كلية الهندسة بشبرا

FACULTY OF ENGINEERING AT SHOUBRA

Course: CCE 538 (Advanced Topics in Computer Engineering)
Term: 231

SENTIMENT ANALYSIS

Group 1

Date handed in: / 12 / 2022

S#	Student Name	Edu Email	Student ID	Mark	
				Report & Slides (30)	Implementat (50)
1	Nada Mahmoud	Nada181369@feng.bu.edu.eg	201901978		
2	Ammar Ahmed	ammar181288@feng.bu.edu.eg	202902055		
3	Fady Zarif	fady18573@feng.bu.edu.eg	201901979		
4	Abdulrahman Essayed	abdulrahman211902100@feng.bu.edu.eg	201901995		
5	Sabah Mohammed	mailto:sabah170409@feng.bu.edu.eg	201901995		

Supervised by:

Dr. lamiaa Elrefaei

PROPLEM FORMULATION

Sentiment analysis is the process of analyzing pieces of writing to determine the emotional tone they carry, it is considered one of NLP techniques where your classifier is asked to determine if a phrase or group of words has a positive, negative, or neutral attitude. To keep the classification issues a binary result, the third property is occasionally ignored. In our project, we are aiming to use different classification algorithms to determine the sentiment of text and predict the positive and negative reviews.

This is used in organizations for a variety of applications, including:

- Identifying brand awareness, reputation, and popularity at a specific moment or over time.
- Tracking consumer reception of new products or features.
- Evaluating the success of a marketing campaign.
- Pinpointing the target audience or demographics.
- Collecting customer feedback from social media, websites, or online forms.
- Conducting market research.

Overall, it's a great way for businesses to connect and understand what their customers want.

Dataset

We will use the Internet Movie Database (IMDb) dataset consisting of nearly 50K movie reviews which is considered to have a substantial amount of information for natural language processing purposes (NLP). The IMDb dataset classified into positive and negative movie review which is also known as binary sentiment analysis dataset.

We will also use Restaurant Reviews dataset consisting of 1000 records

TECHNICAL DISCUSSION

Libraries used

pandas is a python library used for analyzing, cleaning, exploring, and manipulating data.[1]

NumPy is a Python library used for working with arrays. It provides an array object that is faster than traditional Python lists. The library contains many mathematical, algebraic, and transformation functions.[2]

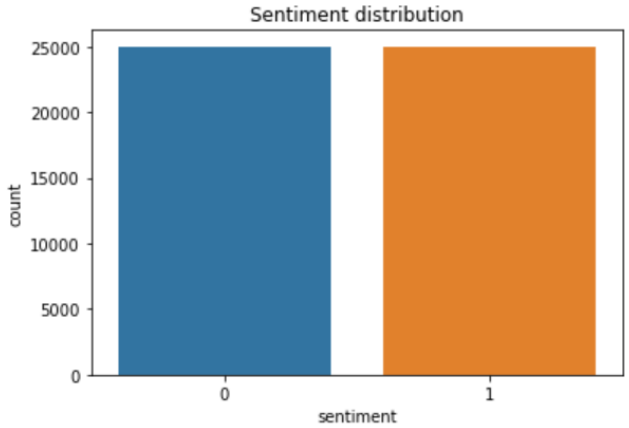
Matplotlib-pyplot Matplotlib is a python library used to create 2D graphs and plots by using python scripts.

Regular expressions This module offers a set of functions that allows us to search a string for a match can be used to check if a string contains the specified search pattern[3]

NLTK is a leading platform for building Python programs to work with human language data. It provides many text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning. [4]

Preprocessing

The dataset we are working with consists of 50k labeled movie reviews to visualize the data we used a count plot to show the amount of positive (1) to negative (0) reviews.



An Imbalanced dataset pose a challenge for predictive modeling as most of the machine learning algorithms used for classification were

designed around the assumption of an equal
number of examples for each class. This results in models that have poor predictive performance, specifically for the minority class. Fortunately As shown above this dataset has an equal ratio of positive to negative reviews so we considered it balanced.

After that we checked for any duplicate records and found 418 duplicates, so we dropped them.

