```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
data=pd.read_csv("/content/fake_news_dataset.csv")
```

```
data.head()
```

	id	title	author	text	state	date_published	source	category	sen
0	1	Breaking News 1	Jane Smith	This is the content of article 1. It contains ...	Tennessee	30-11-2021	The Onion	Entertainment	
1	2	Breaking News 2	Emily Davis	This is the content of article 2. It contains ...	Wisconsin	02-09-2021	The Guardian	Technology	
2	3	Breaking News 3	John Doe	This is the content of article 3. It contains ...	Missouri	13-04-2021	New York Times	Sports	
3	4	Breaking News 4	Alex Johnson	This is the content of article 4. It contains ...	North Carolina	08-03-2020	CNN	Sports	
4	5	Breaking News 5	Emily Davis	This is the content of article 5. It contains ...	California	23-03-2022	Daily Mail	Technology	

5 rows × 24 columns

```
data.drop_duplicates(inplace=True)
```

data

	id	title	author	text	state	date_published	source	categor
0	1	Breaking News 1	Jane Smith	This is the content of article 1. It contains ...	Tennessee	30-11-2021	The Onion	Entertainmer
1	2	Breaking News 2	Emily Davis	This is the content of article 2. It contains ...	Wisconsin	02-09-2021	The Guardian	Technolog
2	3	Breaking News 3	John Doe	This is the content of article 3. It contains ...	Missouri	13-04-2021	New York Times	Sport
3	4	Breaking News 4	Alex Johnson	This is the content of article 4. It contains ...	North Carolina	08-03-2020	CNN	Sport
4	5	Breaking News 5	Emily Davis	This is the content of article 5. It contains ...	California	23-03-2022	Daily Mail	Technolog
...	...	...	...	...	...	...	...	.
3995	3996	Breaking News 3996	John Doe	This is the content of article 3996. It contai...	Ohio	25-04-2020	InfoWars	Technolog
3996	3997	Breaking News 3997	Alex Johnson	This is the content of article 3997. It contai...	Washington	09-01-2022	CNN	Sport
				This is the				

3997	3998	Breaking News 3998	Alex Johnson	the content of article 3998. It contai...	California	03-03-2023	Breitbart	Entertainmer
3998	3999	Breaking News 3999	John Doe	This is the content of article 3999. It contai...	Illinois	13-04-2021	New York Times	Healt
3999	4000	Breaking News 4000	John Doe	This is the content of article 4000. It contai...	Texas	20-12-2023	The Guardian	Healt

4000 rows × 24 columns

```
df.drop_duplicates()
```

```
data.columns
```

```
Index(['id', 'title', 'author', 'text', 'state', 'date_published', 'source',
       'category', 'sentiment_score', 'word_count', 'char_count', 'has_images',
       'has_videos', 'readability_score', 'num_shares', 'num_comments',
       'political_bias', 'fact_check_rating', 'is_satirical', 'trust_score',
       'source_reputation', 'clickbait_score', 'plagiarism_score', 'label'],
      dtype='object')
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4000 entries, 0 to 3999
Data columns (total 24 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 4000 non-null   int64
 1   title              4000 non-null   object
 2   author             4000 non-null   object
 3   text               4000 non-null   object
 4   state              4000 non-null   object
 5   date_published     4000 non-null   object
 6   source             4000 non-null   object
 7   category           4000 non-null   object
 8   sentiment_score    4000 non-null   float64
 9   word_count         4000 non-null   int64
 10  char_count         4000 non-null   int64
 11  has_images         4000 non-null   int64
 12  has_videos         4000 non-null   int64
 13  readability_score  4000 non-null   float64
 14  num_shares         4000 non-null   int64
 15  num_comments       4000 non-null   int64
 16  political_bias     4000 non-null   object
 17  fact_check_rating  4000 non-null   object
 18  is_satirical       4000 non-null   int64
```

```
 19  trust_score        4000 non-null   int64
 20  source_reputation  4000 non-null   int64
 21  clickbait_score    4000 non-null   float64
 22  plagiarism_score   4000 non-null   float64
 23  label              4000 non-null   object
dtypes: float64(4), int64(10), object(10)
memory usage: 750.1+ KB
```

```
data.isnull().sum()
```

	0
id	0
title	0
author	0
text	0
state	0
date_published	0
source	0
category	0
sentiment_score	0
word_count	0
char_count	0
has_images	0
has_videos	0
readability_score	0
num_shares	0
num_comments	0
political_bias	0
fact_check_rating	0
is_satirical	0
trust_score	0
source_reputation	0
clickbait_score	0
plagiarism_score	0
label	0

dtype: int64

```
data
```

	id	title	author	text	state	date_published	source	categor
0	1	Breaking News 1	Jane Smith	This is the content of article 1. It contains ...	Tennessee	30-11-2021	The Onion	Entertainmer
1	2	Breaking News 2	Emily Davis	This is the content of article 2. It contains ...	Wisconsin	02-09-2021	The Guardian	Technolog
2	3	Breaking News 3	John Doe	This is the content of article 3. It contains ...	Missouri	13-04-2021	New York Times	Sport
3	4	Breaking News 4	Alex Johnson	This is the content of article 4. It contains ...	North Carolina	08-03-2020	CNN	Sport
4	5	Breaking News 5	Emily Davis	This is the content of article 5. It contains ...	California	23-03-2022	Daily Mail	Technolog
...	...	...	...	...	...	...	...	.
3995	3996	Breaking News 3996	John Doe	This is the content of article 3996. It contai...	Ohio	25-04-2020	InfoWars	Technolog
				This is				
			Johnson					

```
data.drop_duplicates(inplace=True)
```

```
data
```

	id	title	author	text	state	date_published	source	categor
3997	3998	Breaking News	Alex	the content of	California	03-03-2023	Breitbart	Entertainmer
0	1	Breaking News 1	Jane Smith	This is the content of article 1. It contains ...	Tennessee	30-11-2021	The Onion	Entertainmer
1	2	Breaking News 2	Emily Davis	This is the content of article 2. It contains ...	Wisconsin	02-09-2021	The Guardian	Technolog
3999	4000	Breaking News 4000	John Doe	This is the content of article 4000. It	Texas	20-12-2023	The Guardian	Healt
2	3	Breaking News 3	John Doe	This is the content of article 3. It contains ...	Missouri	13-04-2021	New York Times	Sport
3	4	Breaking News 4	Alex Johnson	This is the content of article 4. It contains ...	North Carolina	08-03-2020	CNN	Sport
4	5	Breaking News 5	Emily Davis	This is the content of article 5. It contains ...	California	23-03-2022	Daily Mail	Technolog
...	...	...	...	...	...	...	...	.
3995	3996	Breaking News 3996	John Doe	This is the content of article 3996. It contai...	Ohio	25-04-2020	InfoWars	Technolog
3996	3997	Breaking News 3997	Alex Johnson	This is the content of article 3997. It contai...	Washington	09-01-2022	CNN	Sport
				This is the				

