جامعة الملك سعود
King Saud University
كلية علوم الحاسب والمعلومات
قسم هندسة البرمجيات

King Saud University

College of Computer and Information SciencesDepartment

of Software Engineering

**SWE 486 – Cloud Computing and Big Data**


# Sentiment Analysis

**PHASE 2**

---



| # | NAME | ID |
|---|---|---|
| 1 | Lina Alkhodhiri | 441201109 |
| 2 | Noura Alsaif | 439200832 |
| 3 | Alanoud Alomar | 441201113 |

| | |
|---|---|
| **GROUP #** | 4 |
| **SUPERVISOR** | L. Hailah Almazrua |



***Submission Date:*** *August 14, 2022*

## Table of contents



## Table of figures

# Phase 2 Model planning and building

## 1. Descriptive Analytics

Descriptive Analysis is the type of data that helps describe, show, or summarize data points in a constructive way such that patterns might emerge that fulfill every condition of the data. It is one of the most important steps for conducting statistical data analysis.

- **import libraries**

```
import pandas as pd
import numpy as np
```

*Figure 1. import libraries*

- **Shape and columns of the data frame**

```
[131] data_df.shape

      (1524, 5)

[133] data_df.columns

      Index(['Text', 'classification', 'clean text', 'tokens', 'text length'], dtype='object')
```

*Figure 2. data frame columns and shape*

The shape of the data frame has 1524 rows and 5 columns.

- **data frame summary**

```
[132] data_df.info(verbose = True)

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1524 entries, 0 to 1523
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Text            1524 non-null   object
 1   classification  1519 non-null   object
 2   clean text      1524 non-null   object
 3   tokens          1524 non-null   object
 4   text length     1524 non-null   int64
dtypes: int64(1), object(4)
memory usage: 59.7+ KB
```

*Figure 3. data frame summary*

.info() function is used by pandas to show a brief summary about a DataFrame including the index dtype and columns, non-null values, and memory usage.

- **Statistics**

```
data_df.describe()
```

| | text length |
|---|---|
| count | 1524.000000 |
| mean | 1900.550525 |
| std | 1150.112428 |
| min | 140.000000 |
| 25% | 1120.000000 |
| 50% | 1588.000000 |
| 75% | 2239.250000 |
| max | 13684.000000 |

*Figure 4. view statistical details*

In figure 4 the result after we used describe() method which is helping to view some basic statistical details like percentile, mean, std, min, max of the data frame.

- **Word occurrences**

We had to read a data frame to count the idf for getting term frequency.

```
[160] #libraries for word occurrence()
      from sklearn.feature_extraction.text  import TfidfTransformer
      from sklearn.feature_extraction.text  import CountVectorizer

[167] #instantiate CountVectorizer()
      countVec = CountVectorizer()

      #generate word counts for the words
      word_count_vector = countVec.fit_transform(data_df['Text'].astype('U'))
      word_count_vector.shape

      (1524, 72745)

[168] #Transform a count matrix to a normalized tf-idf representation
      tfidf_transformer = TfidfTransformer(smooth_idf=True,use_idf=True)
      #idf values
      tfidf_transformer.fit(word_count_vector)

      TfidfTransformer()

[169] #print idf values
      df_idf = pd.DataFrame(tfidf_transformer.idf_, index = countVec.get_feature_names(),columns = ["idf_weights"])
```

*Figure 5. import sklearn libraries to count the idf*

- **Most frequent terms**

```
[170] #most frequent terms
      df_idf.sort_values(by=["idf_weights"]).head(10)
```

| | idf_weights |
|---|---|
| من | 1.007900 |
| الفيلم | 1.166577 |
| على | 1.226428 |
| في | 1.265846 |
| عن | 1.277894 |
| فيلم | 1.355271 |
| كان | 1.394379 |
| لا | 1.440158 |
| هذا | 1.451423 |
| ما | 1.457622 |

*Figure 6. most frequent terms using sort by idf*

- **Least frequent terms**

```
[171] #least frequent terms
      df_idf.sort_values(by=["idf_weights"]).tail(10)
```

| | idf_weights |
|---|---|
| جافيير | 7.636603 |
| جافييه | 7.636603 |
| جاكوين | 7.636603 |
| جاكي | 7.636603 |
| جالا | 7.636603 |
| جالاكسي | 7.636603 |
| جالسا | 7.636603 |
| جالسين | 7.636603 |
| جاعلا | 7.636603 |
| لإيقانه | 7.636603 |

*Figure 7. least frequent terms using sort by idf*

## 2. Predictive Analytics

In our project, we chose the Naïve Bayes model and used several techniques to evaluate the results. We will discuss the steps of our implementation of the Naïve Bayes but before that, we will mention the advantages of the model and the definition of the technologies that we used:

### • Naïve Bayes Model and its features

Naïve Bayes algorithm is a supervised learning algorithm used for solving classification problems and it is mainly used in text classification that includes a high-dimensional training dataset. Naïve Bayes Classifier is one of the simplest and most effective Classification algorithms which helps in building fast machine learning models that can make quick predictions. Some popular examples of Naïve Bayes Algorithm are spam filtration, classifying articles, or in our case Sentimental analysis [2].

### • ROC Curve

An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. It basically shows two parameters: true positive rate and false positive rate. And by the looks of it, we can determine the best performance by observing how the curve is plotted as demonstrated in the figure below [3].

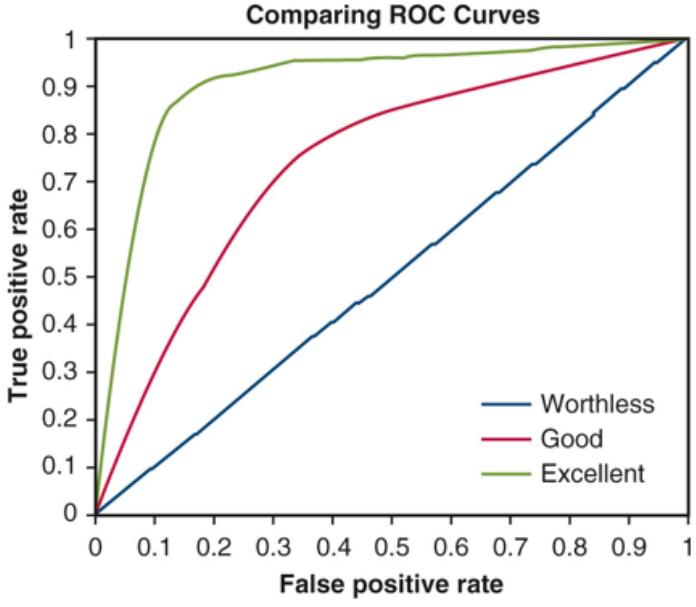


*Figure 8. ROC Curve*

## • Ten-Folds Cross Validation

The basic idea of ten folds cross validation, as the name implies, is to divide the data set into 10 folds fires. Second, we will iterate 10 times over the data, each time holding one-fold for testing and the 9 remaining folds will be used for training. And so on until the 10 folds had all been used for testing once and for training 9 times. The accuracy will be the average accuracy of all 10 iterations. Which makes ten folds cross validation results in skill estimates that generally have a lower bias than other methods.

## • Confusion Matrix

The confusion matrix provides many helpful evaluation measurements. Precision, Recall, …etc. are all derived from the confusion matrix. Which essentially provides counts of true positive, false positive, true negative and false negative predictions as illustrated in figure 11. Precision - Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. Recall (Sensitivity) - Recall is the ratio of correctly predicted positive observations to all observations in actual class.

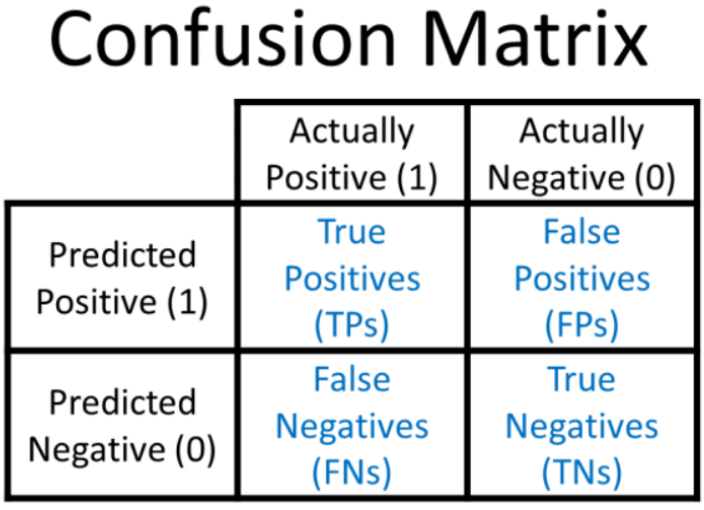


*Figure 9. Confusion Matrix*

## 2.1 Naïve Bayes Model

### • Libraries used

**NumPy:** It is a Python library that provides a multidimensional array object, various derived objects, and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more [4].

