

```
from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

```
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score
```

```
df = pd.read_excel('/content/drive/MyDrive/AI_Assignment/tamil_sentences.xlsx')
```

```
df.head()
```

	Ungrammatical Statement	Standard Tamil
0	அவளே குரலுடன் பாடினான்.	அவன் குரலுடன் பாடினான்.
1	அவன் விளையாட்டில் சிறந்தான்.	அவன் விளையாட்டில் சிறந்தான்.
2	அவர்கள் கதையை கேட்டான்.	அவர்கள் கதையை கேட்டார்கள்.
3	அவள் மொபைலை பிடித்தான்.	அவள் மொபைலை பிடித்தாள்.
4	அவளை பார்த்தான் அவன் நண்பர்கள்.	அவளை அவன் நண்பர்கள் பார்த்தார்கள்.

```
# Check for missing values and drop them
print("Missing values in each column:")
print(df.isnull().sum())
df = df.dropna()
```

Missing values in each column:

Ungrammatical Statement	702
Standard Tamil	702

dtype: int64

```
# Split the data into features and target
X = df['Ungrammatical Statement'] # Ungrammatical Tamil statements
y = df['Standard Tamil'] # Corrected Tamil statements
```

```
# Vectorize the input text using character n-grams
vectorizer = CountVectorizer(ngram_range=(1, 2), analyzer='char', token_pattern=None)
X_vectorized = vectorizer.fit_transform(X)
```

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_vectorized, y, test_size=0.2, random_state=42)
```

```
# Initialize and train the model
model = MultinomialNB()
model.fit(X_train, y_train)
```

▼ MultinomialNB ⓘ ?

MultinomialNB()

```
# Make predictions on the test set
y_pred = model.predict(X_test)
```

```
# Evaluate the model's accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"Model Accuracy: {accuracy * 100:.2f}%")
```

Model Accuracy: 43.04%

```
from sklearn.metrics import classification_report
```

```
# Generate the classification report
report = classification_report(y_test, y_pred)
print("Classification Report:")
print(report)
```



அவள் கதையை கூறினாள்.	0.00	0.00	0.00	1
அவள் சாப்பிடுகிறாள்.	0.00	0.00	0.00	1
அவள் சைக்கிள் ஓட்டினாள்.	0.00	0.00	0.00	2
அவள் நடனம் ஆடினாள்.	0.00	0.00	0.00	3
அவள் நாளை பாடம் படிப்பாள்.	0.00	0.00	0.00	1
அவள் பரிசுகளை பெற்றாள்.	0.00	0.00	0.00	1
அவள் பல் தேய்ப்பாள்.	0.00	0.00	0.00	1
அவள் பள்ளிக்குச் சென்றாள்.	0.00	0.00	0.00	0
அவள் பாடம் படிக்கிறாள்.	0.40	1.00	0.57	2
அவள் பாடம் படித்தாள்.	0.20	1.00	0.33	1
அவள் புத்தகம் படித்தாள்.	0.00	0.00	0.00	1
அவள் மொபைலை பிடித்தாள்.	0.00	0.00	0.00	1
அவள் வேலை செய்கிறாள்.	0.00	0.00	0.00	1
அவள் வேலை செய்தாள்.	0.20	1.00	0.33	1
நான் இன்று கதை சொன்னேன்.	0.00	0.00	0.00	1
நான் இன்று கதை சொல்வேன்.	0.00	0.00	0.00	1
நான் இன்று சாப்பிட்டேன்.	1.00	1.00	1.00	1
நான் இன்று சினிமா பார்த்தேன்.	1.00	1.00	1.00	1
நான் இன்று பாடம் படித்தேன்.	0.00	0.00	0.00	1
நான் இன்று விளையாடினேன்.	0.00	0.00	0.00	1
நான் இப்போது ஓடுகிறேன்.	0.00	0.00	0.00	2
நான் இப்போது பாடம் படிக்கிறேன்.	0.33	1.00	0.50	1
நான் இப்போது பாடம் படித்தேன்.	0.00	0.00	0.00	1
நான் இப்போது விளையாடினேன்.	0.00	0.00	0.00	1
நான் இப்போது வேலை செய்தேன்.	0.00	0.00	0.00	1
நான் என் பாடத்தை முடித்துவிட்டேன்.	0.00	0.00	0.00	1
நான் தினமும் காலையில் பல் தேய்கிறேன்.	1.00	1.00	1.00	1
நான் நாளை கதை சொன்னேன்.	1.00	1.00	1.00	1
நான் நாளை கதை சொல்வேன்.	0.50	1.00	0.67	1
நான் நாளை படம் பார்ப்பேன்.	0.00	0.00	0.00	1
நான் நாளை பாடம் படித்தேன்.	0.00	0.00	0.00	3
நான் நாளை பாடம் படிப்பேன்.	0.25	1.00	0.40	2
நான் நாளை விளையாடினேன்.	0.00	0.00	0.00	1
நான் நாளை விளையாடுவேன்.	0.50	1.00	0.67	2
நான் நாளை வேலை செய்தேன்.	0.00	0.00	0.00	1
நான் நாளை வேலை செய்வேன்.	0.33	1.00	0.50	1
நான் நேற்று கதை சொன்னேன்.	0.00	0.00	0.00	0
நான் நேற்று பாடம் படித்தேன்.	0.50	1.00	0.67	2
நான் நேற்று விளையாடினேன்.	0.60	1.00	0.75	3
நான் நேற்று வேலை செய்தேன்.	1.00	1.00	1.00	1
accuracy			0.43	79
macro avg	0.21	0.34	0.24	79
weighted avg	0.26	0.43	0.31	79

```

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined and
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Recall is ill-defined and be
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined and
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Recall is ill-defined and be
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined and
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Recall is ill-defined and be
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))