3997	3998	Breaking News 3998	Alex Johnson	the content of article 3998. It contai...	California	03-03-2023	Breitbart	Entertainmer
3998	3999	Breaking News 3999	John Doe	This is the content of article 3999. It contai...	Illinois	13-04-2021	New York Times	Healt
3999	4000	Breaking News 4000	John Doe	This is the content of article 4000. It contai...	Texas	20-12-2023	The Guardian	Healt

4000 rows × 24 columns

```
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
data_scaled=data.copy()
data_scaled[["clickbait_score","plagiarism_score"]]=scaler.fit_transform(data[["clickbait_score","
data_scaled
```

	id	title	author	text	state	date_published	source	categor
0	1	Breaking News 1	Jane Smith	This is the content of article 1. It contains ...	Tennessee	30-11-2021	The Onion	Entertainmer
1	2	Breaking News 2	Emily Davis	This is the content of article 2. It contains ...	Wisconsin	02-09-2021	The Guardian	Technolog
2	3	Breaking News 3	John Doe	This is the content of article 3. It contains ...	Missouri	13-04-2021	New York Times	Sport
3	4	Breaking News 4	Alex Johnson	This is the content of article 4. It contains ...	North Carolina	08-03-2020	CNN	Sport
4	5	Breaking News 5	Emily Davis	This is the content of article 5. It contains ...	California	23-03-2022	Daily Mail	Technolog
...	...	...	...	...	...	...	...	.
3995	3996	Breaking News 3996	John Doe	This is the content of article 3996. It contai...	Ohio	25-04-2020	InfoWars	Technolog
3996	3997	Breaking News 3997	Alex Johnson	This is the content of article 3997. It contai...	Washington	09-01-2022	CNN	Sport
				This is the				

3997	3998	Breaking News 3998	Alex Johnson	the content of article 3998. It contai...	California	03-03-2023	Breitbart	Entertainmer
3998	3999	Breaking News 3999	John Doe	This is the content of article 3999. It contai...	Illinois	13-04-2021	New York Times	Healt
3999	4000	Breaking News 4000	John Doe	This is the content of article 4000. It contai...	Texas	20-12-2023	The Guardian	Healt

4000 rows × 24 columns

```
from sklearn.preprocessing import MinMaxScaler
Scaler=MinMaxScaler()

data_scaled[["clickbait_score","plagiarism_score"]]=Scaler.fit_transform(data[["clickbait_score","
data_scaled
```

	id	title	author	text	state	date_published	source	categor
0	1	Breaking News 1	Jane Smith	This is the content of article 1. It contains ...	Tennessee	30-11-2021	The Onion	Entertainmer
1	2	Breaking News 2	Emily Davis	This is the content of article 2. It contains ...	Wisconsin	02-09-2021	The Guardian	Technolog
2	3	Breaking News 3	John Doe	This is the content of article 3. It contains ...	Missouri	13-04-2021	New York Times	Sport
3	4	Breaking News 4	Alex Johnson	This is the content of article 4. It contains ...	North Carolina	08-03-2020	CNN	Sport
4	5	Breaking News 5	Emily Davis	This is the content of article 5. It contains ...	California	23-03-2022	Daily Mail	Technolog
...	...	...	...	...	...	...	...	.
3995	3996	Breaking News 3996	John Doe	This is the content of article 3996. It contai...	Ohio	25-04-2020	InfoWars	Technolog
3996	3997	Breaking News 3997	Alex Johnson	This is the content of article 3997. It contai...	Washington	09-01-2022	CNN	Sport
				This is the				

3997	3998	Breaking News 3998	Alex Johnson	the content of article 3998. It contai...	California	03-03-2023	Breitbart	Entertainmer
3998	3999	Breaking News 3999	John Doe	This is the content of article 3999. It contai...	Illinois	13-04-2021	New York Times	Healt
3999	4000	Breaking News 4000	John Doe	This is the content of article 4000. It contai...	Texas	20-12-2023	The Guardian	Healt

4000 rows × 24 columns

```
data_encoded=pd.get_dummies(data,columns=["label"],drop_first=True)
print(data_encoded)
```

```
3996  This is the content of article 3997. It contai...      Washington
3997  This is the content of article 3998. It contai...      California
3998  This is the content of article 3999. It contai...        Illinois
3999  This is the content of article 4000. It contai...           Texas

     date_published          source       category  sentiment_score  \
0        30-11-2021       The Onion  Entertainment            -0.22
1        02-09-2021    The Guardian     Technology             0.92
2        13-04-2021  New York Times         Sports             0.25
3        08-03-2020             CNN         Sports             0.94
4        23-03-2022      Daily Mail     Technology            -0.01
...             ...             ...            ...              ...
3995     25-04-2020        InfoWars     Technology             0.91
3996     09-01-2022             CNN         Sports            -0.57
```

```
3                 TRUE             1            18                 10
4                Mixed             0            95                  6
...                ...           ...           ...                ...
3995             Mixed             0            29                 10
3996             FALSE             1            53                  3
3997             FALSE             0            22                  9
3998             FALSE             1             3                  6
3999              TRUE             1            73                  4

     clickbait_score  plagiarism_score  label_Real
0               0.84             53.35       False
1               0.85             28.28       False
2               0.72              0.38       False
3               0.92             32.20       False
4               0.66             77.70        True
...              ...               ...         ...
3995            0.22             95.46       False
3996            0.42             16.54       False
3997            0.50             28.51       False
3998            0.17             71.16        True
3999            0.09             27.65        True

[4000 rows x 24 columns]
```

```
from sklearn.preprocessing import LabelEncoder
encoder=LabelEncoder()
data["label"]=encoder.fit_transform(data["label"])
print(data)
```

```
         id               title        author  \
0        1     Breaking News 1    Jane Smith
1        2     Breaking News 2   Emily Davis
2        3     Breaking News 3      John Doe
3        4     Breaking News 4  Alex Johnson
4        5     Breaking News 5   Emily Davis
...    ...                 ...           ...
3995  3996  Breaking News 3996      John Doe
3996  3997  Breaking News 3997  Alex Johnson
3997  3998  Breaking News 3998  Alex Johnson
3998  3999  Breaking News 3999      John Doe
3999  4000  Breaking News 4000      John Doe

                                                 text           state  \
0     This is the content of article 1. It contains ...       Tennessee
1     This is the content of article 2. It contains ...       Wisconsin
2     This is the content of article 3. It contains ...        Missouri
3     This is the content of article 4. It contains ...  North Carolina
4     This is the content of article 5. It contains ...      California
...                                                 ...             ...
3995  This is the content of article 3996. It contai...            Ohio
3996  This is the content of article 3997. It contai...      Washington
3997  This is the content of article 3998. It contai...      California
3998  This is the content of article 3999. It contai...        Illinois
3999  This is the content of article 4000. It contai...           Texas

     date_published          source       category  sentiment_score  \
0        30-11-2021       The Onion  Entertainment            -0.22
1        02-09-2021    The Guardian     Technology             0.92
2        13-04-2021  New York Times         Sports             0.25
3        08-03-2020             CNN         Sports             0.94
4        23-03-2022      Daily Mail     Technology            -0.01
...             ...             ...            ...              ...
3995     25-04-2020        InfoWars     Technology             0.91
3996     09-01-2022             CNN         Sports            -0.57
3997     03-03-2023       Breitbart  Entertainment            -0.17
```

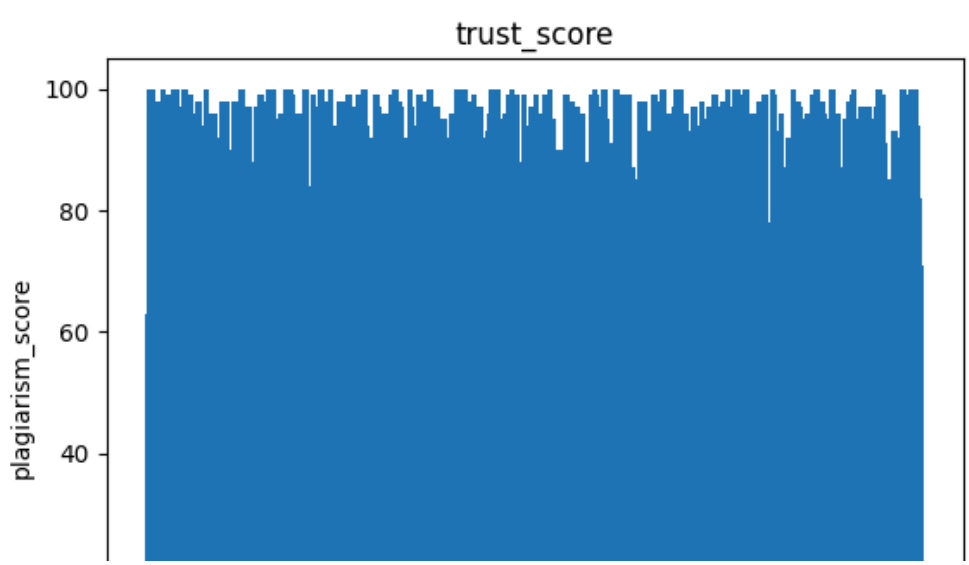
```
3998     13-04-2021  New York Times         Health            -0.88
3999     20-12-2023    The Guardian         Health            -0.95

      word_count  ...  num_shares  num_comments  political_bias  \
0           1302  ...       47305           450          Center
1            322  ...       39804           530            Left
2            228  ...       45860           763          Center
3            155  ...       34222           945          Center
4            962  ...       35934           433           Right
...          ...  ...         ...           ...             ...
3995        1227  ...       38880           697           Right
3996        1296  ...        3650           925            Left
3997         522  ...       35391           577            Left
3998         169  ...       40424           201            Left
3999         465  ...       48913           279           Right

      fact_check_rating  is_satirical  trust_score source_reputation  \
0                 FALSE             1           76                 6
1                 Mixed             1            1                 5
2                 Mixed             0           57                 1
3                  TRUE             1           18                10
4                 Mixed             0           95                 6
```

```
def performance_category(clickbait_score):
  if clickbait_score>=0.80:
    return "High"
  elif clickbait_score>=0.50:
    return "Medium"
  else:
    return "Low"


data["performance"]=data["clickbait_score"].apply(performance_category)
print (data)
```

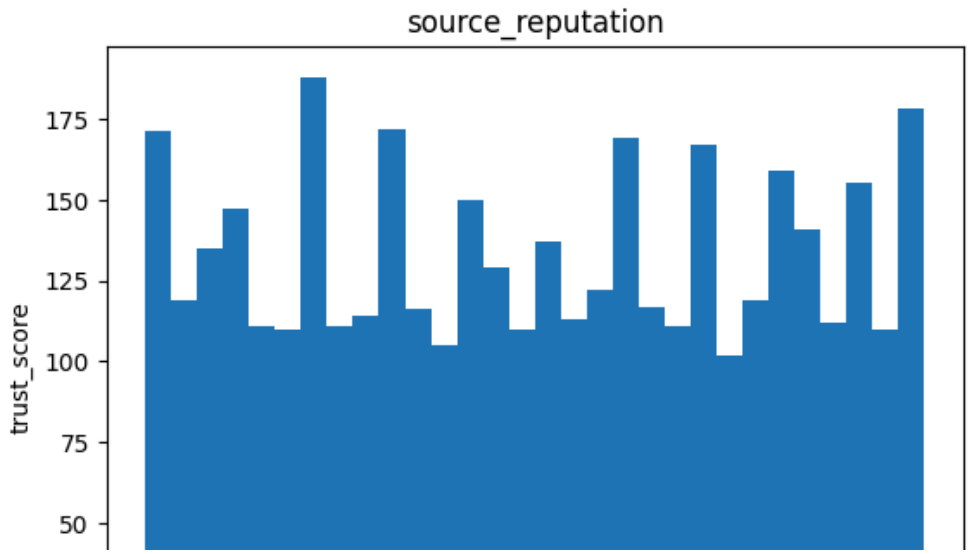
```
         id               title        author  \
0        1     Breaking News 1    Jane Smith
1        2     Breaking News 2   Emily Davis
2        3     Breaking News 3      John Doe
3        4     Breaking News 4  Alex Johnson
4        5     Breaking News 5   Emily Davis
...    ...                 ...           ...
3995  3996  Breaking News 3996      John Doe
3996  3997  Breaking News 3997  Alex Johnson
3997  3998  Breaking News 3998  Alex Johnson
3998  3999  Breaking News 3999      John Doe
3999  4000  Breaking News 4000      John Doe

                                                 text           state  \
0     This is the content of article 1. It contains ...       Tennessee
1     This is the content of article 2. It contains ...       Wisconsin
2     This is the content of article 3. It contains ...        Missouri
3     This is the content of article 4. It contains ...  North Carolina
4     This is the content of article 5. It contains ...      California
...                                                 ...             ...
3995  This is the content of article 3996. It contai...            Ohio
3996  This is the content of article 3997. It contai...      Washington
3997  This is the content of article 3998. It contai...      California
3998  This is the content of article 3999. It contai...        Illinois
3999  This is the content of article 4000. It contai...           Texas

     date_published          source       category  sentiment_score  \
0        30-11-2021       The Onion  Entertainment            -0.22
1        02-09-2021    The Guardian     Technology             0.92
```

```
2        13-04-2021  New York Times         Sports             0.25
3        08-03-2020             CNN         Sports             0.94
4        23-03-2022      Daily Mail     Technology            -0.01
...             ...             ...            ...              ...
3995     25-04-2020        InfoWars     Technology             0.91
3996     09-01-2022             CNN         Sports            -0.57
3997     03-03-2023       Breitbart  Entertainment            -0.17
3998     13-04-2021  New York Times         Health            -0.88
3999     20-12-2023    The Guardian         Health            -0.95

      word_count  ...  num_comments  political_bias  fact_check_rating  \
0           1302  ...           450          Center              FALSE
1            322  ...           530            Left              Mixed
2            228  ...           763          Center              Mixed
3            155  ...           945          Center               TRUE
4            962  ...           433           Right              Mixed
...          ...  ...           ...             ...                ...
3995        1227  ...           697           Right              Mixed
3996        1296  ...           925            Left              FALSE
3997         522  ...           577            Left              FALSE
3998         169  ...           201            Left              FALSE
3999         465  ...           279           Right               TRUE

      is_satirical  trust_score  source_reputation clickbait_score  \
0                1           76                  6            0.84
1                1            1                  5            0.85
2                0           57                  1            0.72
3                1           18                 10            0.92
```

```
plt.bar(data["plagiarism_score"],data["trust_score"])
plt.xlabel("clickbait_score")
plt.ylabel("plagiarism_score")
plt.title("trust_score")
plt.show()
```




```
plt.hist(data["trust_score"], bins=30)
plt.xlabel("clickbait_score")
plt.ylabel("trust_score")
plt.title("source_reputation")
plt.show()
```




```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import seaborn as sns


# Assuming 'data' is your processed DataFrame
X = data.drop('label', axis=1)  # Features
y = data['label']  # Target variable


# Convert non-numeric columns to numerical using one-hot encoding if needed
X = pd.get_dummies(X, drop_first=True)


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


# Initialize and train a Logistic Regression model (you can choose other models)
model = LogisticRegression()
model.fit(X_train, y_train)
```

```
/usr/local/lib/python3.11/dist-packages/sklearn/linear_model/_logistic.py:465: C
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regressio
  n_iter_i = _check_optimize_result(
```

```
▾ LogisticRegression
LogisticRegression()
```

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

```
[0 0 0 0 0 1 0 1 1 1 1 1 1 0 0 0 0 1 0 0 1 1 0 1 0 0 0 0 1 0 0 0 0 0 0 1 0
 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 1 0 0 1 0 1 0 0 0 1 1 0 0 1 0 0 0 1 0 0 0 1
 0 0 0 1 0 1 0 0 1 1 0 1 0 1 0 0 1 0 0 1 1 1 0 0 0 1 0 0 0 1 0 1 1 0 0 0 0
 0 0 1 1 0 0 1 1 0 0 1 1 0 1 1 1 0 1 0 1 0 1 0 1 1 1 0 0 0 1 1 0 1 0 1 0 0
 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 1 1 1 0 1 0 1 1 1 1 0 0 1 0 1 1
 1 0 0 1 0 0 0 0 0 0 1 1 1 1 0 1 1 0 1 0 0 1 1 1 0 0 0 0 0 0 1 0 1 1 0 1 1
 0 0 1 1 0 1 1 0 1 0 0 0 1 1 0 0 1 0 1 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 1 0
 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 1 0 1 0 0 0 1 1 0 1 0 0 0 0 0 1 0
 1 0 1 1 0 0 1 0 0 0 0 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 1 0 1 1 0 1 0 0 0 0 1
 0 0 1 1 1 1 0 0 0 1 1 0 1 0 1 0 0 1 1 0 0 0 0 1 1 0 0 0 0 0 1 0 1 0 0 1 0
 1 1 1 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1
 1 1 1 1 0 1 1 0 1 1 1 0 0 0 0 0 1 0 0 1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 0 1 1
 0 1 1 0 0 1 1 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 1 0
 1 0 1 0 1 0 0 1 1 1 1 0 0 0 0 1 1 1 0 1 1 1 1 1 0 0 1 0 0 0 0 1 0 0 0 1 1
 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 0 0 0 1 0 0 0 1 1
 0 0 1 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 1 0 0 0 1 1 0 1 0 0 0 1 0 1 1 0 1 0 1
 1 0 0 0 0 0 0 1 0 1 1 1 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 0 1 1 0 0
 0 1 0 1 1 1 0 0 1 0 1 1 0 0 1 1 1 1 0 1 0 1 1 0 0 0 0 0 0 0 1 1 0 0 0 1 0
 0 0 0 1 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 1 0 1 0 1 0 0 0 0 0 1 0 0 0
 1 0 0 1 1 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 1
 1 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 1 0 0 0 0 1 1 1 1 0 0 0 0
 1 0 0 0 1 0 0 1 0 1 0 0 0 1 0 1 0 0 0 1 0 0 0] y_Prediction
```

```
#random forest
from sklearn.ensemble import RandomForestClassifier
model=RandomForestClassifier()
model.fit(X_train,y_train)
```

```
▾ RandomForestClassifier
RandomForestClassifier()
```

```
#evaluate
y_pred=model.predict(X_test)
print(y_pred)
```

```
[1 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 1 1 0 0 1 1 0 0 0 1 1 0 0 0 0 1 0 0 1 0 1
 1 1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 1 1 0 0 1
 0 0 1 1 0 0 0 0 1 1 1 1 0 0 1 1 1 0 0 0 0 1 0 1 0 1 1 0 1 1 0 0 1 1 1 1 1
 0 1 1 1 1 0 1 1 0 0 0 1 0 1 0 0 1 0 1 0 0 1 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0
 1 1 1 0 1 1 1 1 1 0 0 1 1 0 1 1 1 0 0 1 0 1 1 0 1 0 0 1 1 1 1 1 0 1 0 1 1
 1 0 0 1 0 0 0 1 0 1 0 1 0 1 1 1 0 0 1 0 1 1 0 1 0 0 1 1 0 1 1 0 1 1 1 1 0
 0 1 1 0 0 0 1 0 0 0 0 0 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 0 1 1 0 0 1 0
 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0 0 0 1 0 0 1 0 1 0 0 0 1 0 1 1 1 0 0 0 0 1 0
 0 1 0 1 1 1 1 0 1 0 0 1 1 1 0 1 1 0 0 0 1 0 0 0 1 1 1 1 0 1 0 1 0 1 1 1 0
 1 1 0 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 1 0 1 0 0 1 0 0 0 1 0 1 0 0 1 1
 1 0 0 1 0 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 1 0 1 1 1 0 1 1 1 0 1 1 1 1 1 1 0
 1 0 0 0 0 0 1 0 0 1 1 0 0 1 0 1 0 0 1 1 1 1 0 0 1 1 0 1 1 0 0 0 0 1 1 0 1
 1 0 1 0 1 1 1 1 0 1 1 0 1 0 0 1 1 0 1 1 0 0 0 1 1 0 0 1 0 0 0 1 0 0 1 1 1
 1 1 0 1 0 1 0 0 0 0 1 1 0 1 0 1 1 0 0 1 1 1 1 1 0 1 0 1 1 0 0 1 0 1 0 1 1
 0 0 1 0 1 0 0 0 0 0 0 1 1 1 0 1 0 0 1 0 0 0 1 1 1 1 1 1 0 1 1 0 1 0 0 1 1
 0 1 1 1 0 0 0 1 1 1 0 1 0 0 0 0 0 0 0 0 1 0 1 1 0 1 1 1 1 0 0 1 0 0 0 1 0
 0 0 1 1 0 1 1 0 1 1 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 1 0 0 1 0 1 0 0 0
 0 1 1 1 0 1 1 1 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 1 1 1 0 0 1 0 1 0 0
 0 0 1 0 0 1 0 0 1 1 1 0 0 0 1 1 1 1 0 0 0 1 0 1 1 0 1 1 0 1 1 0 0 0 0 0 0
 0 1 0 0 0 1 1 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 1 0 0
 0 0 0 0 1 1 1 1 0 1 0 1 0 0 1 1 0 1 0 1 0 0 0 1 0 0 1 0 1 1 1 0 0 1 1 0 1
 1 1 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0]
```

```
y_random_model=model.predict(X_test)
print(y_random_model)
```

```
[1 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 1 1 0 0 1 1 0 0 0 1 1 0 0 0 0 1 0 0 1 0 1
 1 1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 1 1 0 0 1
 0 0 1 1 0 0 0 0 1 1 1 1 0 0 1 1 1 0 0 0 0 1 0 1 0 1 1 0 1 1 0 0 1 1 1 1 1
 0 1 1 1 1 0 1 1 0 0 0 1 0 1 0 0 1 0 1 0 0 1 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0
 1 1 1 0 1 1 1 1 1 0 0 1 1 0 1 1 1 0 0 1 0 1 1 0 1 0 0 1 1 1 1 1 0 1 0 1 1
 1 0 0 1 0 0 0 1 0 1 0 1 0 1 1 1 0 0 1 0 1 1 0 1 0 0 1 1 0 1 1 0 1 1 1 1 0
 0 1 1 0 0 0 1 0 0 0 0 0 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 0 1 1 0 0 1 0
 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0 0 0 1 0 0 1 0 1 0 0 0 1 0 1 1 1 0 0 0 0 1 0
 0 1 0 1 1 1 1 0 1 0 0 1 1 1 0 1 1 0 0 0 1 0 0 0 1 1 1 1 0 1 0 1 0 1 1 1 0
 1 1 0 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 1 0 1 0 0 1 0 0 0 1 0 1 0 0 1 1
 1 0 0 1 0 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 1 0 1 1 1 0 1 1 1 0 1 1 1 1 1 1 0
 1 0 0 0 0 0 1 0 0 1 1 0 0 1 0 1 0 0 1 1 1 1 0 0 1 1 0 1 1 0 0 0 0 1 1 0 1
 1 0 1 0 1 1 1 1 0 1 1 0 1 0 0 1 1 0 1 1 0 0 0 1 1 0 0 1 0 0 0 1 0 0 1 1 1
 1 1 0 1 0 1 0 0 0 0 1 1 0 1 0 1 1 0 0 1 1 1 1 1 0 1 0 1 1 0 0 1 0 1 0 1 1
 0 0 1 0 1 0 0 0 0 0 0 1 1 1 0 1 0 0 1 0 0 0 1 1 1 1 1 1 0 1 1 0 1 0 0 1 1
 0 1 1 1 0 0 0 1 1 1 0 1 0 0 0 0 0 0 0 0 1 0 1 1 0 1 1 1 1 0 0 1 0 0 0 1 0
 0 0 1 1 0 1 1 0 1 1 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 1 0 0 1 0 1 0 0 0
 0 1 1 1 0 1 1 1 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 1 1 0 0 1 0 1 0 0
 0 0 1 0 0 1 0 0 1 1 1 0 0 0 1 1 1 1 0 0 0 1 0 1 1 0 1 1 0 1 1 0 0 0 0 0 0
 0 1 0 0 0 1 1 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 1 0 0
 0 0 0 0 1 1 1 1 0 1 0 1 0 0 1 1 0 1 0 1 0 0 0 1 0 0 1 0 1 1 1 0 0 1 1 0 1
 1 1 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0]
```

```
#accuracy score, classification report, classifier matrix
from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
print("Accuracy Score",accuracy_score(y_test,y_pred))
print("Classification Report",classification_report(y_test,y_pred))
print("Confusion Matrix",confusion_matrix(y_test,y_pred))
```