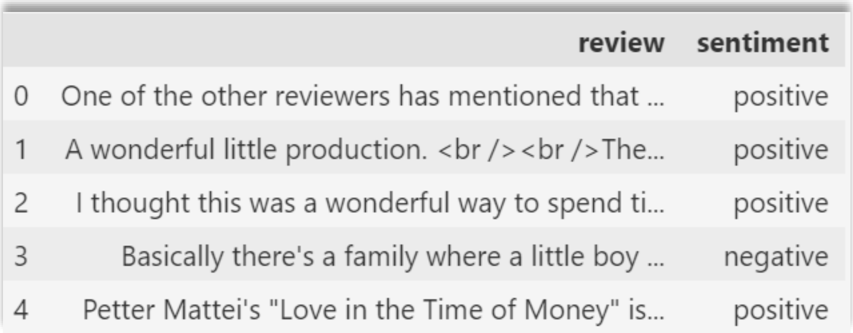
We created a text processing function that will clean the reviews from URLs, hashtags, and unnecessary characters like punctuation or markup. Using the Regular expression library to leave only the text. Next, we tokenize the reviews. [5]

Tokenization is a way of separating a piece of text into smaller units called tokens. Here, tokens can be either words, characters, or sub-words. Word Tokenization is the most used tokenization algorithm. It splits a piece of text into individual words based on a certain delimiter. We used the NLTK-tokenize built in library [6].

Stop words are the words which are generally filtered out before processing a textual data. These are the most common words in any language and does not add much information to the text. Examples of a few stop words in English are “the”, “a”, “an”, “so”, “what”. Stop words are available in abundance in any human language. By removing these words, we remove the low-level information from our text to give more focus to the important information[7]. We imported the English stop words from the NLTK corpus, we filtered them out from the reviews.

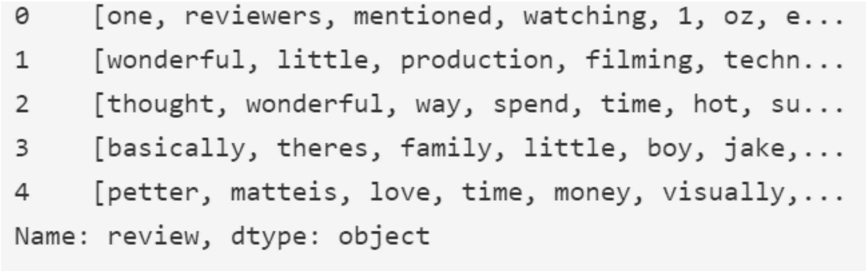
Stemming is the process of reducing a word to its stem. This is important since the English language has several variants of a single term. The presence of these variances in a text corpus results in data redundancy when developing machine learning models. Such models may be ineffective. To build a robust model, it is essential to normalize text by removing repetition and transforming words to their base form through stemming. their are different types of stemming algorithms like LancasterStemmer, Porter’s Stemmer and Snowball Stemmer algorithms

Porter’s Stemmer is one of the most popular stemming methods proposed in 1980. It is based on the idea that the suffixes in the English language are made up of a combination of smaller and simpler suffixes. This stemmer is known for its speed and simplicity, which we used in our processing



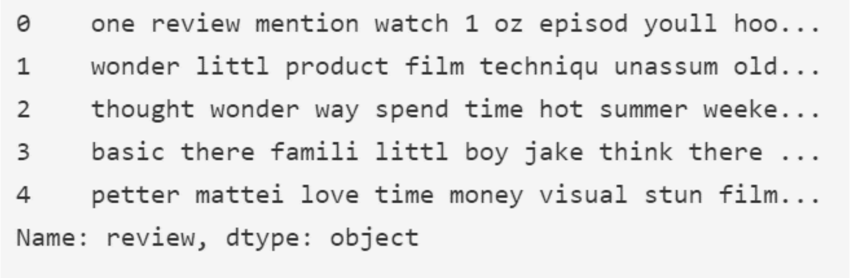
	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. <br /><br />The...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive

Fig. The first 5 records before any preprocessing



```
0    [one, reviewers, mentioned, watching, 1, oz, e...
1    [wonderful, little, production, filming, techn...
2    [thought, wonderful, way, spend, time, hot, su...
3    [basically, theres, family, little, boy, jake,...
4    [petter, matteis, love, time, money, visually,...
Name: review, dtype: object
```

Fig. The first 5 records after tokenization and the removal of stop words



```
0    one review mention watch 1 oz episod youll hoo...
1    wonder littl product film techniqu unassum old...
2    thought wonder way spend time hot summer weeke...
3    basic there famili littl boy jake think there ...
4    petter mattei love time money visual stun film...
Name: review, dtype: object
```

Fig. The first 5 records stemming

We decided to take 2 different approaches when dealing with the dataset. We will apply the same preprocessing mentioned above for both but for the final form or the features that will be the input of the classification model first we will try handling each review as a whole sentence, then we will try handling the reviews by extracting key words features.

For the **first** approach we take the preprocessed data and apply the TF-IDF vectorizer with maximum features set to 200. For the **second** approach we split the dataset into positive and negative reviews and used a counter to count the number of times each word was used in regards of negative and positive reviews to help identify the keywords then applied the TF-IDF.

TF-IDF stands for Term frequency-inverse document frequency. It is a text vectorizer that transforms text into a vector form. It is a combination of term frequency and inverse document frequency. Term frequency is defined as the number of times a word *i* appears in a document *j* divided by the total number of words in the document.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$$

Inverse document frequency refers to the log of the total number of documents divided by the number of documents that contain the word. The logarithm is added to dampen the importance of a very high value of IDF.

$$idf(w) = log(\frac{N}{df_t})$$

The full equation:

$$w_{i,j} = tf_{i,j} \times \log(\frac{N}{df_i})$$

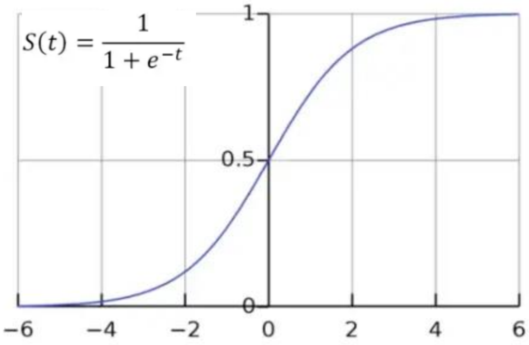
$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

TFIDF is works on the logic that if a particular word is a very high occurrence or very low occurrence, in both cases, that word is not significant for finding any meaningful information. So, the Higher value of TFIDF depicts the higher significance of the words while lower values represent lower significance.

Classification models

Logistic regression

Logistic Regression is a classification technique used in machine learning. it’s used to predict the probability of a target variable. In natural language processing, logistic regression is the baseline supervised machine learning algorithm for classification. To perform logistic regression, the sigmoid function, presented below with its plot.



The equation for logistic regression is as follows where g is the sigmoid function and z is the input features [8]

$$f_{\vec{w},b}(\vec{x}) = g(\underbrace{\vec{w}\cdot\vec{x}+b}_{z}) = \frac{1}{1+e^{-(\vec{w}\cdot\vec{x}+b)}}$$

Naive Bayes classification

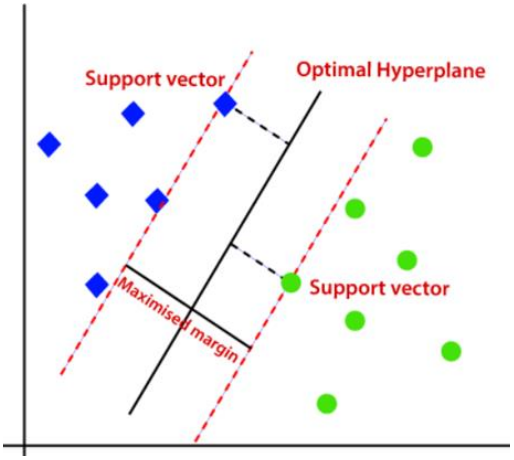
It is a classification technique based on Bayes’ Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

$$P(c\,|\,d) = \frac{\overset{\text{Likelihood}}{P(d\,|\,c)}\,\overset{\text{Prior}}{P(c)}}{\underset{\text{Normalization Constant}}{P(d)}}$$

