In [1]:

*# Import necessary libraries*

**from** sklearn.datasets **import** fetch\_20newsgroups

**from** sklearn.model\_selection **import** train\_test\_split

**from** sklearn.feature\_extraction.text **import** TfidfVectorizer

**from** sklearn.naive\_bayes **import** MultinomialNB

**from** sklearn.pipeline **import** make\_pipeline

**from** sklearn.metrics **import** classification\_report, accuracy\_score, confusion\_mat

**import** seaborn **as** sns

**import** matplotlib.pyplot **as** plt

In [11]:

*# 1. Load and preprocess the dataset*

data **=** fetch\_20newsgroups(subset**=**'all', shuffle**=True**, remove**=**('headers', 'footer

In [3]:

In [5]:

X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(data**.**data, data**.**target, test

*# Create a pipeline with TF-IDF vectorizer and Naive Bayes classifier*

model **=** make\_pipeline(TfidfVectorizer(stop\_words**=**'english', max\_df**=**0.5), Multino

*#Train the model*

print("Training the classifier...") model**.**fit(X\_train, y\_train)

y\_pred **=** model**.**predict(X\_test) Training the classifier...

*#Evaluate the model*

print("\nAccuracy:", accuracy\_score(y\_test, y\_pred)) print("\nClassification Report:")

print(classification\_report(y\_test, y\_pred, target\_names**=**data**.**target\_names))

Accuracy: 0.7222811671087533

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Classification Report: | precision | recall | f1-score | support |
| alt.atheism | 0.74 | 0.28 | 0.40 | 151 |
| comp.graphics | 0.70 | 0.68 | 0.69 | 202 |
| comp.os.ms-windows.misc | 0.68 | 0.66 | 0.67 | 195 |
| comp.sys.ibm.pc.hardware | 0.55 | 0.78 | 0.64 | 183 |
| comp.sys.mac.hardware | 0.87 | 0.67 | 0.76 | 205 |
| comp.windows.x | 0.90 | 0.81 | 0.85 | 215 |
| misc.forsale | 0.79 | 0.70 | 0.74 | 193 |
| rec.autos | 0.84 | 0.76 | 0.80 | 196 |
| rec.motorcycles | 0.49 | 0.77 | 0.60 | 168 |
| rec.sport.baseball | 0.92 | 0.83 | 0.88 | 211 |
| rec.sport.hockey | 0.88 | 0.92 | 0.90 | 198 |
| sci.crypt | 0.70 | 0.86 | 0.77 | 201 |
| sci.electronics | 0.85 | 0.63 | 0.72 | 202 |
| sci.med | 0.91 | 0.86 | 0.88 | 194 |
| sci.space | 0.80 | 0.83 | 0.82 | 189 |
| soc.religion.christian | 0.43 | 0.94 | 0.59 | 202 |
| talk.politics.guns | 0.70 | 0.80 | 0.75 | 188 |
| talk.politics.mideast | 0.79 | 0.83 | 0.81 | 182 |
| talk.politics.misc | 0.92 | 0.44 | 0.60 | 159 |
| talk.religion.misc | 0.80 | 0.03 | 0.06 | 136 |
| accuracy |  |  | 0.72 | 3770 |
| macro avg | 0.76 | 0.70 | 0.70 | 3770 |
| weighted avg | 0.76 | 0.72 | 0.71 | 3770 |

In [7]:

*#Confusion matrix heatmap*

plt**.**figure(figsize**=**(12, 10))

sns**.**heatmap(confusion\_matrix(y\_test, y\_pred), annot**=False**,

cmap**=**"Blues",

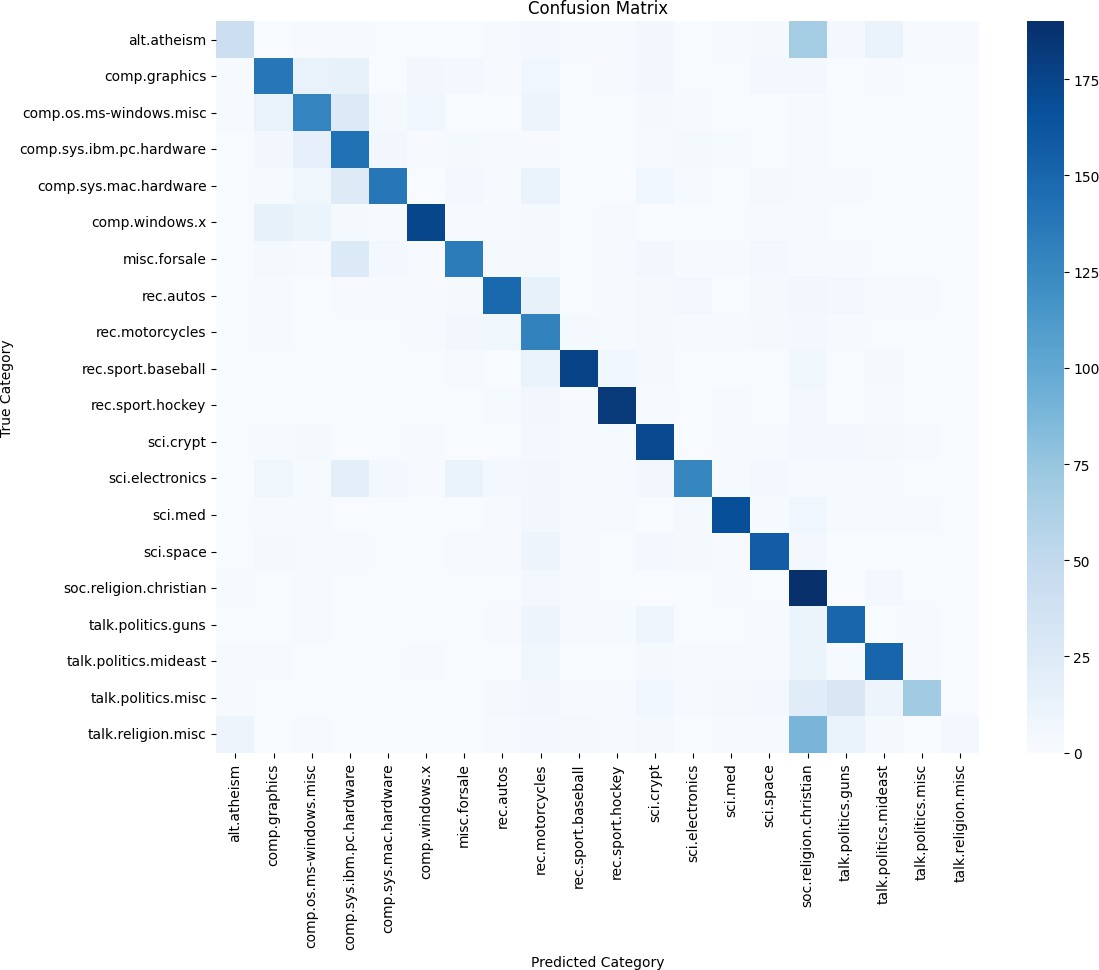
xticklabels**=**data**.**target\_names, yticklabels**=**data**.**target\_names)

plt**.**title("Confusion Matrix")

plt**.**xlabel("Predicted Category") plt**.**ylabel("True Category")

plt**.**xticks(rotation**=**90) plt**.**yticks(rotation**=**0) plt**.**tight\_layout()

plt**.**show()



In [ ]: