In [ ]:
```
# install packages
%pip install pandas numpy matplotlib seaborn joblib scikit-learn nltk
```

```
Requirement already satisfied: pandas in c:\users\sigei\appdata\local\programs\py
thon\python312\lib\site-packages (2.2.0)
Requirement already satisfied: numpy in c:\users\sigei\appdata\local\programs\pyt
hon\python312\lib\site-packages (1.26.4)
Requirement already satisfied: matplotlib in c:\users\sigei\appdata\local\program
s\python\python312\lib\site-packages (3.8.3)
Requirement already satisfied: seaborn in c:\users\sigei\appdata\local\programs\p
ython\python312\lib\site-packages (0.13.2)
Requirement already satisfied: joblib in c:\users\sigei\appdata\local\programs\py
thon\python312\lib\site-packages (1.3.2)
Requirement already satisfied: scikit-learn in c:\users\sigei\appdata\local\progr
ams\python\python312\lib\site-packages (1.4.1.post1)
Requirement already satisfied: nltk in c:\users\sigei\appdata\local\programs\pyth
on\python312\lib\site-packages (3.8.1)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\sigei\appdata\r
oaming\python\python312\site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\sigei\appdata\local\progr
ams\python\python312\lib\site-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\sigei\appdata\local\pro
grams\python\python312\lib\site-packages (from pandas) (2024.1)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\sigei\appdata\local\p
rograms\python\python312\lib\site-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\sigei\appdata\local\progr
ams\python\python312\lib\site-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\sigei\appdata\local
\programs\python\python312\lib\site-packages (from matplotlib) (4.49.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\sigei\appdata\local
\programs\python\python312\lib\site-packages (from matplotlib) (1.4.5)
Requirement already satisfied: packaging>=20.0 in c:\users\sigei\appdata\local\pr
ograms\python\python312\lib\site-packages (from matplotlib) (23.2)
Requirement already satisfied: pillow>=8 in c:\users\sigei\appdata\local\programs
\python\python312\lib\site-packages (from matplotlib) (10.2.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\sigei\appdata\local\p
rograms\python\python312\lib\site-packages (from matplotlib) (3.1.1)
Requirement already satisfied: scipy>=1.6.0 in c:\users\sigei\appdata\local\progr
ams\python\python312\lib\site-packages (from scikit-learn) (1.12.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\sigei\appdata\loc
al\programs\python\python312\lib\site-packages (from scikit-learn) (3.3.0)
Requirement already satisfied: click in c:\users\sigei\appdata\local\programs\pyt
hon\python312\lib\site-packages (from nltk) (8.1.7)
Requirement already satisfied: regex>=2021.8.3 in c:\users\sigei\appdata\local\pr
ograms\python\python312\lib\site-packages (from nltk) (2023.12.25)
Requirement already satisfied: tqdm in c:\users\sigei\appdata\local\programs\pyth
on\python312\lib\site-packages (from nltk) (4.66.2)
Requirement already satisfied: six>=1.5 in c:\users\sigei\appdata\local\programs
\python\python312\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.
0)
Requirement already satisfied: colorama in c:\users\sigei\appdata\local\programs
\python\python312\lib\site-packages (from click->nltk) (0.4.6)
Note: you may need to restart the kernel to use updated packages.
```

importing libraries

In [ ]:
```
# Import libraries
import pandas as pd
import nltk
import re
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline
from sklearn.metrics import classification_report, accuracy_score
import joblib
import matplotlib.pyplot as plt
import seaborn as sns
```

download nltk data

In [ ]:
```
# Download NLTK resources
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\sigei\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\sigei\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\sigei\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

Out[ ]: True

dataset

In [ ]:
```
# Load the dataset
data = pd.read_csv("./fake_reviews_dataset.csv")
```

Preposeccessing technique

In [ ]:
```
# Data preprocessing
stop_words = set(stopwords.words('english'))
lemmatizer = WordNetLemmatizer()

def preprocess_text(text):
    text = text.lower()
    text = re.sub(r'[^\w\s]', '', text)
    tokens = word_tokenize(text)
    tokens = [lemmatizer.lemmatize(word) for word in tokens if word not in stop_
    return ' '.join(tokens)