P (c | d) is the posterior probability of class (c, sentiment) given a document d. Naive Bayes classification when applied to a text classification problem, it is referred to as "Multinomial Naive Bayes" classification.[9]

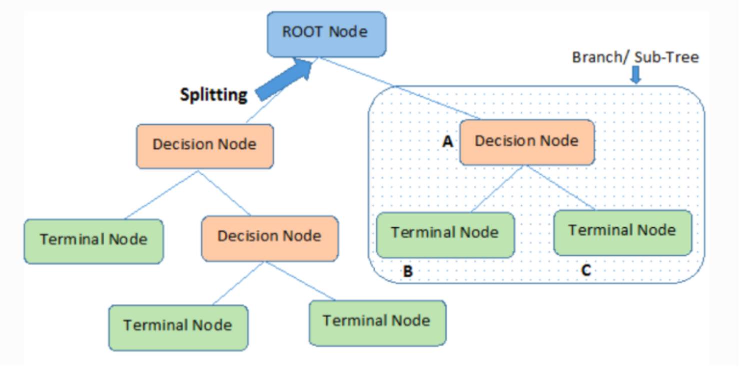
Support Vector Machine

SVM is a supervised machine learning algorithm that can be used for both classification and regression challenges. The objective of SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points. SVMs are different from other classification algorithms because of the way they choose the decision boundary that maximizes the distance from the nearest data points of all the classes. The decision boundary created by SVMs is called the maximum margin classifier.[10]



Decision Tree Classifier

Decision Tree is a supervised learning algorithm. They can be used for solving regression and classification problems. The goal of using a Decision Tree is to create a training model that can use to predict the class of the target by learning simple decision rules. They are Relatively inexpensive to construct, extremely fast at classifying unknown records and Easy to interpret for small-sized trees.[11]



Evaluation techniques

classification report

It is one of the performance evaluation metrics of a classification-based machine learning models. It displays the model's precision, recall, F1 score and support.

Precision is the ability of a model to find all the relevant cases within a data set. Precision is defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall is the ability of a model to find all the relevant cases within a data set. Recall is defined as follows:

$$\text{Recall} = \frac{TP}{TP + FN}$$

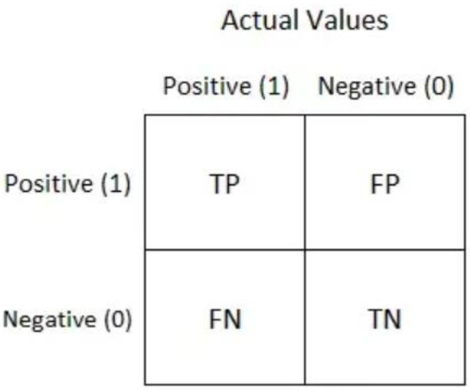
F1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. F1 scores are lower than accuracy measures as they embed precision and recall into their computation. F1 is defined as follows:

$$F1\ Score = 2 \times \frac{recall \times precision}{recall + precision}$$

Accuracy is one metric for evaluating classification models. accuracy is the fraction of predictions the model got right. accuracy has the following definition:[12]

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

Confusion matrix is a performance measurement for machine learning classification problem where output can be two or more classes. It is a table with 4 different combinations of predicted and actual values [13]

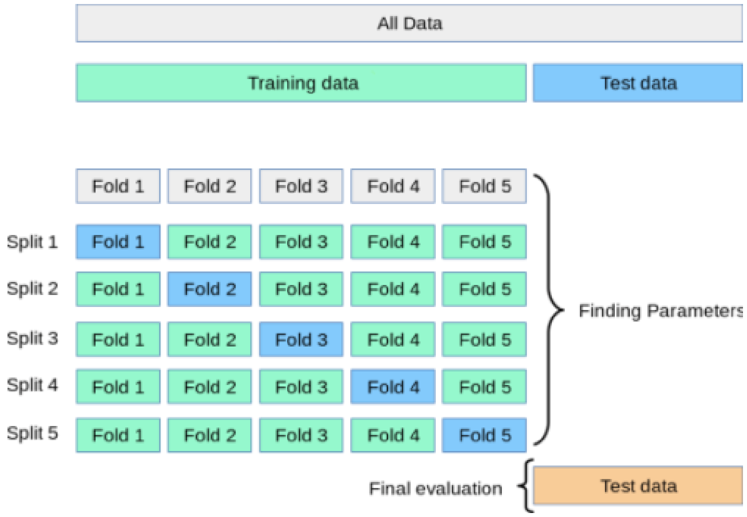


	Actual Values	
	Positive (1)	Negative (0)
Positive (1)	TP	FP
Negative (0)	FN	TN

Cross validation

Cross-validation is a technique in which we train our model using the subset of the dataset and then evaluate using the complementary subset of the dataset.