**Sklearn:** Is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering, and dimensionality reduction via a consistency interface in Python [5].

**Pandas:** Pandas is an open-source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays [6].

**Matplotlib:** Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy [7].

### • steps of our implementation

1- We have removed the neutral rows since the neutral class is incomprehensible and does not relate to a specific and clear class.

```
# remove the "Neutral" class
data_df=data_df[data_df['classification'] != "Neutral"]
```

*Figure 10. Remove neutral class*

2- To process each text's class label, we converted the values from "Positive" class to (1) and the values that correspond to "Negative" class to (0) as figure 9 shows.

```
# change values to numeric
data_df['classification'] = data_df['classification'].map({'Positive':1, 'Negative':0})
```

```
data_df.head(5)
```

| | Text | classification | clean text | text length | tokens |
|---|---|---|---|---|---|
| | تلك الايام " مغامرة سينمائية متميزة إنتظرت عرض فيلم تلك الايام ' كثيراً ، خاصة لانه ماخوذ عن روايه للاديب الكبير فتحى غانم و تحمل نفس الاسم الجذاب والتي تعتبر واحدة من أهم اعماله، وعندما عرفت أن من سيقوم باخراج الفيلم هو "أحمد غانم " ابن الروائى فتحى غانم ، شعرت بأنها شجاعة كبيرة منه خصوصاً أن هذا العمل هو أولى تجاربه السينمائية وهو عمل ليس سهلاً ، فتحويل عمل ادبى إلى فيلم سينمائى يحتاج لخرج متمرس ، لذا كنت متخوف بعض الشيء من مستوى الاخراج ولكن وجود الفنان محمود حميدة فى الفيلم جعلنى أتوقع إنه سيكون فيلم جيد . ولم يخيب حميدة توقعاتى ، فعند مشاهدة الفيلم ادركت إننى أمام عمل سينمائى متميز ومتخلف كثيراً بداية من مخرج خالف توقعاتى فقدم صورة متميزة للغاية ، ونجح فى التعبير عن الحرب النفسية الموجودة بين الابطال وانفسهم وذلك من خلال الاضاءة وزوايا التصوير التى برع فى استخدامها ، وكانت أروع المشاهد فى الفيلم هى المطاردات الخارجية التى يقوم بها احمد الفيشاوى فكانت على اعلى مستوى سواء من ناحية التمثيل او التصوير او الاخراج. تدور قصة الفيلم حول شخصية سالم عبيد "محمود حميدة" وهو استاذ جامعي ومفكر سياسي وهو شخص من الممكن ان يفعل اى شيء ليرضى النظام وذلك طمعاً فى المنصب السياسى الكبير ، وهو أيضاً نفس الشخص الذى تزوج فتاه لا تحبه وذلك طمعاً فى الحب الذى شاهدها وهو تعطيه إلى حبيبها السابق الذى توفى فى حادث إرهابى ، وهو أيضاً الشخص الذى ممن الممكن ان يفعل أى شيء للوصول إلى أهدافه ، ويجب أن اشير أنه كان من | 1 | تلك الايام مغامره سينمائيه متميزه انتظرت عرض فيلم تلك الايام كثيراً خاصه لانه ماخوذ عن روايه للاديب الكبير فتحي غانم و تحمل نفس الاسم الجذاب والتي تعتبر واحده من اهم اعماله وعندما عرفت ان من سيقوم باخراج الفيلم هو احمد غانم ابن الروائي فتحي غانم شعرت بانها شجاعه كبيره منه خصوصاً أن هذا العمل هو اولي تجاربه السينمائيه وهو عمل ليس سهلاً فتحويل عمل ادبي الي فيلم سينمائي يحتاج لخرج متمرس لذا كنت متخوف بعض الشيء من مستوي الاخراج ولكن وجود الفنان محمود حميده في الفيلم جعلني اتوقع انه سيكون فيلم جيد ولم يخيب حميده توقعاتي فعند مشاهده الفيلم ادركت انني امام عمل سينمائي متميز ومتخلف كثيراً بدايه من مخرج خالف توقعاتي فقدم صوره متميزه للغايه ونجح في التعبير عن الحرب النفسيه الموجوده بين الابطال وانفسهم وذلك من خلال الاضاءه وزوايا التصوير التي برع في استخدامها وكانت اروع المشاهد في الفيلم هي المطاردات الخارجيه التي يقوم بها احمد الفيشاوي فكانت علي اعلي مستوي سواء من ناحيه التمثيل او التصوير او الاخراج تدور قصه الفيلم حول شخصيه سالم عبيد محمود حميده وهو استاذ جامعي ومفكر سياسي وهو شخص من الممكن ان يفعل اي شيء ليرضي النظام وذلك طمعاً في المنصب السياسي الكبير وهو ايضاً نفس الشخص الذي تزوج فتاه لا تحبه وذلك طمعاً في الحب الذي شاهدها وهو تعطيه الي حبيبها السابق الذي توفي في حادث ارهابي وهو ايضاً الشخص الذي ممن الممكن ان يفعل اي شيء للوصول الي اهدافه ويجب ان اشير انه كان من | 1368 | [تلك, الايام, مغامره, سينمائيه, متميزه, انتظرت, عرض, فيلم, تلك, الايام, كثيرا, خاصه, لانه, ماخوذ, عن, روايه, للاديب, الكبير, فتحي, غانم, و, تحمل, نفس, الاسم, الجذاب, والتي, تعتبر, واحده, من, اهم, اعماله, وعندما, عرفت, ان, من, سيقوم, باخراج, الفيلم, هو, |

