ANALYSIS OF REVIEWS OF MOVIES USING ML

[1] Ravi Bolleddula [2]G.Triveni , [3]K.Sreedevi, [4]K.Sambhavi ,[5]B.Sirisha
[1]Guide [2,3,4] U.G. Scholar
[1,2,3,4] Ravindra College of Engineering for Women
Email : triveni9494@gmail.com, sreedevi6231@gmail.com, sambhavikonnipati02@gmail.com, bangisirisha@gmail.com

ABSTRACT

To understand text analytics and natural language processing, we need to understand what makes a language "natural". In simple terms, a natural language is a language developed and evolved by humans through natural use and communication rather than constructing and creating the language artificially, like a computer programming language. Various human languages, such as English, Japanese, or Sanskrit, can be called natural languages. Natural languages can be communicated in different ways, including speech, writing, or even using signs. There has been a lot of interest in trying to understand the origins, nature, and philosophy of language.

I. INTRODUCTION

The nature of meaning in a language is concerned with the semantics of a language and the nature of meaning itself. Here, philosophers of language or linguistics try to find out what it means to actually "mean" anything, i.e., how the meaning of any word or sentence came into being and how different words in a language can be synonyms of each other and form relations. Another thing of importance here is how structure and syntax in the language paved the way for semantics or, to be more specific, how words that have their own meaning are structured together to form meaningful sentences. Linguistics is the scientific study of language, a special field that deals with some of these problems.

Syntax, semantics, grammar, and parse trees are some ways to solve these problems. The nature of meaning can be expressed in linguistics between two human beings, notably a sender and a receiver. From a non-linguistic standpoint, things like body language, prior experiences, and psychological effects are contributors to the meaning of language, where each human being perceives or infers meaning in their own way, taking into account some of these factors. The use of language is more concerned with how language is used as an entity in various

scenarios and communication between human beings. This includes analyzing speech and the usage of language when speaking, including the speaker's intent, tone, content, and actions involved in expressing a message.

This is often called a "speech act" in linguistics. More advanced concepts like language creation and human cognitive activities like language acquisition—which study the learning and usage of languages— are also of prime interest. Language cognition specifically focuses on how the cognitive functions of the human brain are responsible for understanding and interpreting language. Considering the example of a typical sender and receiver, there are many actions involved, from message communication to interpretation. Cognition tries to find out how the mind combines and relates specific words into sentences and then into a meaningful message and what is the relation of language is to the thought process of the sender and receiver when they use the language to communicate messages. The relationship between language and reality explores the extent of truth of expressions originating from language. Language philosophers try to measure how factual these expressions are and how they relate to certain affairs in our world which are true. This relationship can be expressed in several ways and we explore some of them.

One of the most popular models is the "triangle of reference," which is used to explain how words convey meaning and ideas in the minds of the receiver and how that meaning relates back to a real-world entity or fact. The triangle of reference was proposed by Charles Ogden and Ivor Richards in their book, The Meaning of Meaning, and is denoted in the below Figure 1.

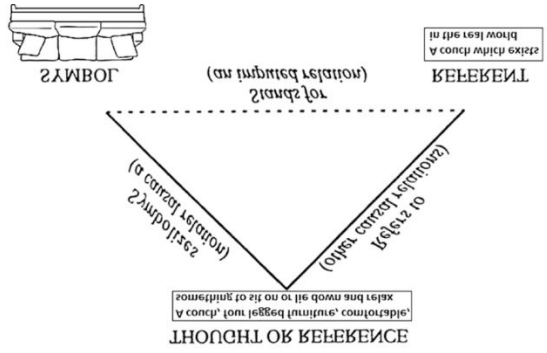

Figure 1. The triangle of reference model

The triangle of reference model is also known as the meaning of meaning model. Figure 1-1 shows a real example of a couch being perceived by a person. A symbol is denoted as a linguistic symbol like a word or an object that evokes thought in a person's mind. In this case, the symbol is the couch and this evokes thoughts like what is a couch, a piece of furniture that can be used for sitting on or lying down and relaxing, something that gives us comfort. These thoughts are known as a reference and through this

reference, the person is able to relate it to something that exists in the real world, which is called a referent. In this case, the referent is the couch that the person perceives to be present in front of him.