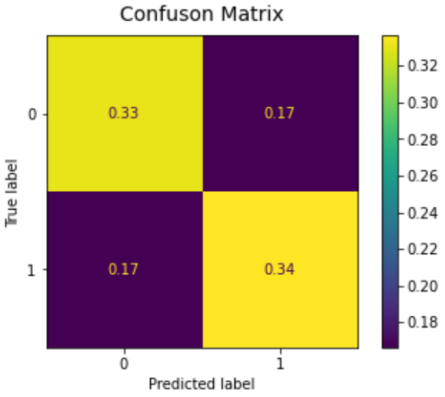
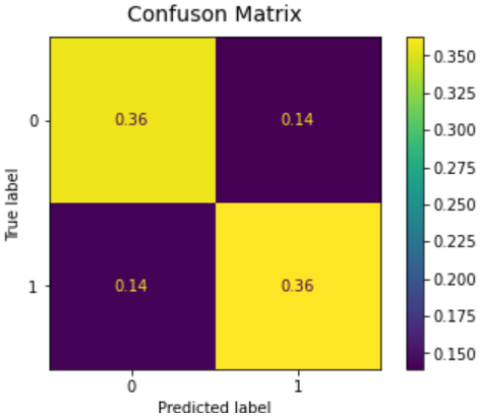
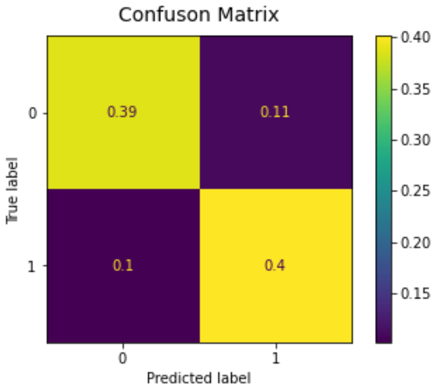
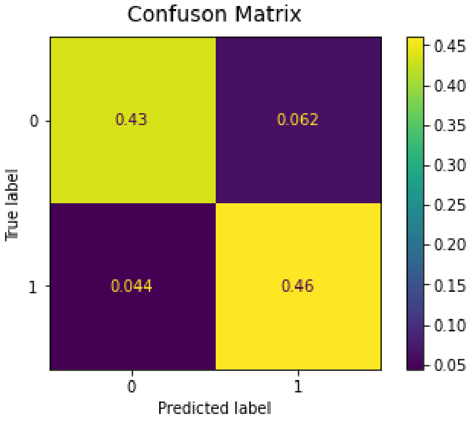
In the k-folds cross validation we split the dataset into k number of subsets (known as folds) then we perform training on the all the subsets but leave one(k-1) subset for the evaluation of the trained model. In this method, we iterate k times with a different subset reserved for testing purpose each time.