data['clean_text'] = data['text_'].apply(preprocess_text)
```

Feature engineering

In [ ]:
```
# Feature Engineering
X = data['clean_text']
y = data['label']
```

Splitting data into train and test sets, 20% test set size and random state 42

In [ ]:
```
# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_
```

Pipeline definition

In [ ]:
```
# Define pipeline
pipeline = Pipeline([
    ('tfidf', TfidfVectorizer()),
    ('clf', LogisticRegression(max_iter=1000))
])
```

Hypermeter definition

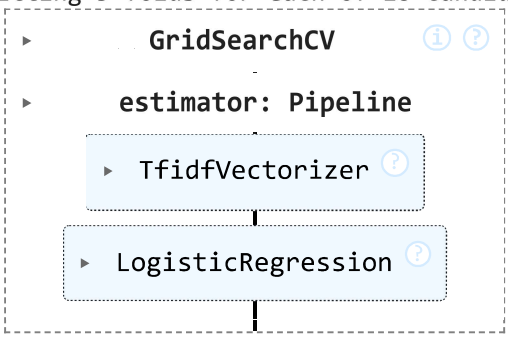
In [ ]:
```
# Define hyperparameters grid for grid search
param_grid = {
    'tfidf__max_features': [1000, 2000, 3000],
    'tfidf__ngram_range': [(1, 1), (1, 2)],
    'clf__C': [0.1, 1, 10]
}
```

Grid search for the best hyperparameters for the model

In [ ]:
```
# Grid search for hyperparameter tuning
grid_search = GridSearchCV(pipeline, param_grid, cv=5, verbose=2, n_jobs=-1)
grid_search.fit(X_train, y_train)
```

```
Fitting 5 folds for each of 18 candidates, totalling 90 fits
```

Out[ ]:



Best hyperparameter

In [ ]:
```
# Best hyperparameters
best_params = grid_search.best_params_
print("Best Hyperparameters:", best_params)
```