The second way to determine relationships between language and reality is known as the "direction of fit" and we talk about two main directions here. The "word-to-world" direction of fit talks about instances, where the usage of language can reflect reality. This indicates using words to match or relate to something that's happening or has already happened in the real world. An example would be the sentence, "The Eiffel Tower is really big," which accentuates a fact in reality. The other direction of fit is known as "world-to-word" and talks about instances where the usage of language can change reality. An example here would be the sentence, "I am going to take a swim," where you are changing reality by taking a swim and are representing this fact in the sentence you are communicating. Figure 2 shows the relationship between both directions of fits.

Based on the referent that is perceived from the real world, a person can form a representation in the form of a symbol or word and consequently can communicate the same to another person. This forms a representation of the real world based on the received symbol, thus forming a cycle.

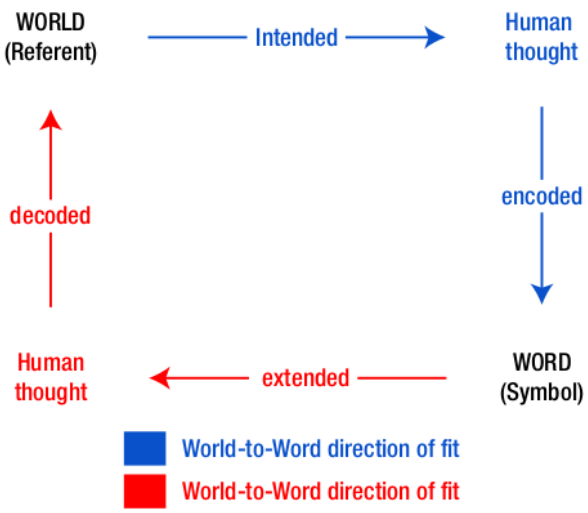

Figure 2. The direction of fit representation.

Linguistics:

linguistics is defined as the scientific study of language, including the form and syntax of language, the meaning and semantics depicted by the usage of language, and the context of use. The origins of linguistics can be dated back to the 4th century when Indian scholar and linguist Panini formalized the Sanskrit language description. The term linguistics was first defined to indicate the scientific study of languages in 1847 approximately before which the term philology was used to indicate the same. While a detailed exploration of linguistics is not needed for text analytics, it is useful to know the different areas of linguistics because some of them are used extensively in natural language processing and text analytics algorithms. The main distinctive areas of study under linguistics are mentioned next.

- **Phonetics:** This is the study of the acoustic properties of sounds produced by the human vocal tract during a speech. This includes studying the sound properties of how they are created as well as perceived by human beings. The smallest individual unit of human speech is termed a phoneme, which is usually distinctive to a specific language as opposed to a more generic term, called a phone.
- **Phonology:** This is the study of sound patterns as interpreted in the human mind and used for distinguishing between different phonemes. The structure, combination, and interpretations of phonemes are studied in detail, usually by taking into account a specific language at a time. The English language consists of around 45 phonemes. Phonology usually extends beyond just studying phonemes and includes things like accents, tone, and syllable structures.
- **Syntax:** This is usually the study of sentences, phrases, words, and their structures. This includes researching how words are combined grammatically to form phrases and sentences. Syntactic order of words used in a phrase or a sentence matter since the order can change the meaning entirely.
- **Semantics:** This involves the study of meaning in language and can be further subdivided into lexical and compositional semantics.
    - **Lexical semantics:** This involves the study of the meanings of words and symbols using morphology and syntax.
    - **Compositional semantics:** This involves studying relationships among words and combinations of words and understanding the meaning of phrases and sentences and how they are related.
- **Morphology:** By definition, a morpheme is the smallest unit of language that has a distinctive meaning. This includes things like words, prefixes, suffixes, and so on, which have their own distinct meaning. Morphology is the study of the structure and meaning of these distinctive units or morphemes in a language. There are specific rules and syntaxes that govern the way morphemes can combine.
- **Lexicon:** This is the study of the properties of words and phrases

used in a language and how they build the vocabulary of the language. These include what kinds of sounds are associated with meanings for words, as well as the parts of speech that words belong to and their morphological forms.

- **Pragmatics:** This is the study of how linguistic and non-linguistic factors like context and scenario might affect the meaning of an expression of a message or an utterance. This includes trying to infer if there are any hidden or indirect meanings in communication.
- **Discourse analysis:** This analyzes language and the exchange of information in the form of sentences across conversations among human beings. These conversations could be spoken, written, or even signed.
- **Stylistics:** This is the study of language with a focus on the style of writing including the tone, accent, dialogue, grammar, and type of voice.
- **Semiotics:** This is the study of signs, symbols, and sign processes and how they communicate meaning. Things like analogies, metaphors, and symbolism are covered in this area.