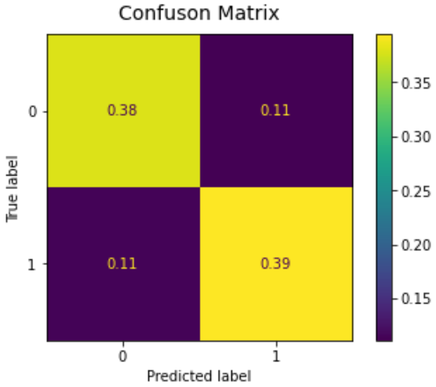
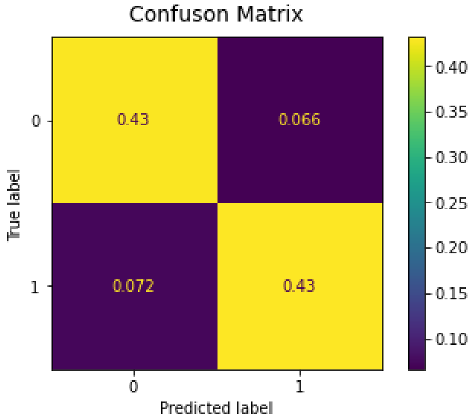
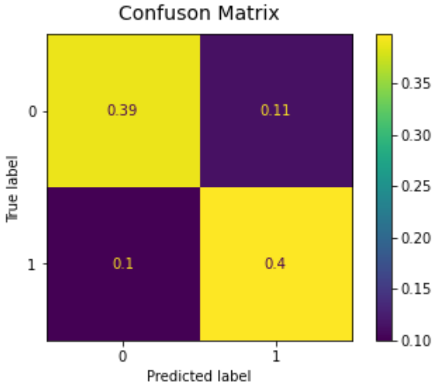
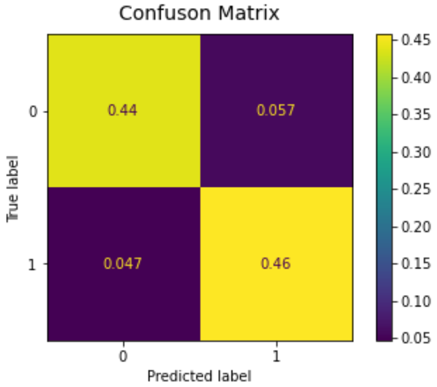


For validating our models, we picked a k of 7.

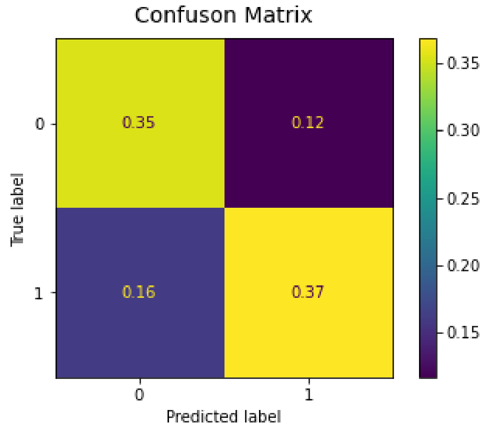
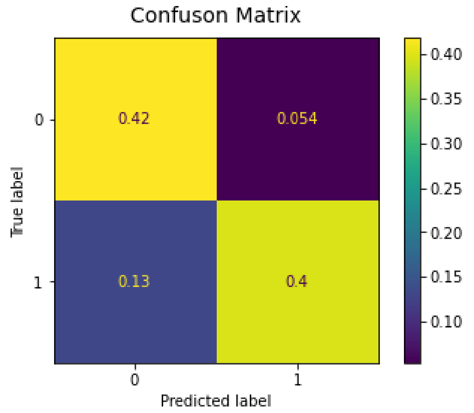
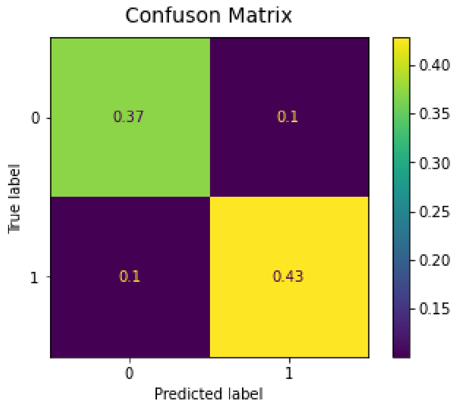
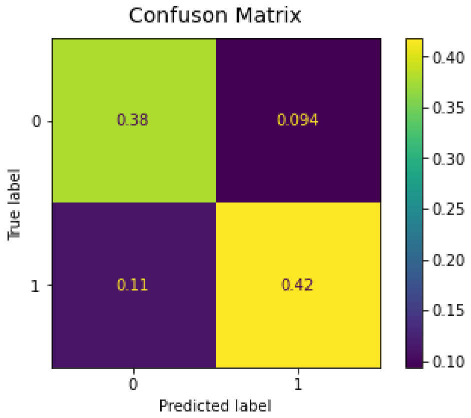
Before applying any of the previously discussed classification models, we split the data into a training set and testing set of size 20%