```
Best Hyperparameters: {'clf__C': 10, 'tfidf__max_features': 3000, 'tfidf__ngram_r
ange': (1, 2)}
```

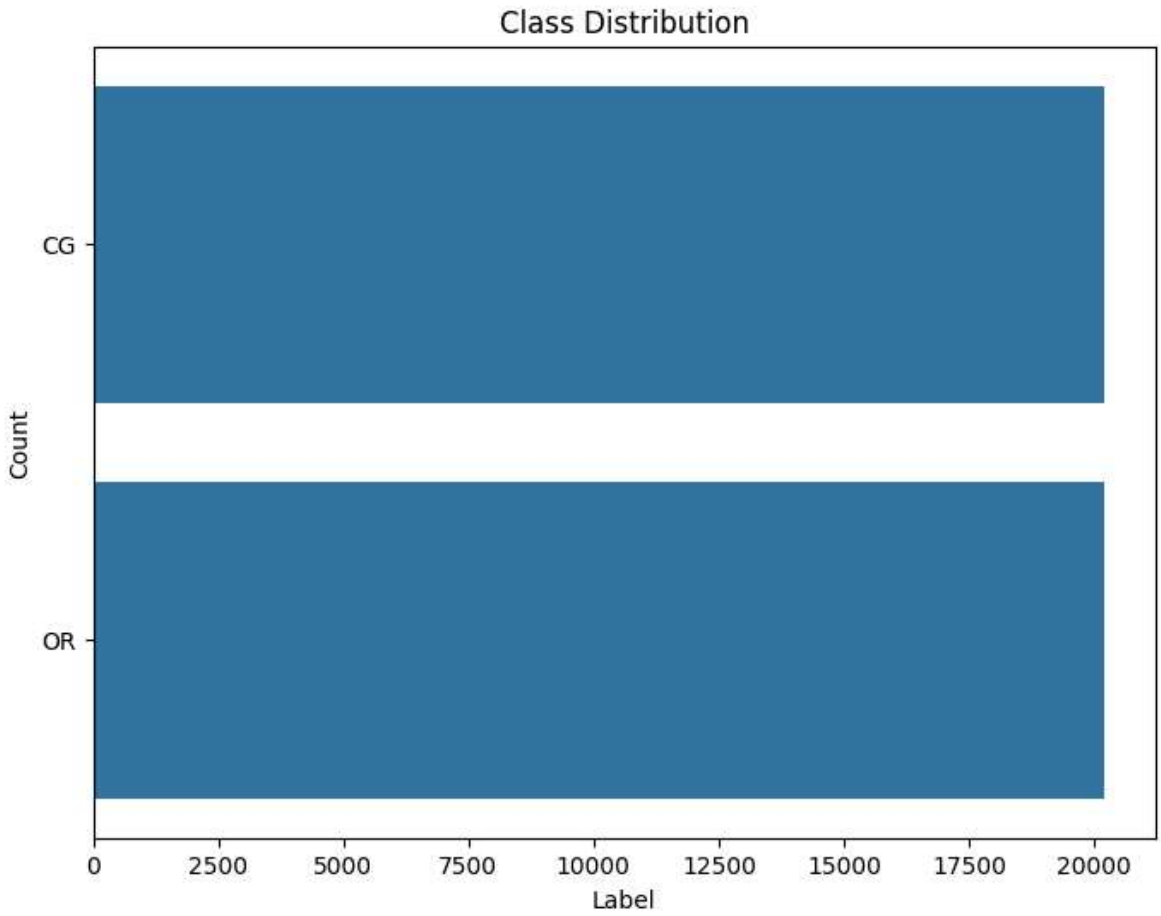
Model Description

In [ ]:
```
# Evaluate model
y_pred = grid_search.predict(X_test)
print("Classification Report:")
print(classification_report(y_test, y_pred))
print("Accuracy:", accuracy_score(y_test, y_pred))
```

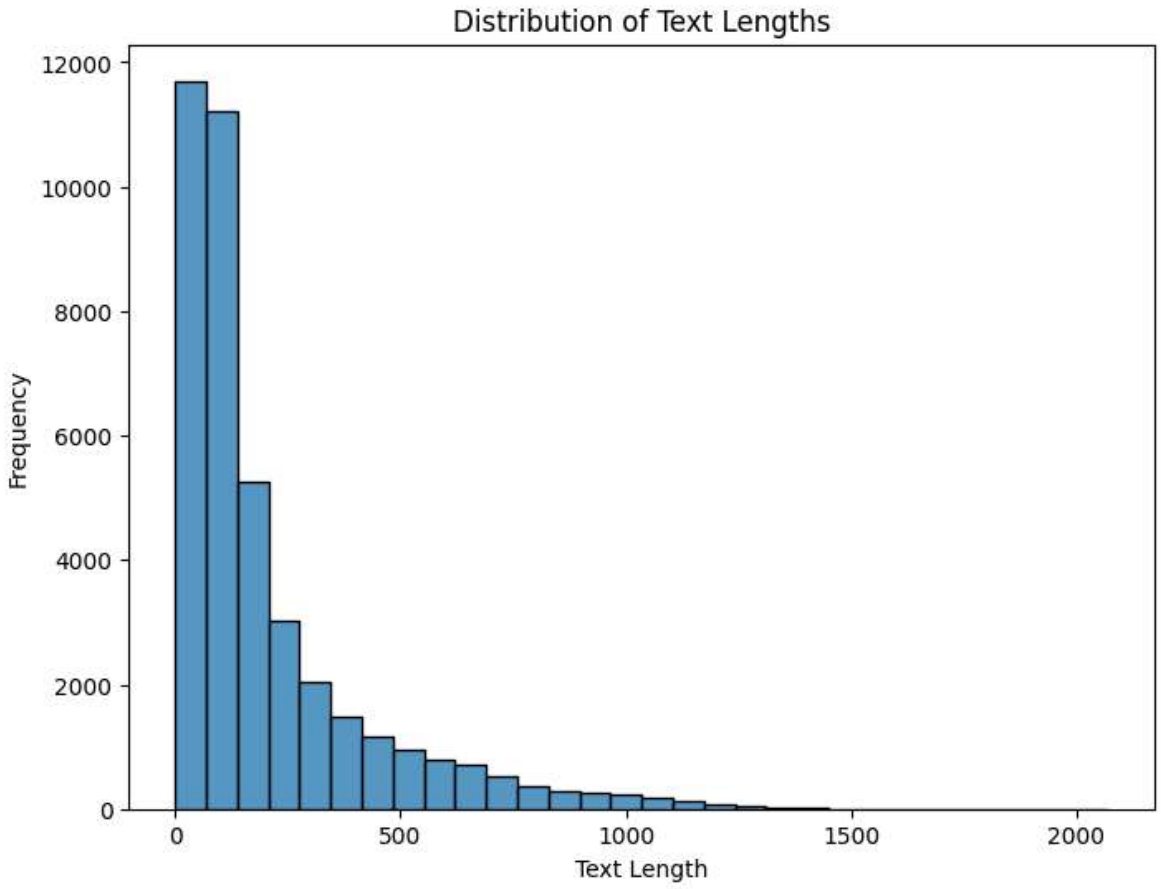
```
Classification Report:
              precision    recall  f1-score   support