While these are the main areas of study and research, linguistics is an enormous field and has a much bigger scope. However, things like language syntax and semantics are some of the most important concepts and often form the foundations of natural language processing (NLP).

II. Speech Recognition Systems

This is perhaps the most difficult application for NLP. One of the main and perhaps the most difficult tests of true intelligence in artificial intelligence systems is the Turing test. This test states that if a question is given by the user to the computer and to a human, it would be unable to distinguish the responses obtained. Over a period of time, a lot of progress has been made in this area by using techniques like speech synthesis, analysis, syntactic parsing, and contextual reasoning. However, one chief limitation for speech recognition systems still remains that they are very domain-specific and will not work if the user strays even a little bit from the expected scripted inputs needed by the system. Speech recognition systems are now found in a large variety of places from your computers, to mobile phones, to virtual assistance systems.

III. MODULES

Question Answering Systems

Question Answering Systems (QAS) are built on the principle of question answering based on using techniques from NLP and information retrieval (IR). QAS is primarily concerned with building robust and scalable systems that provide answers to questions given by users in natural language form. Imagine being in a completely different country, asking a question to your personalized assistant in your phone in pure natural language, and getting a similar response from it. This is the ideal state toward which researchers and technologists are working day in and day out. We have achieved some success in this field with personalized assistants like Siri and Cortana, but their scope is still limited since they understand only a subset of key clauses and phrases in the entire human natural language.

To build a successful QAS, you need a huge knowledge base consisting of data about various domains. Efficient querying systems into this knowledge base would be leveraged by the QAS to provide answers to questions in natural language form. Creating and maintaining a queryable vast knowledge base is extremely difficult, hence you will find the rise of QAS in niche domains like food, healthcare, e-commerce, and so on. Chatbots are one of the emerging trends that extensively use QAS.

Contextual Recognition and Resolution

This covers a wide area in understanding natural language, which includes syntactic and semantic based reasoning. Word sense disambiguation is a popular application where we want to find the contextual sense of a word in a given sentence. Consider the word “book”. It can mean an object containing knowledge and information when used as a noun and it can also mean to reserve something like a seat or a table when used as a verb. Detecting these differences in sentences based on context is the main premise of word-sense disambiguation and it is a daunting task. Co-reference resolution is another problem in linguistics that NLP is trying to address. By definition, co-reference is said to occur when two or more terms/expressions in a body of text refer to the same entity. Then they are said to have the same referent.

Consider the example sentence, “John just told me that he is going to the exam hall”. In

this sentence, the pronoun “he” has the referent “John”. Resolving these pronouns is part of co-reference resolution and it becomes challenging once we have multiple referents in a body of text. An example body of text would be, “John just talked with Jim. He told me we have a surprise test tomorrow”. In this body of

text, the pronoun “he” could refer to either “John” or “Jim”, thus making it difficult to pinpoint to the exact referent.

Text Summarization

The main aim of text summarization is to take a corpus of text documents, which could be a collection of texts, paragraphs, or sentences, and reduce the content appropriately to create a summary that retains the key points of the collection of documents. Summarization can be carried out by looking at the various documents and trying to find the keywords, phrases, and sentences that have prominence in the collection. Two main types of techniques for text summarization include extraction-based summarization and abstraction-based summarization. With the advent of huge amounts of text and unstructured data, the need for text summarization for getting to valuable insights quickly is in great demand. Text summarization systems usually perform two main types of operations. The first one is generic summarization, which tries to provide a generic summary of the collection of documents under analysis. The second type of operation is query-based summarization, which provides query-relevant text summaries where the corpus is filtered further based on specific queries and relevant keywords and phrases are extracted relevant to the query and the summary is constructed.

Text Categorization

The main aim of text categorization is to identify to which category or class a specific document should be placed based on the contents of the document. This is one of the most popular applications of NLP and machine learning because with the right data, it is extremely simple to understand the principles behind its internals and implement a working text categorization system. Both supervised and unsupervised machine learning techniques can be used to solve this problem and sometimes a combination of both are used. This has helped build many successful and practical applications, including spam filters and news article categorizations.

IV. Evaluating Classification Models

Training, tuning, and building models are an important part of the whole analytics lifecycle, but it’s even more important to know how well these models are performing. Performance of classification models is usually based on how well they are predicting outcomes for new data points. Usually this performance is measured against a test or holdout dataset,

which consists of data points that were not used to influence or train the classifier in any way. This test dataset has several observations and their corresponding labels. We extract features in the same way as when training the model. These features are fed to the already trained model and we obtain predictions for each data point. These predictions are then matched against the actual labels to see how well or how accurately the model has predicted. There are several metrics to determine a model's prediction performance. We mainly focus on the following metrics.

- Accuracy
- Precision
- Recall
- F1-score

Let's take a classic example of the very popular Wisconsin Diagnostic Breast Cancer dataset. This dataset has 30 attributes or features and a corresponding label for each data point (breast mass) depicting if it has cancer (malignant: label value 1) or no cancer (benign: label value 0). Let's assume we already have this data and will be building a basic logistic regression model and evaluating it. We take the assumption that we have 398 observations in our train dataset and 171 observations in our test dataset. We will be leveraging a nifty module we have created for model evaluation.

Confusion Matrix

A confusion matrix is one of the most popular ways to evaluate a classification model. Although the matrix by itself is not a metric, the matrix representation can be used to define a variety of metrics, all of which become important in some specific case or scenario. A confusion matrix can be created for both a binary classification as well as a multi-class classification model. A confusion matrix is created by comparing the predicted class label of a data point with its actual class label. This comparison is repeated for the whole dataset and the results of this comparison are compiled in a matrix or tabular format. This resultant matrix is our confusion matrix.

		Predicted:	
		0	1
Actual:	0	59	4
	1	2	106

The preceding output depicts the confusion matrix with necessary annotations. We can see that out of 63 observations with label 0 (benign), our model has correctly predicted 59 observations. Similarly, out of 108 observations with label 1 (malignant), our model has correctly predicted 106 observations.

Understanding the Confusion Matrix

While the name itself sounds pretty overwhelming, understanding the confusion matrix is not that confusing once you have the basics right! To reiterate what we learned in the previous section, the confusion matrix is a tabular structure that keeps a track of correct classifications as well as misclassifications. This is useful to evaluate the performance of a classification model for which we know the true data labels and can compare with the predicted data labels. Each column in the confusion matrix represents classified instance counts based on predictions from the model and each row of the matrix represents instance counts based on the actual/true class labels. This structure can also be reversed, i.e. predictions depicted by rows and true labels by columns. In a typical binary classification problem, we usually have a class label that's defined as the positive class, which is basically the class of our interest. For instance, in our breast cancer dataset, we are interested in detecting breast cancer, hence label 1 is our positive class. Figure 6 shows a typical confusion matrix for a binary classification problem, where p denotes the positive class and n denotes the negative class.

Figure 6 should make things more clear with regard to the structure of confusion matrices. In general, we usually have a positive class as we discussed earlier and the other class is the negative class. Based on this structure, we can clearly see four terms of importance.

- True Positive (TP): This is the count of the total number of instances from the positive class where the true class label was equal to the predicted class label, i.e., the total instances where we correctly predicted the positive class label with our model.

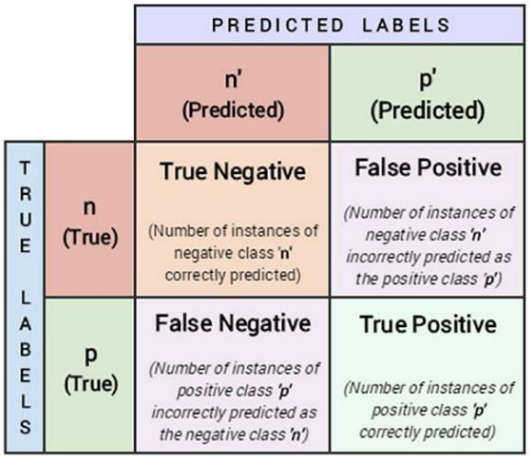

Figure 6. Typical structure of a confusion matrix

- False Positive (FP): This is the count of the total number of instances from the negative class where our model misclassified them by predicting them as positive. Hence, the name, "false" positive.
- True Negative (FN): This is the count of the total number of instances from the negative class, where the true class label was equal to the predicted class label, i.e., the

total instances where we correctly predicted the negative class label with our model.