```

```

# Test the model with a new sentence
test_sentence = ["அவளே குரலுடன் பாடினான்."]
test_vector = vectorizer.transform(test_sentence)
predicted = model.predict(test_vector)

```

```

# Display the result
print("Ungrammatical Sentence:", test_sentence[0])
print("Predicted Standard Tamil:", predicted[0])

```



Ungrammatical Sentence: அவளே குரலுடன் பாடினான்.  
Predicted Standard Tamil: அவள் பாடம் படிக்கிறாள்.

```

# Test the model with a new sentence
test_sentence = ["அவள் வேலை செய்தாள்."]
test_vector = vectorizer.transform(test_sentence)
predicted = model.predict(test_vector)

```

```

# Display the result
print("Ungrammatical Sentence:", test_sentence[0])
print("Predicted Standard Tamil:", predicted[0])

```

➡ Ungrammatical Sentence: அவள் வேலை செய்தான்.  
Predicted Standard Tamil: அவள் வேலை செய்தாள்.

```
# Test the model with a new sentence
test_sentence = ["நான் இன்று பாடம் படிக்கிறேன்."]
test_vector = vectorizer.transform(test_sentence)
predicted = model.predict(test_vector)
```

```
# Display the result
print("Ungrammatical Sentence:", test_sentence[0])
print("Predicted Standard Tamil:", predicted[0])
```

➡ Ungrammatical Sentence: நான் இன்று பாடம் படிக்கிறேன்.  
Predicted Standard Tamil: நான் நேற்று பாடம் படித்தேன்.

```
# Test the model with a new sentence
test_sentence = ["அவர்கள் கதையை கேட்டான்."]
test_vector = vectorizer.transform(test_sentence)
predicted = model.predict(test_vector)
```

```
# Display the result
print("Ungrammatical Sentence:", test_sentence[0])
print("Predicted Standard Tamil:", predicted[0])
```

➡ Ungrammatical Sentence: அவர்கள் கதையை கேட்டான்.  
Predicted Standard Tamil: அவள் கதை சொல்லினாள்.

```
# Test the model with a new sentence
test_sentence = ["நான் நேற்று பாடம் படிக்கிறேன்."]
test_vector = vectorizer.transform(test_sentence)
predicted = model.predict(test_vector)
```

```
# Display the result
print("Ungrammatical Sentence:", test_sentence[0])
print("Predicted Standard Tamil:", predicted[0])
```

➡ Ungrammatical Sentence: நான் நேற்று பாடம் படிக்கிறேன்.  
Predicted Standard Tamil: நான் நேற்று பாடம் படித்தேன்.