Figure 11. Convert value to numeric

3- The next step was identifying the column that will contain the texts we want to predict their sentiment and the column that will contain the classification class label as figure 10 shows.

```
# idneitfy the data and the labels
data= data_df['clean text']
target= data_df['classification']
```

Figure 12. Identifying the column

4- As figure 11 shows, we used TF-IDF vectorizer to extract the features from our data, and the vectorizer discovered 60467 words out of the 1368 rows of the given data.

```
1  # Use TfidfVectorizer for feature extraction (TFIDF to convert textual data to numeric form):
2  tf_vec = TfidfVectorizer()
3  X = tf_vec.fit_transform(data)
4  X.shape
```

```
(1368, 60467)
```

Figure 13. TF-IDF vectorizer

5- After that we plugged in our X array which contains the words after transforming them using TFIDF vectorizer, and our target array which contains the class labels, and we set up the test sample size to be 50% which leaves 50% for training as figure 12 shows.

```
# Training Phase
X_train, X_test, y_train, y_test = train_test_split(X, target, test_size=0.50, random_state=0)
```

```
print("Training: ", X_train.shape, y_train.shape)
print("Testing: ", X_test.shape, y_test.shape)
```

```
Training:  (684, 60467) (684,)
Testing:  (684, 60467) (684,)
```

*Figure 14. Training phase*

6- Finally, we plugged in our training and testing data to Multinominal Naïve Bayesian classifier function and got an accuracy of 70% as figure 13 shows.

```
# create the classifer and fit the training data and lables
classifier_nb = MultinomialNB().fit(X_train.todense(),y_train)

print("MultinomialNB accuracy: %.2f"%classifier_nb.score(X_test.todense(), y_test))
print('_'*100)

#do a 10 fold cross-validation
results_nb = cross_val_score(classifier_nb, X.todense(),target, cv=10)
print("\n10-fold cross-validation:")
print(results_nb)

print("The average accuracy of the MultinomialNB classifier is : %.2f" % np.mean(results_nb))
print('_'*100)

print("\nConfusion matrix of the MultinomialNB classifier:")
predicted_nb = classifier_nb.predict(X_test.todense())
print(confusion_matrix(y_test,predicted_nb))
print('_'*100)

print("\nClassification_report of MultinomialNB classifier:")
print(classification_report(y_test,predicted_nb))
print('_'*100)
```

```
MultinomialNB accuracy: 0.70
____________________________________________________________________________________________________
10-fold cross-validation:
[0.69343066 0.69343066 0.69343066 0.69343066 0.69343066 0.69343066
 0.69343066 0.69343066 0.69852941 0.69852941]
The average accuracy of the MultinomialNB classifier is : 0.69
____________________________________________________________________________________________________
Confusion matrix of the MultinomialNB classifier:
[[  0 202]
 [  0 482]]
____________________________________________________________________________________________________
Classification_report of MultinomialNB classifier:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       202
           1       0.70      1.00      0.83       482

    accuracy                           0.70       684
   macro avg       0.35      0.50      0.41       684
weighted avg       0.50      0.70      0.58       684
```

*Figure 15. Classification report*

As figure 15 shows, we plotted the True Positive Rate (TPR) and False Positive Rate (FPR) using the ROC Curve.