- False Negative (FN): This is the count of the total number of instances from the positive class where our model misclassified them by predicting them as negative. Hence the name, “false” negative.

Based on this information, can you compute these metrics for our confusion matrix based on the model predictions on the breast cancer test data?

positive_class = 1

TP = 106

FP = 4

TN = 59

FN = 2

V. Performance Metrics

The confusion matrix by itself is not a performance measure for classification models. But it can be used to calculate several metrics that are useful measures for different scenarios. We describe how the major metrics can be calculated from the confusion matrix, compute them manually using necessary formulae, compare the results with functions provided by Scikit-Learn on our predicted results, and give an intuition of scenarios where each of those metric can be used.

Accuracy is one of the most popular measures of classifier performance. It is defined as the overall proportion of correct predictions of the model. The formula for computing accuracy from the confusion matrix is as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Accuracy is normally used when our classes are almost balanced and correct predictions of those classes are equally important.

Precision, also known as a positive predictive value, is another metric that can be derived from the confusion matrix. It is defined as the number of predictions made that are actually correct or relevant out of all the predictions based on the positive class. The formula for precision is as follows:

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

A model with high precision will identify a higher fraction of positive classes as compared to a model with a lower precision. Precision becomes important in cases where we are more concerned about finding the maximum number of positive classes even if the total accuracy reduces.

Recall, also known as sensitivity, is a measure of a model to identify the percentage of relevant data points. It is defined as the number of instances of the positive class that were correctly predicted. This is also known as hit rate, coverage, or sensitivity. The formula for recall is as follows:

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

Recall becomes an important measure of classifier performance when we want to catch the most number of instances of a particular class even when it increases our false positives. For example, consider the case of bank fraud. A model with high recall will give us higher number of potential fraud cases. But it will also help us raise alarm for most of the suspicious cases There are some cases in which we want a balanced optimization of both precision and recall.

The **F1-score** is the harmonic mean of precision and recall and helps us optimize a classifier for balanced precision and recall performance. The formula for the F1-score is as follows:

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

VI. RESULTS

Sentiment Analysis of IMDB Movie Reviews
Problem Statement:

In this, we have to predict the number of positive and negative reviews based on sentiments by using different classification models.

```
print(lr_tfidf_report)
```

```
              precision    recall  f1-score   support

    Positive       0.75      0.75      0.75      4993
    Negative       0.75      0.75      0.75      5007

    accuracy                           0.75     10000
   macro avg       0.75      0.75      0.75     10000
weighted avg       0.75      0.75      0.75     10000

              precision    recall  f1-score   support

    Positive       0.74      0.77      0.75      4993
    Negative       0.76      0.73      0.75      5007

    accuracy                           0.75     10000
   macro avg       0.75      0.75      0.75     10000
weighted avg       0.75      0.75      0.75     10000
```

Confusion matrix

In [25]:

```
1
#confusion matrix for bag of words
2
cm_bow=confusion_matrix(test_sentiments,lr_bow_predict,labels=[1,0])
3
print(cm_bow)
4
#confusion matrix for tfidf features
```

```
cm_tfidf=confusion_matrix(test_sentiments,lr_tfidf_predict,labels=[1,0])
print(cm_tfidf)
```

```
[[3768 1239]
 [1249 3744]]
[[3663 1344]
 [1156 3837]]
```

Stochastic gradient descent or Linear support vector machines for bag of words and tfidf features

In [26]:

```
#training the linear svm
svm=SGDClassifier(loss='hinge',max_iter=500,random_state=42)
#fitting the svm for bag of words
svm_bow=svm.fit(cv_train_reviews,train_sentiments)
print(svm_bow)
#fitting the svm for tfidf features
svm_tfidf=svm.fit(tv_train_reviews,train_sentiments)
print(svm_tfidf)
```

```
C:\Users\bangi\anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  return f(*args, **kwargs)
SGDClassifier(max_iter=500, random_state=42)
C:\Users\bangi\anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  return f(*args, **kwargs)
SGDClassifier(max_iter=500, random_state=42)
```

Model performance on test data

In [27]:

```
#Predicting the model for bag of words
svm_bow_predict=svm.predict(cv_test_reviews)
print(svm_bow_predict)
#Predicting the model for tfidf features
svm_tfidf_predict=svm.predict(tv_test_reviews)
print(svm_tfidf_predict)
```