          CG       0.89      0.88      0.89      4016
          OR       0.88      0.89      0.89      4071

    accuracy                           0.89      8087
   macro avg       0.89      0.89      0.89      8087
weighted avg       0.89      0.89      0.89      8087

Accuracy: 0.886608136515395
```

Vizualization

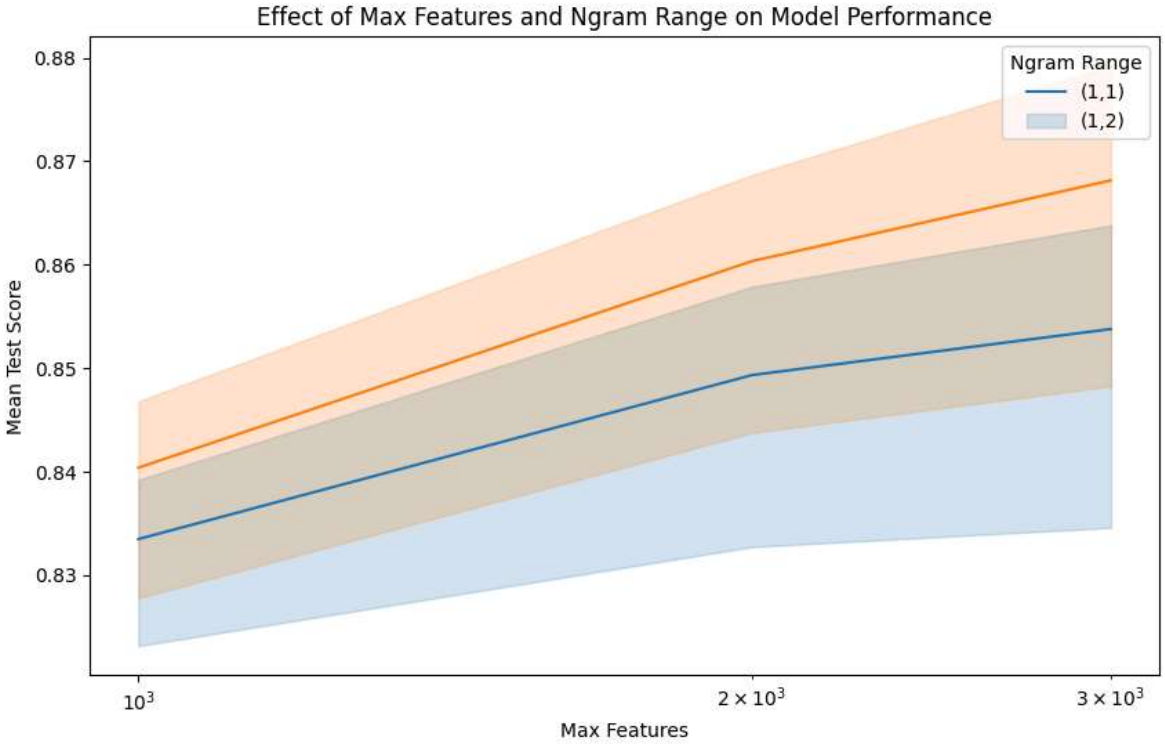
In [ ]:
```
# Visualization of class distribution in the dataset
plt.figure(figsize=(8, 6))
sns.countplot(data['label'])
plt.title('Class Distribution')
plt.xlabel('Label')
plt.ylabel('Count')
plt.show()
```



In [ ]:
```
# Visualization of distribution of text lengths
text_lengths = data['clean_text'].apply(len)
plt.figure(figsize=(8, 6))
sns.histplot(text_lengths, bins=30)
plt.title('Distribution of Text Lengths')
plt.xlabel('Text Length')
plt.ylabel('Frequency')
plt.show()
```



In [ ]:
```
# Visualization of effect of Hyperparameters on Model Performance
param_results = pd.DataFrame(grid_search.cv_results_)
plt.figure(figsize=(10, 6))
sns.lineplot(data=param_results, x='param_tfidf__max_features', y='mean_test_sco
plt.xscale('log')
plt.title('Effect of Max Features and Ngram Range on Model Performance')
plt.xlabel('Max Features')
plt.ylabel('Mean Test Score')
plt.legend(title='Ngram Range', labels=['(1,1)', '(1,2)'])
plt.show()
```


Effect of Max Features and Ngram Range on Model Performance
Ngram Range
(1,1)
(1,2)
Mean Test Score
0.88
0.87
0.86
0.85
0.84
0.83
10^3
2×10^3
3×10^3
Max Features