. Use `zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/home/admin1/anaconda3/lib/python3.12/site-packages/sklearn/metrics/_classification.p
y:1531: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in label
s with no predicted samples. Use `zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/home/admin1/anaconda3/lib/python3.12/site-packages/sklearn/metrics/_classification.p
y:1531: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in label
s with no predicted samples. Use `zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

NAIVE BAYES

```
In [46]: nb_model = MultinomialNB()
nb_model.fit(X_train_vec, y_train)
```

```
Out[46]: ▾ MultinomialNB ⓘ ?
```

```
MultinomialNB()
```

```
In [47]: nb_pred = nb_model.predict(X_test_vec)
```

```
In [32]: nb_pred = nb_model.predict(X_test)
nb_acc = accuracy_score(y_test, nb_pred)
nb_report = classification_report(y_test, nb_pred)
nb_matrix = confusion_matrix(y_test, nb_pred)
```

```
/home/admin1/anaconda3/lib/python3.12/site-packages/sklearn/metrics/_classification.p
y:1531: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in label
s with no predicted samples. Use `zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/home/admin1/anaconda3/lib/python3.12/site-packages/sklearn/metrics/_classification.p
y:1531: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in label
s with no predicted samples. Use `zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/home/admin1/anaconda3/lib/python3.12/site-packages/sklearn/metrics/_classification.p
y:1531: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in label
s with no predicted samples. Use `zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

```
In [48]: print("NAIVE BAYES RESULTS")
print("Accuracy:", accuracy_score(y_test, nb_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, nb_pred))
print("Classification Report:\n", classification_report(y_test, nb_pred))
```

```
NAIVE BAYES RESULTS
Accuracy: 0.9488666315234581
Confusion Matrix:
[[ 0  97]
 [ 0 1800]]
Classification Report:
      precision    recall  f1-score   support
          0       0.00     0.00     0.00      97
          1       0.95     1.00     0.97    1800

      accuracy                           0.95      1897
     macro avg       0.47     0.50     0.49      1897
weighted avg       0.90     0.95     0.92      1897
```

```
/home/admin1/anaconda3/lib/python3.12/site-packages/sklearn/metrics/_classification.p
y:1531: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in label
s with no predicted samples. Use `zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/home/admin1/anaconda3/lib/python3.12/site-packages/sklearn/metrics/_classification.p
y:1531: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in label
s with no predicted samples. Use `zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/home/admin1/anaconda3/lib/python3.12/site-packages/sklearn/metrics/_classification.p
y:1531: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in label
s with no predicted samples. Use `zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

SVM

```
In [53]: svm = LinearSVC()
```

```
In [54]: svm.fit(X_train_vec, y_train)
svm_pred = svm.predict(X_test_vec)
```

```
In [55]: print("SVM")
print("Accuracy:", accuracy_score(y_test, svm_pred))
print(confusion_matrix(y_test, svm_pred))
print(classification_report(y_test, svm_pred))
```

```
SVM
Accuracy: 0.9520295202952029
[[ 10   87]
 [  4 1796]]
      precision    recall  f1-score   support
          0       0.71     0.10     0.18      97
          1       0.95     1.00     0.98    1800
accuracy                          0.95    1897
macro avg                      0.83     0.55     0.58    1897
weighted avg                   0.94     0.95     0.93    1897
```

EXAMPLE REVIEW

```
In [18]: review = ["This product is very bad and useless"]
review_vec = vectorizer.transform(review)
result = model.predict(review_vec)
print("Positive" if result[0] == 1 else "Negative")
```

Positive

```
In [56]: print("Model Comparison")
print("Logistic Regression Accuracy:", accuracy_score(y_test, lr_pred))
print("Naive Bayes Accuracy:", accuracy_score(y_test, nb_pred))
print("SVM Accuracy:", accuracy_score(y_test, svm_pred))
```

```
Model Comparison
Logistic Regression Accuracy: 0.9488666315234581
Naive Bayes Accuracy: 0.9488666315234581
SVM Accuracy: 0.9520295202952029
```

In []: