Customer sentiment analysis result

1)Best model for Customer sentiment analysis

The best model Is support vector classifier where Accuracy is.84, Bias .99 and variance is .84 In this model I included stop words during cleaning the text and Used TFIDF vectorization

```
   ...: print("\nAll Accuracy Scores:")
   ...: for model_name, accuracy in accuracy_scores.items():
   ...:     print(f"{model_name}: {accuracy:.4f}")
Model Rankings based on Accuracy Score:
Rank 1: Support Vector Machine - Accuracy: 0.8400
Rank 2: Logistic Regression - Accuracy: 0.8300
Rank 3: Random Forest - Accuracy: 0.7950
Rank 4: K-Nearest Neighbors - Accuracy: 0.7650
Rank 5: Naive Bayes - Accuracy: 0.7150
Rank 6: Decision Tree - Accuracy: 0.7000

All Accuracy Scores:
Random Forest: 0.7950
Logistic Regression: 0.8300
Support Vector Machine: 0.8400
K-Nearest Neighbors: 0.7650
Decision Tree: 0.7000
Naive Bayes: 0.7150
```

```
In [9]: SVM=SVC()
   ...: SVM.fit(X_train, y_train)
Out[9]: SVC()

In [10]:
    ...: y_pred = SVM.predict(X_test)

In [11]:
    ...: from sklearn.metrics import confusion_matrix
    ...: cm = confusion_matrix(y_test, y_pred)
    ...: print(cm)
[[87 10]
 [22 81]]

In [12]: from sklearn.metrics import accuracy_score
    ...: ac = accuracy_score(y_test, y_pred)
    ...: print(ac)
0.84

In [13]: bias = SVM.score(X_train,y_train)
    ...: bias
Out[13]: 0.99875

In [14]: variance = SVM.score(X_test,y_test)
    ...: variance
Out[14]: 0.84
```

2)The second best performing model is Logistic regression without stopword and with TFIDF vectorization . the accuracy is .75, bias .94 and Variance .75

```
    ...:     print(f"{model_name}: {accuracy:.4f}")
Model Rankings based on Accuracy Score:
Rank 1: Logistic Regression - Accuracy: 0.7550
Rank 2: Support Vector Machine - Accuracy: 0.7550
Rank 3: Random Forest - Accuracy: 0.7350
Rank 4: Naive Bayes - Accuracy: 0.7200
Rank 5: Decision Tree - Accuracy: 0.7100
Rank 6: K-Nearest Neighbors - Accuracy: 0.6800

All Accuracy Scores:
Random Forest: 0.7350
Logistic Regression: 0.7550
Support Vector Machine: 0.7550
K-Nearest Neighbors: 0.6800
Decision Tree: 0.7100
Naive Bayes: 0.7200

    ...: LG=LogisticRegression(penalty='l2', C=1.0, solver='liblinear')
    ...: LG.fit(X_train, y_train)
Out[11]: LogisticRegression(solver='liblinear')

In [12]:
    ...: y_pred = LG.predict(X_test)

In [13]:
    ...: from sklearn.metrics import confusion_matrix
    ...: cm = confusion_matrix(y_test, y_pred)
    ...: print(cm)
[[85 12]
 [37 66]]

In [14]: from sklearn.metrics import accuracy_score
    ...: ac = accuracy_score(y_test, y_pred)
    ...: print(ac)
0.755

In [15]: bias = LG.score(X_train,y_train)
    ...: bias
Out[15]: 0.9475

In [16]: variance = LG.score(X_test,y_test)
    ...: variance
Out[16]: 0.755
```

The third performing model is Support vector classifier with BOW and stop words, model accuracy is .73, bias is .95 and variance .73

```
0.73

In [2]: bias = SVM.score(X_train,y_train)
   ...: bias
Out[2]: 0.95625

In [3]: variance = SVM.score(X_test,y_test)
   ...: variance
Out[3]: 0.73

In [4]:
```

```
In [12]: runcell(0, '/Users/myyntiimac/untitled2.py')
Model Rankings based on Accuracy Score:
Rank 1: Support Vector Machine - Accuracy: 0.7300
Rank 2: Naive Bayes - Accuracy: 0.7300
Rank 3: Logistic Regression - Accuracy: 0.7100
Rank 4: Random Forest - Accuracy: 0.7050
Rank 5: Decision Tree - Accuracy: 0.6750
Rank 6: K-Nearest Neighbors - Accuracy: 0.6300

All Accuracy Scores:
Random Forest: 0.7050
Logistic Regression: 0.7100
Support Vector Machine: 0.7300
K-Nearest Neighbors: 0.6300
Decision Tree: 0.6750
Naive Bayes: 0.7300
```