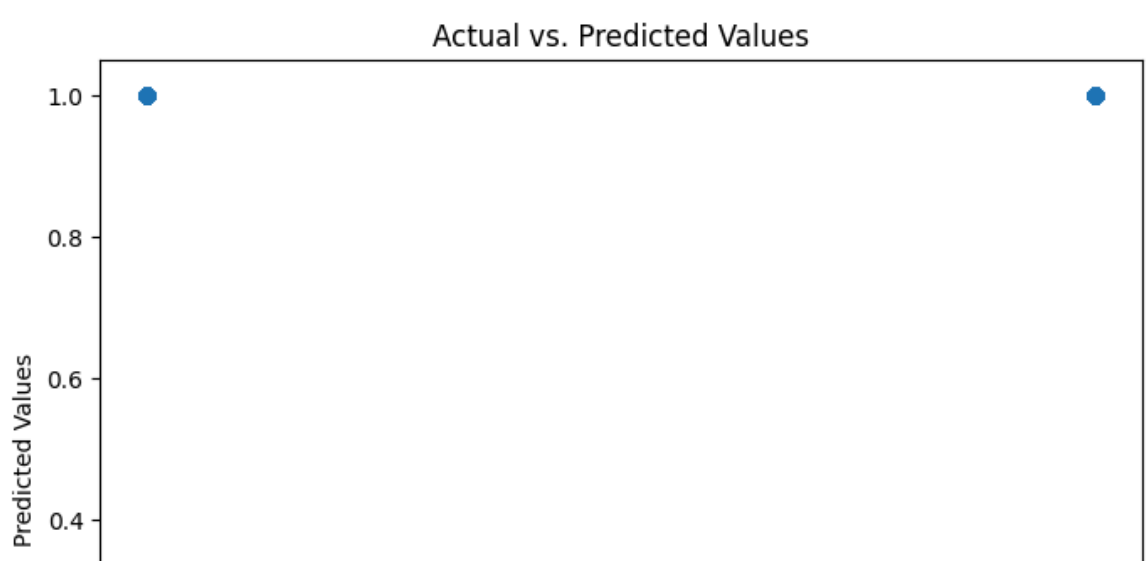
```
Accuracy Score 0.51625
Classification Report               precision    recall  f1-score   support

           0       0.53      0.56      0.54       411
           1       0.50      0.47      0.49       389

    accuracy                           0.52       800
   macro avg       0.52      0.52      0.51       800
weighted avg       0.52      0.52      0.52       800

Confusion Matrix [[229 182]
 [205 184]]
```

```
#accuracy score, classification report, classifier matrix
from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
print("Accuracy Score",accuracy_score(y_test,y_random_model))
print("Classification Report",classification_report(y_test,y_random_model))
print("Confusion Matrix",confusion_matrix(y_test,y_random_model))
```

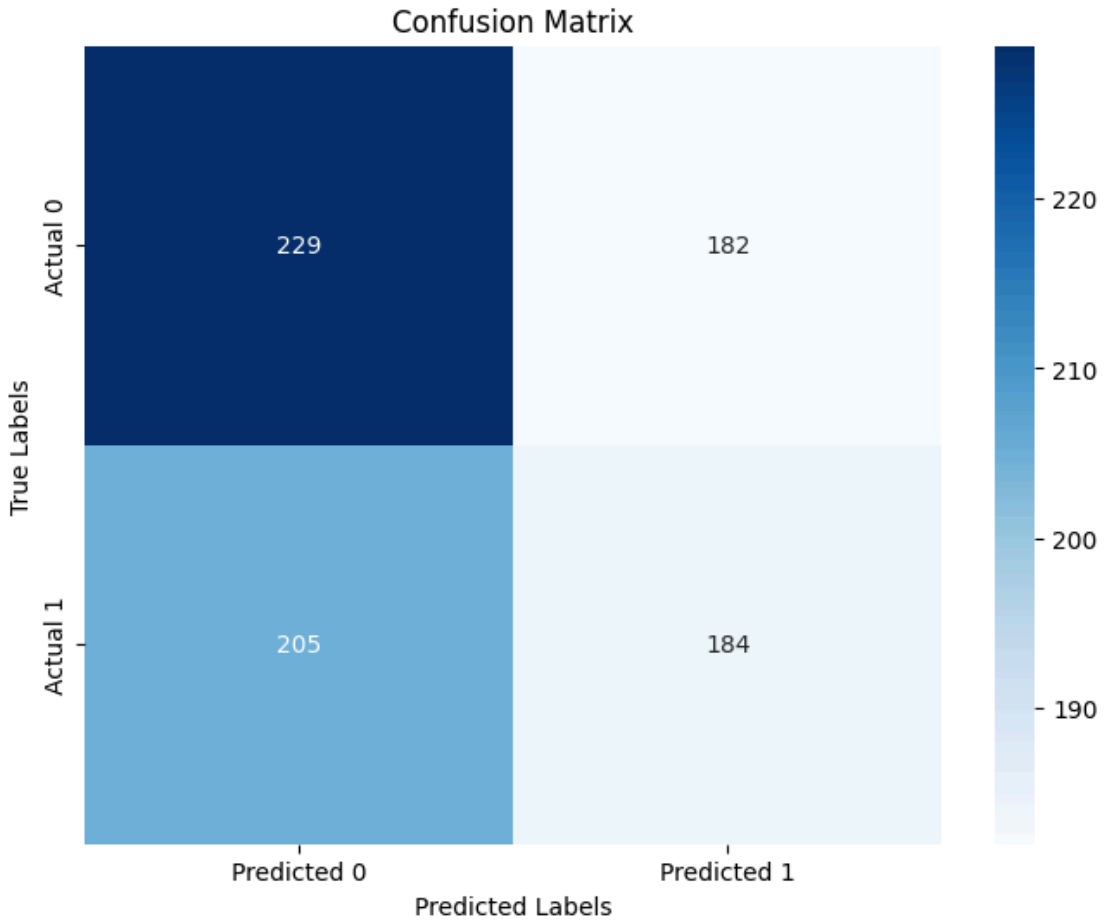
```
Accuracy Score 0.51625
Classification Report               precision    recall  f1-score   support

           0       0.53      0.56      0.54       411
           1       0.50      0.47      0.49       389

    accuracy                           0.52       800
   macro avg       0.52      0.52      0.51       800
weighted avg       0.52      0.52      0.52       800
```