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

Accuracy of the model

In [28]:

```
#Accuracy score for bag of words
svm_bow_score=accuracy_score(test_sentiments,svm_bow_predict)
print("svm_bow_score :",svm_bow_score)
svm_tfidf_score=accuracy_score(test_sentiments,svm_tfidf_predict)
print("svm_tfidf_score :",svm_tfidf_score)
```

```
svm_bow_score : 0.5829
svm_tfidf_score : 0.5112
```

```
              precision    recall  f1-score   support

    Positive       0.94      0.18      0.30      4993
    Negative       0.55      0.99      0.70      5007

    accuracy                           0.58     10000
   macro avg       0.74      0.58      0.50     10000
weighted avg       0.74      0.58      0.50     10000
```

```
              precision    recall  f1-score   support
```

```
   Positive       1.00      0.02      0.04      499
3
   Negative       0.51      1.00      0.67       50
07

    accuracy                           0.51     10000
   macro avg       0.75      0.51      0.36     10
000
weighted avg       0.75      0.51      0.36     1
0000
```

Plot the confusion matrix

In [30]:

```
[[4948   59]
 [4112  881]]
[[5007    0]
 [4888  105]]
```

Model performance on test data

In [33]:

```
#Predicting the model for bag of words
mnb_bow_predict=mnb.predict(cv_test_re
views)
print(mnb_bow_predict)
#Predicting the model for tfidf features
mnb_tfidf_predict=mnb.predict(tv_test_re
views)
print(mnb_tfidf_predict)
```

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

Accuracy of the model

In [34]:

```
#Accuracy score for bag of words
mnb_bow_score=accuracy_score(test_sent
iments,mnb_bow_predict)
print("mnb_bow_score :",mnb_bow_score
)
#Accuracy score for tfidf features
mnb_tfidf_score=accuracy_score(test_sent
iments,mnb_tfidf_predict)
print("mnb_tfidf_score :",mnb_tfidf_score
)
```

```
mnb_bow_score : 0.751
mnb_tfidf_score : 0.7509
```

Print the classification report

In [35]:

```
#Classification report for bag of words
mnb_bow_report=classification_report(tes
t_sentiments,mnb_bow_predict,target_na
mes=['Positive','Negative'])
print(mnb_bow_report)
#Classification report for tfidf features
mnb_tfidf_report=classification_report(tes
t_sentiments,mnb_tfidf_predict,target_nam
es=['Positive','Negative'])
print(mnb_tfidf_report)
```

```
              precision    recall  f1-score   suppo
rt

   Positive       0.75      0.76      0.75      499
3
   Negative       0.75      0.75      0.75       50
07

    accuracy                           0.75     10000
   macro avg       0.75      0.75      0.75     10
000
weighted avg       0.75      0.75      0.75     1
0000

              precision    recall  f1-score   suppo
rt

   Positive       0.75      0.76      0.75      499
3
   Negative       0.75      0.74      0.75       50
07

    accuracy                           0.75     10000
```

macro avg	0.75	0.75	0.75	10000
weighted avg	0.75	0.75	0.75	10000

The precision is the ratio tp / (tp + fp) where tp is the number of true positives and fp the number of false positives. The precision is intuitively the ability of the classifier not to label a negative sample as positive.

The recall is the ratio tp / (tp + fn) where tp is the number of true positives and fn the number of false negatives. The recall is intuitively the ability of the classifier to find all the positive samples.

The F-beta score can be interpreted as a weighted harmonic mean of the precision and recall, where an F-beta score reaches its best value at 1 and worst score at 0.

The F-beta score weights recall more than precision by a factor of beta. beta == 1.0 means recall and precision are equally important.

The support is the number of occurrences of each class in y_true.

Plot the confusion matrix

In [36]:

```
#confusion matrix for bag of words
cm_bow=confusion_matrix(test_sentiments,mnb_bow_predict,labels=[1,0])
print(cm_bow)
#confusion matrix for tfidf features
cm_tfidf=confusion_matrix(test_sentiments,mnb_tfidf_predict,labels=[1,0])
print(cm_tfidf)
```

```
[[3736 1271]
 [1219 3774]]
[[3729 1278]
 [1213 3780]]
```

VII. Conclusion

- We can observed that both logistic regression and multinomial naive bayes model performing well compared to linear support vector machines.
- We present a model that performs sentiment analysis on the given dataset.
- The model classifies whether a given review is positive or negative.
- The model we presented has shown better accuracy.