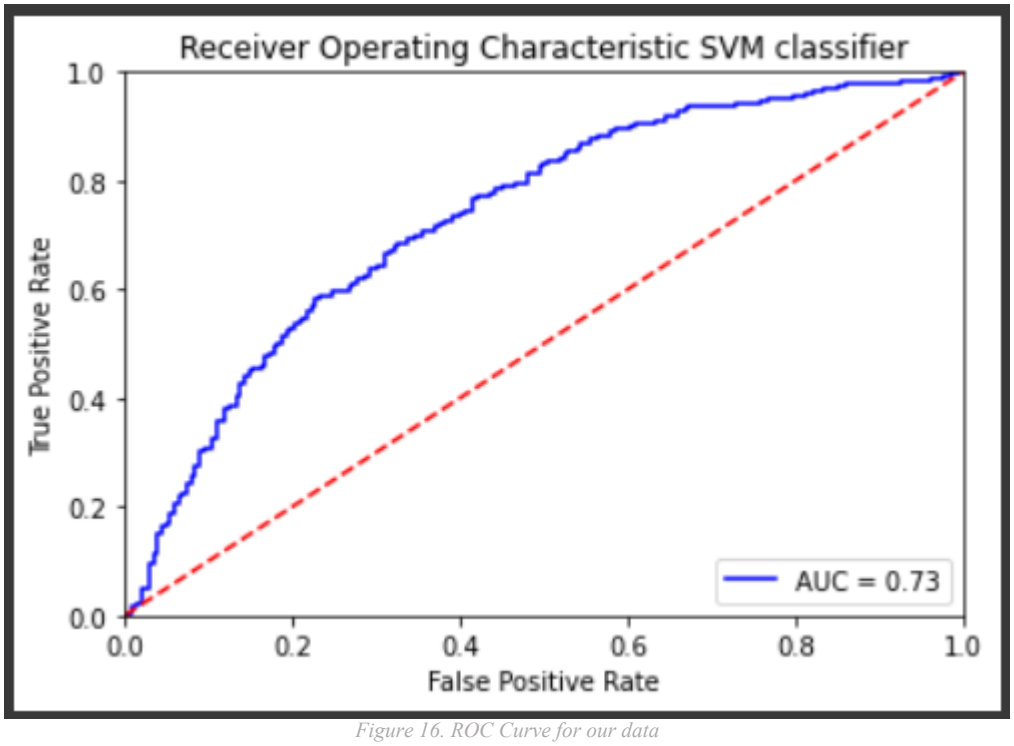


*Figure 16. ROC Curve for our data*

# 3. Discussion questions

### 3.1 Does the model appear valid and accurate on the test data?

When looking at the results and comparing them with the evaluation techniques, we see that there is a discrepancy in the unbalanced data, so it does not seem valid, although the accuracy result is good.

### 3.2 Does the model output/behavior make sense to the domain experts?

We learned that it is not possible for the model to record a result of 100%, and although the model did not show any unexpected behavior or provide any unusual result, but the output was not very compelling.

### 3.3 Do the parameter values make sense in the context of the domain?

It can be expected that the model may not record very high results, especially in sentiment analysis data because it is rather complex, so we believe that there is a hidden problem regardless of the results and the scope on which it is.

### 3.4 Is the model sufficiently accurate to meet the goals?

Based on the outputs of our model, we see that the results are good and cover our goals, bearing in mind that we analyze people's feelings towards films, and that does not require very high accuracy, so the accuracy of our model is considered good, which is 70%.

### 3.5 Are more data or inputs needed?

Yes, when the data increases, the accuracy increases, and the model outputs will become better. In addition, we see that there is a discrepancy in the data, so a balance must be established between the data to see a much better result for our model.

# 4. Reference

[1] Rawat, A., 2022. *What is Descriptive Analysis?- Types and Advantages | Analytics Steps*. [online] Analyticssteps.com. Available at: <https://www.analyticssteps.com/blogs/overview-descriptive-analysis> [Accessed 8 August 2022].

[2] https://www.javatpoint.com/machine-learning-naive-bayes-classifier

[3] https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

[4] https://numpy.org/doc/stable/user/whatisnumpy.html

[5] https://www.tutorialspoint.com/scikit_learn/scikit_learn_introduction.htm

[6] https://www.activestate.com/resources/quick-reads/what-is-pandas-in-python-everything-you-need-to-know/

[7] https://www.activestate.com/resources/quick-reads/what-is-matplotlib-in-python-how-to-use-it-for-plotting/