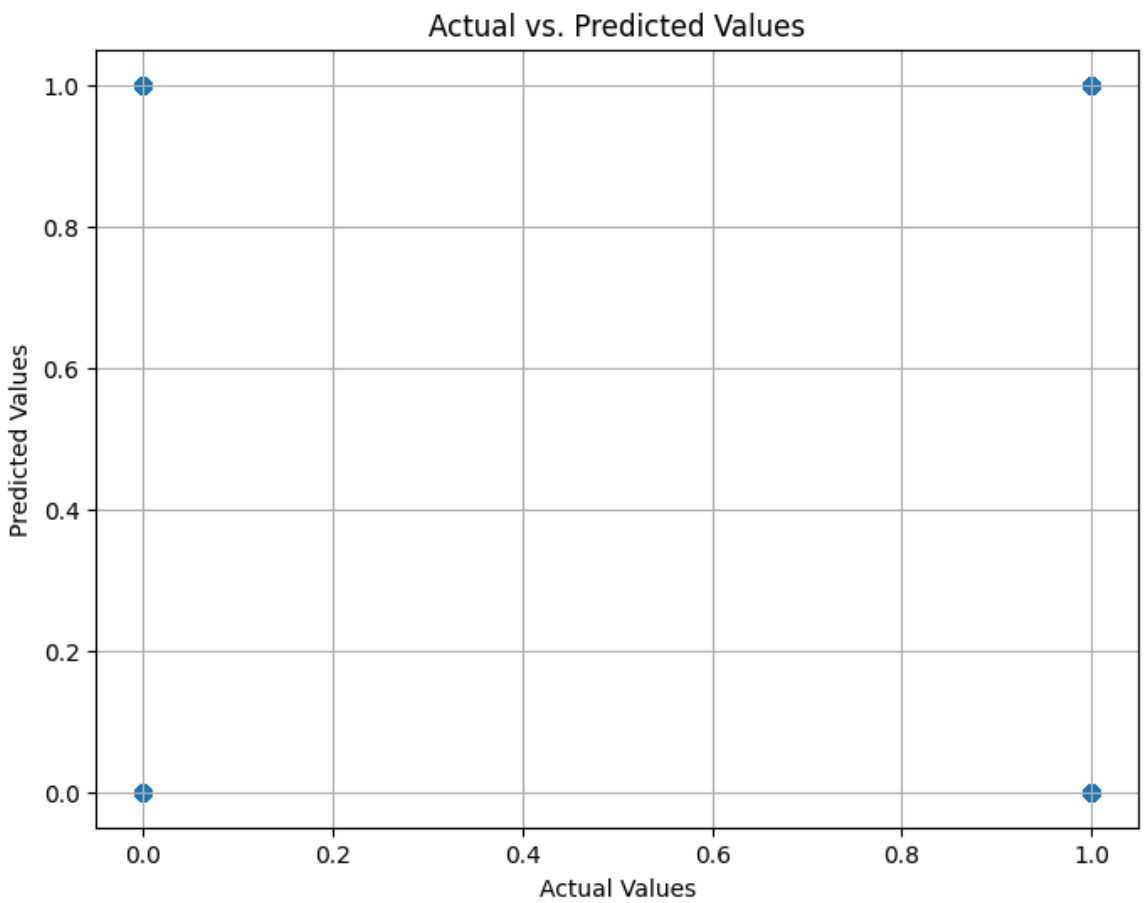
```
Confusion Matrix [[229 182]
 [205 184]]
```

```
plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred)
plt.xlabel("Actual Values")
plt.ylabel("Predicted Values")
plt.title("Actual vs. Predicted Values")
plt.show()
```

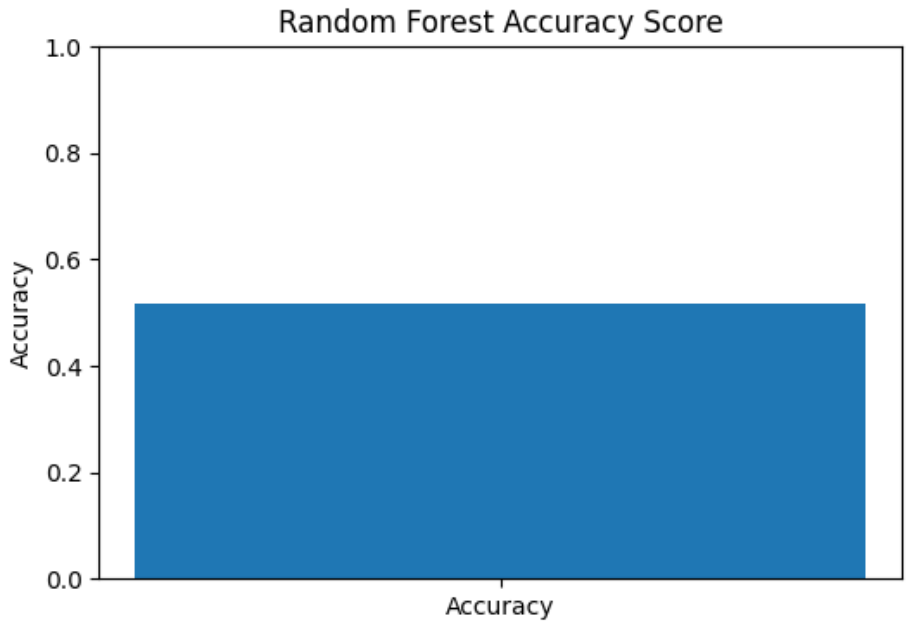



```
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix
import seaborn as sns

# Assuming y_test and y_pred are already defined from your model evaluation

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",
            xticklabels=["Predicted 0", "Predicted 1"],
            yticklabels=["Actual 0", "Actual 1"])
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title("Confusion Matrix")
plt.show()


# Actual vs Predicted Values chart
plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred, alpha=0.5)  # Added alpha for better visibility with overlapping point
plt.xlabel("Actual Values")
plt.ylabel("Predicted Values")
plt.title("Actual vs. Predicted Values")
plt.grid(True) # Added grid for better readability
plt.show()
```

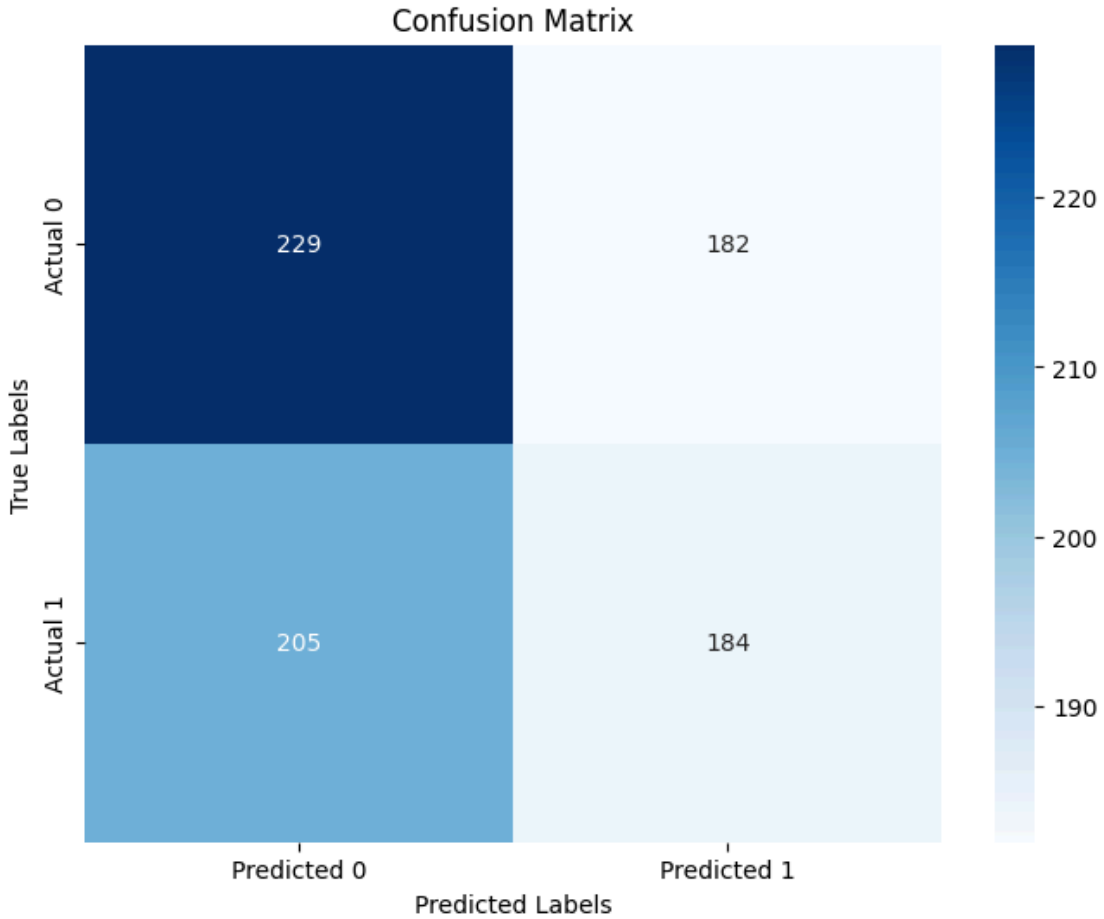

Confusion Matrix
229
182
205
184
Actual 0
Actual 1
True Labels
Predicted 0
Predicted 1
Predicted Labels
220
210
200
190


Actual vs. Predicted Values
1.0
0.8
0.6
0.4
0.2
0.0
Predicted Values
0.0
0.2
0.4
0.6
0.8
1.0
Actual Values

```
# Accuracy Score
accuracy = accuracy_score(y_test, y_random_model)
plt.figure(figsize=(6, 4))
plt.bar(['Accuracy'], [accuracy])
plt.ylim(0, 1)  # Set y-axis limits for better visualization
plt.ylabel('Accuracy')
plt.title('Random Forest Accuracy Score')
plt.show()

# Classification Report
print(classification_report(y_test, y_random_model)) #printing classification report

# Confusion Matrix
cm = confusion_matrix(y_test, y_random_model)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",
            xticklabels=["Predicted 0", "Predicted 1"],
            yticklabels=["Actual 0", "Actual 1"])
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title("Confusion Matrix")
plt.show()
```




```
              precision    recall  f1-score   support

           0       0.53      0.56      0.54       411
           1       0.50      0.47      0.49       389

    accuracy                           0.52       800
   macro avg       0.52      0.52      0.51       800
weighted avg       0.52      0.52      0.52       800
```



```
# Assuming 'y_test', 'y_pred' (from Logistic Regression), and 'y_random_model' (from RandomForest)

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

# --- Logistic Regression Evaluation ---
print("Logistic Regression Evaluation:")
print("Accuracy Score:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))

cm_logistic = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(6, 4))
sns.heatmap(cm_logistic, annot=True, fmt="d", cmap="Blues",
            xticklabels=["Predicted 0", "Predicted 1"],
            yticklabels=["Actual 0", "Actual 1"])
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title("Logistic Regression Confusion Matrix")
plt.show()

# --- Random Forest Evaluation ---
print("\nRandom Forest Evaluation:")
print("Accuracy Score:", accuracy_score(y_test, y_random_model))
print("Classification Report:\n", classification_report(y_test, y_random_model))


cm_random_forest = confusion_matrix(y_test, y_random_model)
plt.figure(figsize=(6, 4))
sns.heatmap(cm_random_forest, annot=True, fmt="d", cmap="Blues",
            xticklabels=["Predicted 0", "Predicted 1"],
            yticklabels=["Actual 0", "Actual 1"])
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title("Random Forest Confusion Matrix")
plt.show()


# --- Comparison ---
print("\nModel Comparison:")
print(f"Logistic Regression Accuracy: {accuracy_score(y_test, y_pred):.4f}")
print(f"Random Forest Accuracy: {accuracy_score(y_test, y_random_model):.4f}")
```