RESULTS

	Sentence approach	***Word approach***
1st classifier ***DESSION TREE***	Test Accuracy: 66% For both classes: Precision: 0.67 Recall: 0.67 F1 score: 0.67  Cross-validation score:0.66	Test Accuracy: 71% For both classes: Precision: 0.72 Recall: 0.72 F1 score: 0.72  Cross-validation score:0.70
2nd classifier ***Logistic Regression***	Test Accuracy: 78% For both class 0 (Negative): Precision: 0.79 Recall: 0.78 F1 score: 0.79 For class 1 (positive): Precision: 0.78 Recall: 0.80 F1 score: 0.79  Cross-validation score:0.79	Test Accuracy: 89% For both class 0 (Negative): Precision: 0.91 Recall: 0.88 F1 score: 0.89 For class 1 (positive): Precision: 0.88 Recall: 0.91 F1 score: 0.90  Cross-validation score:0.88

3rd classifier ***Multinomial Naive Bayes***	Test Accuracy: 77% For both class 0 (Negative): Precision: 0.77 Recall: 0.77 F1 score: 0.77 For class 1 (positive): Precision: 0.78 Recall: 0.77 F1 score: 0.78  Cross-validation score:0.76	Test Accuracy: 86% For both class 0 (Negative): Precision: 0.86 Recall: 0.87 F1 score: 0.86 For class 1 (positive): Precision: 0.87 Recall: 0.86 F1 score: 0.86  Cross-validation score:0.86
4th classifier ***Linear Support Vector Machine***	Test Accuracy: 79% For both class 0 (Negative): Precision: 0.80 Recall: 0.77 F1 score: 0.78 For class 1 (positive): Precision: 0.78 Recall: 0.80 F1 score: 0.79  Cross-validation score:0.78	Test Accuracy: 89% For both class 0 (Negative): Precision: 0.90 Recall: 0.88 F1 score: 0.89 For class 1 (positive): Precision: 0.89 Recall: 0.91 F1 score: 0.90  Cross-validation score:0.89

Restaurant Reviews

1st classifier ***DESSION TREE***	Test Accuracy: 72% <u>For both class 0 (Negative):</u> Precision: 0.69 Recall: 0.75 F1 score: 0.72 <u>For class 1 (positive):</u> Precision: 0.77 Recall: 0.70 F1 score: 0.73 Cross-validation score:0.73 
2nd classifier ***Logistic Regression***	Test Accuracy: 81% <u>For both class 0 (Negative):</u> Precision: 0.76 Recall: 0.89 F1 score: 0.82 <u>For class 1 (positive):</u> Precision: 0.88 Recall: 0.75 F1 score: 0.81 Cross-validation score:0.77 
3rd classifier ***Multinomial Naive Bayes***	Test Accuracy: 80% <u>For both class 0 (Negative):</u> Precision: 0.79 Recall: 0.79 F1 score: 0.79 <u>For class 1 (positive):</u> Precision: 0.81 Recall: 0.81 F1 score: 0.81 Cross-validation score:0.76 
4th classifier ***Linear Support Vector Machine***	Test Accuracy: 79% <u>For both class 0 (Negative):</u> Precision: 0.77 Recall: 0.80 F1 score: 0.79 <u>For class 1 (positive):</u> Precision: 0.82 Recall: 0.79 F1 score: 0.80 Cross-validation score:0.77 

DISSCUSION OF RESULTS

Comparing the results from the first data set we found that the word approach performed better over all the classification models.

Both the linear SVM and the logistic regression had an accuracy of 89% which is the highest accuracy from all our applied models across the datasets

The second dataset had lower scores compared to the word approach in the first dataset, this could be because it is significantly smaller in size.

TASK ASSIGNMENT

Preprocessing D1	AMMAR
Word approach D1	ABDELRAHMAN
Sentence approach D1	NADA
Logistic regression Decision tree D1	FADY
Naïve bayes Linear SVM D1	SABAH
Preprocessing D2	NADA, SABAH
Logistic regression Decision tree D2	ABDELRAHMAN
Naïve bayes Linear SVM D2	AMMAR, FADY